

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE: CUENCA

CARRERA DE INGENIERÍA DE SISTEMAS

**“DESARROLLO DE UN PLAN DE GESTIÓN DE
MANTENIMIENTO DE SOFTWARE PARA EL
DEPARTAMENTO DE SISTEMAS DE LA UNIVERSIDAD
POLITÉCNICA SALESIANA BASADO EN LA NORMA
ISO/IEC 14764:2006”**

**TESIS PREVIA A LA
OBTENCIÓN DEL
TÍTULO DE INGENIERA
DE SISTEMAS.**

Autor: Verónica Alexandra Herrera Caldas

Director: Ing. Mauricio Sergio Ortiz Ochoa

CUENCA – ECUADOR

2015

DECLARATORIA

Yo, Verónica Alexandra Herrera Caldas con CC: 0102993909 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que se han consultado las referencias bibliográficas que se incluyen en este documento.


A través de la presente declaración cedo el derecho de propiedad intelectual correspondiente a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.



Verónica Alexandra Herrera Caldas

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Verónica Alexandra Herrera Caldas, bajo mi supervisión.



Ing. Mauricio Sergio Ortiz Ochoa

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Al Ing. Mauricio Ortiz por su colaboración, entendimiento e infinita paciencia en el seguimiento del desarrollo de este proyecto. A la Ing. Bertha Tacurí por la oportunidad. Wilson y Edison gracias por confiar en mí.

Vero.

DEDICATORIA

A mi madre Ana y mi hermano Juan Pablo por el impulso de todos estos años de estudio,
Maty y Sandro gracias por ser parte de mi vida, son mi alegría y mi todo.

Find the courage to be happy...

Vero.

RESUMEN

La presente tesis contiene el plan de mantenimiento de software basado en la norma ISO/IEC 14764 para el Departamento de Sistemas de la Universidad Politécnica Salesiana. La aplicación del mismo es de suma importancia en la arquitectura del ciclo de vida del software, puesto que proporcionará la guía para realizar el mantenimiento a las aplicaciones desarrolladas y no contratar a empresas externas. En una primera parte se describe la problemática, los objetivos, el alcance, y la revisión del estado del arte en basado en un estudio de artículos y libros.

Como segunda parte se hace hincapié en teoría relacionada, los tipos de mantenimiento, la norma ISO/IEC 14764, y algunos casos de estudio de mantenimiento de software. Más adelante en el capítulo 3 se analizan algunos aspectos a tener en cuenta al desarrollar el plan de mantenimiento como son: problemas, procesos, actividades, software de mantenimiento, y apreciaciones de algunos autores en el momento de interpretar y desarrollar el plan de mantenimiento en base de normas.

Por último, en el capítulo 4 se diseñan las plantillas para el mantenimiento del software en base de la norma. A continuación, se presentan conclusiones y recomendaciones acerca del tema, y futuros trabajos.

ABSTRACT

This thesis contains the software maintenance plan based on ISO / IEC 14764 standard for the Systems Department of the Salesian Polytechnic University. Its usage is important to the architecture of a software's life cycle, as it provides guidance for the maintenance of developed applications and avoids hiring outside companies. The first part describes the importance, objectives, scope, and review of the state of the art based on a study of articles and books.

The second part emphasizes on related theory, types of maintenance, ISO / IEC 14764 standard, and some case studies of software maintenance. Chapter 3 analyzes some aspects to consider when developing a maintenance plan some of these being: problems, processes, activities, software maintenance, and insights of some authors when interpreting and developing a maintenance plan based on standards.

Finally, Chapter 4 contains the design of templates for software maintenance based on the standard. A case study is conducted based on the templates. Then, conclusions and recommendations on the subject and future projects are presented.

