

MANTENIMIENTO DE SISTEMAS DE SOFTWARE

AUTORA: HELLEN CUBERO LEDEZMA

NOVIEMBRE: 2020



San Marcos

Introducción

El mantenimiento de sistemas de software es el proceso que se realiza después de entregar el proyecto a los clientes, generalmente se ejecutan acciones correctivas, preventivas, adaptativas o perfectivas, para garantizar que el software continúe brindando los servicios o cumpliendo los objetivos para los que fue implementado.



Contenido

Introducción y definiciones	3
Tipos de mantenimiento	4
Mantenimiento correctivo	5
Mantenimiento adaptativo	5
Mantenimiento perfectivo	6
Mantenimiento preventivo	6
Principales retos del mantenimiento de software	6
Sistemas heredados	6
Técnicas de gestión del mantenimiento: Ingeniería inversa, Refactorización y Métricas	7
Conclusiones y recomendaciones	10
Referencias bibliográficas	10

Introducción y definiciones

Según la Universidad de Valencia (s.f), un producto de software abarca muchos aspectos y características que provocan que sea totalmente necesario supervisar su funcionamiento correcto durante un tiempo después de la entrega del mismo. Ante la dificultad de garantizar el comportamiento correcto del programa en circunstancias no previstas, los test de aceptación del producto incluyen pruebas a largo plazo del software (a petición del cliente). A esta fase de supervisión se le denomina **fase de operación**. Sólo cuando termina esta fase el cliente acepta definitivamente el producto, que había sido aceptado provisionalmente al ser entregado (fase de transferencia). Más tarde, es posible que el software necesite ser modificado, ya sea consecuencia de la detección de errores o bien ante nuevas exigencias y/o necesidades del usuario del sistema. A esta fase se le conoce como **fase de mantenimiento**. Es importante reseñar que durante estas fases de operación y mantenimiento (OM) se debe generar y actualizar el documento de historia del proyecto (DHP); documento que incluye todos los errores (y sus correcciones) y/o modificaciones realizadas en el producto.

Además menciona que entre las características sobresalientes del software destacan:

- El software no envejece.
- El mantenimiento del software incluye adaptar el paquete o sistema a nuevas situaciones como por ejemplo:
 - Cambio de hardware
 - Cambio de sistema operativo
- Todo sistema de software conlleva mejoras o añadidos de manera indefinida.

Ciclo de vida y el mantenimiento de software

Las principales fases en el ciclo de vida del software definidas según la Universidad de Valencia (s.f), son las siguientes:

- Análisis y definición de requisitos

- Especificación
- Diseño
- Programación
- Prueba e instalación
- Operación y mantenimiento.

Concepto de mantenimiento de software

Según 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. (Universidad de Valencia, s.f).

Conceptos

Autor	Concepto
IEEE 1219	"la modificación de un producto software después de haber sido entregado [a los usuarios o clientes] con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno".
ISO 12207	"El Proceso de Mantenimiento contiene las actividades y tareas realizadas por el mantenedor. Este proceso se activa cuando el producto software sufre modificaciones en el código y la documentación asociada, debido a un problema o a la necesidad de mejora o adaptación. El objetivo es modificar el producto software existente preservando su integridad. Este proceso incluye la migración y retirada del producto software. El proceso termina con la retirada del producto software".
Pressman [1998]	"la fase mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente".

Ilustración 1: Conceptos de mantenimiento de software
Fuente: León Socha, s.f

Tipos de mantenimiento

Existen 4 tipos de mantenimiento:

- Correctivo
- Adaptativo
- Perfectivo
- Preventivo

Según la Universidad de Valencia (s.f), la descripción y objetivos de cada mantenimiento son los siguientes:

Mantenimiento correctivo

Tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de provocar un fallo. Un fallo se produce cuando el comportamiento de un sistema difiere con respecto al comportamiento definido en la especificación.

Los fallos en un sistema software pueden ser:

- Procesamiento (salidas incorrectas de un programa).
- Rendimiento (tiempo de respuesta demasiado alto).
- Programación (inconsistencias en el diseño).
- Documentación (inconsistencias entre la funcionalidad de un programa y el manual de usuario).

Mantenimiento adaptativo

Consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el que se ejecuta. Desde cambios en el sistema operativo, pasando por cambios en la arquitectura física del sistema informático, hasta en el entorno de desarrollo del software. Este tipo de mantenimiento puede ser desde un pequeño retoque hasta una reescritura de todo el código.

Los cambios en el entorno de desarrollo del software pueden ser:

- En el entorno de los datos (por ejemplo, cambiar sistema de ficheros por BD relacional).
- En el entorno de los procesos (por ejemplo, migración a plataforma con procesos distribuidos).

Este mantenimiento es cada vez más frecuente debido a la tendencia actual de actualización de hardware y sistema operativo, cada cierto tiempo.



Mantenimiento perfectivo

Es un conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Se divide en dos tipos:

- Mantenimiento de ampliación: incorporación de nuevas funcionalidades
- Mantenimiento de eficiencia: mejora en la eficiencia de ejecución.

Mantenimiento preventivo

Hace referencia a la modificación del software para mejorar las propiedades de dicho software (calidad y mantenibilidad) sin alterar sus especificaciones funcionales. Incluir sentencias que comprueben la validez de los datos de entrada, reestructuración de los programas para aumentar su legibilidad o incluir nuevos comentarios. Este tipo de mantenimiento utiliza las técnicas de ingeniería inversa y reingeniería. El mantenimiento para la reutilización especializado en mejorar la reusabilidad del software se incluye en este tipo.

