

Actividad 2 - Evolución del software y costos asociados al mantenimiento de software

Jessica Carolina Osorio Hernández (ID 100112053 / josori10@ibero.edu.co)

Duván Arley Hernández Niño (ID 100110002 / dherna84@ibero.edu.co)

Corporación Universitaria Iberoamericana

Ingeniería de Software

Ingeniera: Diana Candia

Villeta Cundinamarca

Mayo 2023

1.Evolucion del Mantenimiento de Software

A medida que la industria del software ha evolucionado y se ha vuelto cada vez más sofisticada, el mantenimiento del software también ha experimentado cambios significativos. En los primeros días del desarrollo de software, el mantenimiento era principalmente reactivo, lo que significaba que los desarrolladores solo se enfocaban en corregir los errores cuando estos surgían. Sin embargo, a medida que la complejidad del software y las necesidades de los usuarios crecían, esto se convirtió en una práctica insostenible.

Hoy en día, los desarrolladores y las empresas de software tienen a su disposición una amplia variedad de metodologías y enfoques de mantenimiento, que les permiten elegir cuándo y cómo realizar el mantenimiento del software. Además, la introducción de tecnologías como la nube, los servidores y las herramientas de automatización ha hecho posible que el mantenimiento del software sea más fácil y menos costoso.

En lugar de depender de un enfoque reactivo, las empresas y desarrolladores pueden ahora planificar y realizar el mantenimiento proactivamente, lo que significa que pueden detectar y solucionar los problemas antes de que se conviertan en un gran problema para los usuarios. Las herramientas de automatización también han permitido a los desarrolladores identificar y corregir errores de manera más eficiente, lo que a su vez les permite centrarse en tareas más críticas y estratégicas.

En resumen, el mantenimiento del software ha evolucionado significativamente en las últimas décadas, pasando de un enfoque reactivo a un enfoque proactivo y estructurado. Los avances en la tecnología y las metodologías de mantenimiento han hecho posible que las empresas y desarrolladores puedan mantener el software de manera más eficiente y efectiva, lo que a su vez mejora la experiencia de los usuarios y aumenta la vida útil del software.

2.Evolucion del Software

La evolución del software ha sido una historia fascinante, desde la primera computadora electrónica en los años 40 hasta la era moderna de los dispositivos móviles y la nube. A medida que la tecnología ha avanzado, también lo ha hecho el mantenimiento de software.

En los años 60, el mantenimiento de software era reactivo y se centraba en corregir errores después de que se presentaran. Las empresas dependían en gran medida de los desarrolladores de software para solucionar los problemas y, en muchos casos, esto resultaba en costosos retrasos y fallas en el software.

En los años 70, se desarrollaron las primeras metodologías de mantenimiento preventivo. Los desarrolladores comenzaron a utilizar técnicas de análisis de fallas y diseño modular para prevenir problemas antes de que ocurrieran. El mantenimiento de software se convirtió en una parte integral del ciclo de vida del software y se comenzó a documentar y registrar los cambios en el software.

En los años 80, la llegada de las redes de computadoras y la distribución de software en varios equipos y sistemas generó la necesidad de una mayor estandarización y control de versiones. Se introdujeron herramientas de gestión de configuración de software que permitían a los desarrolladores rastrear y controlar los cambios en el software.

En los años 90, el mantenimiento de software se centró en la reutilización de código y la integración de sistemas. Se desarrollaron herramientas de ingeniería inversa que permitían a los desarrolladores analizar el software existente y crear modelos de diseño para su posterior reutilización. También se produjo una mayor adopción de la ingeniería de software y las metodologías ágiles.

En la era moderna, la nube, los dispositivos móviles y la inteligencia artificial han transformado la forma en que se desarrolla y mantiene el software. Se han desarrollado nuevas técnicas y herramientas para el mantenimiento de software en la nube, como el monitoreo y la automatización, que permiten a los desarrolladores mantener y actualizar el software de manera más eficiente.

En resumen, la evolución del software ha sido impulsada por la necesidad de mantener el software actualizado y funcional, ante las necesidades del usuario.