ÍNDICE DE CONTENIDOS

DECLARATORIA	ii
CERTIFICACIÓN.....	iii
AGRADECIMIENTO	iv
DEDICATORIA.....	v
RESUMEN	vi
ABSTRACT	vii
ÍNDICE DE CONTENIDOS.....	viii
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS	xi
CAPÍTULO I.....	1
INTRODUCCIÓN.....	1
1.1. Motivación y Justificación.....	1
1.2. Definición del Problema	2
1.3. Objetivos.....	2
1.3.1. Objetivo General:	2
1.3.2. Objetivos específicos:	3
1.4. Alcance y Limitaciones.....	3
1.5. Estado del Arte.....	3
1.6. Organización de Capítulos.....	7
CAPÍTULO 2	9
MANTENIMIENTO DEL SOFTWARE.....	9
2.1. Características del mantenimiento del software	9
2.1.1. Necesidad del mantenimiento	11
2.1.2. Costos del mantenimiento	11
2.1.3. La evolución del software	13
2.1.4. Cuestiones claves en el mantenimiento de Software	15
2.2. Tipos de mantenimiento del software	20
2.3. Plan de mantenimiento de software	23
2.3.1. La norma ISO/IEC 14764 para el proceso de mantenimiento	23
2.4. Antecedentes del mantenimiento y casos de éxito.....	38
CAPÍTULO 3	40

CRITERIOS PARA EL DISEÑO DEL PLAN DE MANTENIMIENTO PARA EL DEPARTAMENTO DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA	40
3.1. Apreciaciones sobre el mantenimiento del software	40
3.1.1. Los problemas de mantenimiento de software	40
3.2. Actividades y Procesos del Mantenimiento de Software.....	43
3.3. Herramientas Informáticas para el mantenimiento	48
CAPÍTULO 4	51
PLANTILLA PARA EL PLAN DE MANTENIMIENTO DE SOFTWARE	51
4.1. Introducción	51
4.2. Alcance y Propósito	53
4.2.1. Alcance.....	53
4.2.2. Propósito.....	53
4.3. Documentos Aplicables	53
4.4. Definiciones	54
4.4.1. Contenido	56
4.4.2. Descripción del sistema.....	57
4.5. Fase 1: Proceso de implementación.....	57
4.6. Fase 2: Análisis de modificaciones.....	60
4.7. Fase 3: Implementación de la Modificación.....	65
4.7. Fase 4: Aceptación/Revisión del Mantenimiento	68
4.8. Fase 5: Migración	70
4.9. Fase 6: Retiro del software	77
CONCLUSIONES Y RECOMENDACIONES	82
REFERENCIAS BIBLIOGRÁFICAS	84
ANEXOS	88
ANEXO A.....PLANTILLAS PARA EL PLAN DE MANTENIMIENTO DE SOFTWARE DE LA UNIVERSIDAD POLITÉCNICA SALESIANA	89

ÍNDICE DE TABLAS

Tabla 1. Costos históricos del mantenimiento.....	13
Tabla 2. Leyes de evolución.....	14
Tabla 3. Pasos para la tarea de solicitud de modificación.....	27
Tabla 4. Pasos para la tarea de opciones.....	28
Tabla 5. Pasos para la tarea de documentación.....	28
Tabla 6. Pasos para la tarea de Revisión y Aprobación.....	30
Tabla 7. Pasos para la tarea de migración.....	33
Tabla 8. Pasos para la tarea de Operación y Formación.....	34
Tabla 9. Pasos para la tarea del plan de retiro.....	37
Tabla 10. Encuesta sobre el mantenimiento de software problemas y percepciones...	41
Tabla 11. Actividades y categorías de los trabajos de mantenimiento.....	47
Tabla 12. Áreas de proceso clave de mantenimiento de software (P = presente, A = ausente).....	48
Tabla 13. Tabla de prioridades.....	58