Principales retos del mantenimiento de software

El proceso de mantenimiento de software no debe deteriorar la calidad del software o sistema de software. Uno de los retos principales es mantener su fiabilidad.

Sistemas heredados

Según la Universidad de Valencia (s.f), el código "heredado" (legacy code), es código desarrollado hace algún tiempo, con técnicas y herramientas en desuso, en ocasiones por personas que no se encargan de su mantenimiento. Además, puede que incluso este código haya pasado varias actividades de mantenimiento, pero por otra parte, el volver a reescribir no compensa por la carga financiera que supuso y la

necesidad de su amortización.

Leyes del mantenimiento de software

- **Continuidad del cambio:** un programa utilizado en un entorno del mundo real debe cambiar para adaptarse al entorno y nuevas funcionalidades.
- **Incremento de la complejidad:** cuando se realizan cambios en un programa la estructura se vuelve más compleja si no se utilizan técnicas de ingeniería del software.
- **Evolución del programa:** es un proceso autorregulado. Se mantienen las tendencias e invariantes de las propiedades del programa.
- **Conservación de la estabilidad organizacional:** la carga que supone el desarrollo de un programa es constante e independiente de los recursos dedicados a lo largo del tiempo de vida del mismo.
- **Conservación de la familiaridad:** el incremento en el número de cambios introducidos con cada versión (release) es casi constante.

Técnicas de gestión del mantenimiento: Ingeniería inversa, Refactorización y Métricas

Según la Universidad de Valencia (s.f), los gestores de las organizaciones deben ser conscientes de:

1. La importancia de las tecnologías de información para la organización.
2. El software es activo corporativo y puede suponer una ventaja competitiva.

Deben centrar las soluciones de gestión en dos aspectos importantes: recursos dedicados al mantenimiento y la gestión de la calidad.

Ingeniería inversa



Es uno de los principales métodos utilizados en el mantenimiento y hace referencia al proceso de analizar un sistema para identificar sus componentes e interrelaciones, así como crear representaciones del sistema en un nivel de abstracción más elevado. Reinterpretar un programa para documentarlo.

Refactorización

Es un proceso que se realiza para mejorar el código, con un código más efectivo y que pueda facilitar la integración de nuevos elementos sin incurrir en errores nuevos, además de mejorar el análisis de errores y la necesidad de mantenimiento del software. No se obtiene un producto nuevo, se mejora el existente aplicando métodos como encapsulación, reformato o la extracción para optimizar el código sin cambiar su contenido. (IONOS, 2020).

Métricas

Según Silicia (2009), se han propuesto cientos de métricas para el software pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones complejas, otras son esotéricas (difícil de comprender) y otras violan las nociones básicas intuitivas.

Las características que deberían acompañar a las métricas efectivas del software, son las siguientes:

- Simples y fáciles de calcular
- Intuitivamente persuasivas
- Consistentes y objetivas
- Independientes del lenguaje de programación
- Eficaz mecanismo para la realimentación de calidad.

Las métricas de software se pueden clasificar en tres categorías:

1. **Métricas de producto:** describen las características del producto que de alguna forma determinan la mantenibilidad, por ejemplo el tamaño, complejidad, diseño.
2. **Métricas del proceso:** pueden ser utilizadas para mejorar el desarrollo y mantenibilidad del software, por ejemplo, la eficacia de eliminar defectos durante el desarrollo.
3. **Métricas de proyecto:** describen las características y ejecución del

proyecto, por ejemplo el número de desarrolladores, el coste, planificación y productividad del software.

Para obtener más información, observe este [enlace](#).

**ESTA LECTURA ESTÁ BASADA
CASI EN SU TOTALIDAD EN LA
LECTURA LLAMADA
MANTENIMIENTO DE LA
UNIVERSIDAD DE VALENCIA, EL
ENLACE SE ENCUENTRA EN LAS
REFERENCIAS BIBLIOGRÁFICAS.**

Conclusiones y recomendaciones

La mantenibilidad de un sistema consiste en la facilidad mantenimiento del software, es la medida cualitativa de la facilidad de comprender, corregir, adaptar y/o mejorar el software. (Universidad de Valencia, s.f).

Este proceso es necesario realizarlo siguiendo las métricas y buenas prácticas de la ingeniería de software, para garantizar que el producto puede continuar siendo utilizado y brindando valor a la empresa o institución.

Todos los procesos de mantenibilidad que se ejecutan en un sistema deben obedecer al principio de calidad y legibilidad del código para facilitar su interpretación. Además, cada cambio y/o mejora debe ser documentada y se recomienda que exista un equipo de trabajo especializado en realizar los mantenimientos necesarios.

A modo de recomendación, observe los enlaces definidos en cada sección y consulte la lectura base de esta lectura.

Referencias bibliográficas

Universidad de Valencia. (s.f). Mantenimiento. Recuperado de <http://informatica.uv.es/iiguia/2000/IPI/material/tema7.pdf>.

IONOS. (2020). Refactorización: cómo mejorar el código fuente. Recuperado de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-la-refactorizacion/>

Silicia, M.A. (2009). Métricas del mantenimiento de software. Recuperado de <https://cutt.ly/Zhvki3p>.



www.usanmarcos.ac.cr

San José, Costa Rica