3.Escenario

Kotex es un supermercado que hace 6 meses adquirió un software de inventariado. Sin embargo, a lo largo del tiempo han surgido algunos inconvenientes con el software, por lo que el encargado ha decidido realizar un mantenimiento. Para ello, ha contratado a un ingeniero de software al que ha entregado un documento con toda la información del software.

Según el documento, el software es un programa de inventariado tipo stand-alone desarrollado con una versión compatible para equipos con sistema operativo Windows 8. Es un programa portable y no requiere conexión a Internet. Maneja una base de datos SQL y un lenguaje de programación PHP, con HTML, CSS y algunas librerías para ajustes visuales del programa. Su funcionalidad se basa en 3 roles: Ventas, Almacén y Administración.

El rol de Almacén es el encargado de registrar todos los productos que se encuentran a la venta, así como asignar los precios. El rol de Ventas es el encargado de registrar cada venta

que se realice y llevar una contabilidad mensual. El rol de Administración tiene acceso a las ventas y al almacén, y es el encargado de supervisar y administrar los usuarios.

El programa cuenta con una ventana inicial de logueo, donde, dependiendo del usuario ingresado, el software redireccionará al usuario a las ventanas indicadas para su rol. También se indica que toda la información de la base de datos depende de un equipo padre, en donde se realizan los ajustes o modificaciones.

Estas son algunas de las especificaciones que se encuentran en el documento.

De la anterior información el encargado indico 5 particularidades o inconvenientes que se vienen presentando con el software.

1. Se ha realizado la actualización del sistema operativo en algunos equipos, sin embargo, el programa no se ejecuta.
2. Es importante destacar que, si el equipo padre se apaga de forma automática, el programa dejará de funcionar en los demás equipos.
3. Durante el uso del software, en algunos casos se han detectado problemas de desajuste en algunos elementos visuales del programa.
4. El encargado de almacén tiene que ingresar producto por producto, no existe una opción de cargar de forma automática los datos de una hoja de calculo
5. Se desea una mejora para el rol de ventas, en donde permita al usuario generar un registro mensual de los productos vendidos y realizar una descarga en Excel de este, con el fin de facilitar la contabilidad que se realiza cada mes.

4. Costos

Mantenimiento Correctivo	Mantenimiento Adaptativo
<p>Costo promedio: Entre 100.000 y 200.000 pesos colombianos por hora de soporte técnico o desarrollo.</p> <p>Ejemplo: Si se requieren 10 horas de trabajo para corregir un problema, el costo promedio estaría entre 1.000.000 y 2.000.000 pesos colombianos.</p>	<p>Costo promedio: Entre 2.000.000 y 5.000.000 pesos colombianos por funcionalidad adicional o modificación del software existente.</p> <p>Ejemplo: Si se necesita desarrollar una nueva característica y el costo promedio es de 3.000.000 pesos colombianos, ese sería el costo asociado a ese tipo de mantenimiento.</p>
Mantenimiento Perfectivo	Mantenimiento Preventivo
<p>Costo promedio: Entre 1.500.000 y 3.000.000 pesos colombianos por análisis y mejoras en calidad, rendimiento o usabilidad.</p> <p>Ejemplo: Si se contrata un consultor de usabilidad por un total de 2.000.000 pesos colombianos para mejorar la experiencia del usuario, ese sería el costo estimado para este tipo de mantenimiento.</p>	<p>Costo promedio: Entre 1.000.000 y 2.500.000 pesos colombianos por servicios de pruebas, auditorías, actualizaciones o monitoreo continuo.</p> <p>Ejemplo: Si se contrata un servicio de pruebas y aseguramiento de calidad por un total de 1.500.000 pesos colombianos al año, ese sería el costo promedio para el mantenimiento preventivo.</p>

VENTAJAS	DESVENTAJAS
<p>Reducción de costos en el mantenimiento correctivo:</p> <p>La documentación facilita la identificación y corrección de problemas, lo que puede reducir el tiempo y el costo asociados con el mantenimiento correctivo. Los desarrolladores y administradores pueden obtener información detallada sobre el software, lo que les permite abordar los problemas de manera más eficaz.</p>	<p>Costos iniciales de creación y mantenimiento de la documentación: existen costos iniciales de creación y actualización de la documentación. Esto puede requerir tiempo y recursos del desarrollador o el empleo de especialistas en documentación.</p>

<p>Ahorro de costos en el mantenimiento adaptativo:</p> <p>La documentación proporciona instrucciones claras sobre la arquitectura y la funcionalidad del software, lo que facilita la realización de cambios o la adición de nuevas funciones. Esto puede reducir los costos de mantenimiento adaptativo al optimizar el proceso de desarrollo y minimizar los errores.</p>	<p>Mayores costos de mantenimiento correctivo: sin documentación, los problemas pueden tardar más en identificarse y resolverse y requieren más recursos. Esto puede aumentar los costos asociados con el mantenimiento correctivo, ya que los desarrolladores deben realizar investigaciones adicionales para comprender el software y solucionar problemas.</p>
<p>Reducción de costos en el mantenimiento perfecto:</p> <p>Con la documentación, se pueden identificar áreas de mejora en la calidad, rendimiento o usabilidad del software. Al realizar mejoras basadas en la documentación, se evitan costos adicionales asociados a pruebas extensivas o errores introducidos durante las modificaciones.</p>	<p>Aumento del costo del mantenimiento adaptativo: la falta de documentación puede dificultar los cambios o la adición de nuevas funciones. Esto puede aumentar el costo del mantenimiento adaptativo porque los desarrolladores deben dedicar más tiempo a examinar la estructura y la funcionalidad del software existente antes de realizar los cambios deseados.</p>
<p>Disminución de costos en el mantenimiento preventivo:</p> <p>La documentación permite un mejor seguimiento y control del software, lo que ayuda a prevenir futuros problemas y reducir costos en caso de errores o interrupciones en el funcionamiento del sistema.</p>	<p>Aumento de los costos totales de mantenimiento: sin documentación, es más difícil identificar áreas de mejora en la calidad, eficiencia o facilidad de uso del software. Esto puede aumentar el costo asociado con el mantenimiento completo, ya que es posible que se requieran pruebas exhaustivas y modificaciones adicionales para lograr los resultados deseados.</p>

Gracias a la documentación, los desarrolladores pueden resolver problemas de manera más eficiente, ahorrando tiempo valioso. Supongamos que el costo promedio por hora de un desarrollador es de \$50.000, y gracias a la documentación, logran resolver un problema en 2 horas en lugar de 4. En este caso, se ahorrarían \$100.000.

En contraste, en el segundo escenario, sin una documentación adecuada, los desarrolladores enfrentan una curva de aprendizaje más pronunciada. Supongamos que debido a la falta de documentación, el tiempo invertido en resolver un problema se duplica a 8 horas en lugar de 4. Esto resultaría en un costo adicional de \$200.000

Además, sin documentación, las actualizaciones y mejoras pueden ser más riesgosas y propensas a errores. Si un desarrollador comete un error durante una actualización debido a la falta de comprensión del sistema, podría causar un tiempo de inactividad de 6 horas, lo que resulta en una pérdida estimada de \$300.000

Bibliografía

- Pérez Carvajal, R. J. (2016). Mantenimiento del software (UF1894). Málaga: IC Editorial (Cap 1 p11-31 y Cap 2 p35-57). Recuperado de: **<https://elibro.net/es/lc/biblioibero/titulos/44523>**
- Bourque, P. & Fairley, R. E. (Eds). (2014). Swebok. Guide to the Software Engineering Body of Knowledge - Versión 3.0. IEEE Computer Society (Cap 5. Sección 1 1.1-1.6 y Sección 2 2.1-2.4). Recuperado de: **<https://ieeecs-media.computer.org/media/education/swebok/swebok-v3.pdf>**
- Channon S. & Straub B. (2021). Pro Git - Versión 2.1.22-4-g2264d13. Apress. Recuperado de: **<https://git-scm.com/book/es/v2/>**
- Solución SaaS <https://www.kizeo-forms.com/es/solucion-saas-en-que-consiste-y-sus-ventajas/#:~:text=%C2%BFQu%C3%A9%20es%20una%20soluci%C3%B3n%20SaaS,ordenadores%20y%20otros%20equipos%20inform%C3%A1ticos.>