ÍNDICE DE FIGURAS

Figura 1. El mantenimiento del software como un proceso primario de la norma ISO/IEC 12207.	4
Figura 2. El mayor costo del ciclo de vida del software es la fase de mantenimiento..	12
Figura 3. Tipos de mantenimiento.....	22
Figura 4. Proceso Iterativo de mantenimiento ISO/IEC 14764.....	24
Figura 5. Actividad del proceso de implementación.....	25
Figura 6. Actividades para problema y modificación.	26
Figura 7. Actividades para la implementación de la modificación.	29
Figura 8. Actividades para la revisión/aceptación del mantenimiento.....	31
Figura 9. Actividades para la migración.	31
Figura 10. Actividades para el retiro.	36
Figura 11. Diagrama de contexto para encargados de mantenimiento.....	44
Figura 12. Fases del Proceso de Mantenimiento.....	52
Figura 13. Portada del Plan de Mantenimiento..	56
Figura 14. Solicitud de Modificación (Plantilla 1).....	59
Figura 15. Tipos de Modificaciones.....	60
Figura 16. Registro del Historial del MR (Plantilla 2).....	61
Figura 17. Registro de Pruebas del MR (Plantilla 3)..	61
Figura 18. Registro de Opciones (Plantilla 4)..	62
Figura 19. Registro de Asignación de Recursos (Plantilla 5).....	63
Figura 20. Registro del Análisis de la Solicitud de Modificación (Plantilla 6).....	64
Figura 21. Registro del Equipo de Trabajo (Plantilla 7)..	64
Figura 22. Listado de los Elementos a Modificar (Plantilla 8).....	66
Figura 23. Listado de los Elementos a Modificar (Plantilla 9).....	67
Figura 24. Registro de Pruebas de Aprobación (Plantilla 10).....	69
Figura 25. Proceso de Migración (Plantilla 10)..	70
Figura 26. Listado de los Elementos a Migrar (Plantilla 11)..	71
Figura 27. Listado de Herramientas (Plantilla 12)..	72
Figura 28. Listado de Herramientas (Plantilla 13)..	72
Figura 29. Registro de la Verificación de la Migración (Plantilla 14)..	74
Figura 30. Registro del Soporte del Antiguo Entorno (Plantilla 15).....	74

Figura 31. Proceso del Retiro del Software.....	78
Figura 32. Registro de Retiro del Producto de Software (Plantilla 16).....	80

CAPÍTULO I

INTRODUCCIÓN

1.1. Motivación y Justificación

“Desde que se ha desarrollado software se ha producido el mantenimiento del software, no hay nada en la teoría o en la práctica que indique que esto vaya a cambiar”. Durante estas últimas seis décadas se han presentado proyectos académicos e industriales basados en el mantenimiento del software, dando como resultado: artículos de investigación, documentos, conferencias, normativas, etc.; no obstante, el mantenimiento del software sigue siendo invisible y no declarado en muchas empresas en sus sistemas informáticos produciendo problemáticas si no se realiza el mantenimiento del software que puede llevar incluso a la baja del sistema completamente.¹

El futuro del software sigue siendo el mantenimiento del software, el seguir los lineamientos de normativas permite a los sistemas poder seguir existiendo.

La norma ISO/IEC 14764:2006² detalla la manera de realizar la gestión del proceso de mantenimiento de software de una manera adecuada, no obstante los pasos son elaborados en función de la aplicación de software. Otras normas como la ISO/IEC 12207³ hacen referencia al mantenimiento como una parte del concepto de diversos tipos de mantenimiento.

Llevar a cabo un plan de mantenimiento de software a medida mejorará la calidad del producto final, ya que el mantenimiento de software se puede hacer combinando herramientas de software, métodos y técnicas; pero todo esto está sujeto a los antecedentes de la aplicación de software desarrollado.

¹ Zvegintzov Nicholas & Parikh Girish, “60 years of Software Maintenance: Lessons Learned”, Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on Software Maintenance, Page(s): 726 - 727.

² 14764 - International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering -Software Life Cycle Processes – Maintenance.

³ ISO/IEC 12207:2002. AMENDMENT 1: Information Technology - Software Life Cycle Processes Amendment 1.

Implementar un plan de mantenimiento de software para el departamento de sistemas de la Universidad Politécnica Salesiana basado en la norma ISO/IEC 14764:2006 permitirá plasmar en hechos y mediante un caso de estudio si se puede lograr que las actividades de operación y mantenimiento del software ayuden a un funcionamiento estable, satisfacción de los clientes, administradores y usuarios finales.

1.2. Definición del Problema

El plan de desarrollo y mantenimiento de software deben realizarse a la par en cualquier organización, ambos generan gran cantidad de información que debe gestionarse y documentarse, el desconocimiento de estas actividades en el mantenimiento del software puede inducir a desvalorar su importancia, y se tiende a asociar el mantenimiento del software únicamente con la corrección de errores en los programas.⁴

El mantenimiento del software es parte de la arquitectura del ciclo de vida del software, por lo que el desarrollar un plan de gestión de mantenimiento de software para el área de desarrollo de software del departamento de Sistemas de la Universidad Politécnica Salesiana basado en la norma ISO/IEC 14764:2006, proporcionará la guía para realizar el mantenimiento a las aplicaciones desarrolladas y no contratar a empresas externas.

1.3. Objetivos

1.3.1. Objetivo General:

- Diseñar y desarrollar un plan de gestión de mantenimiento de software para el área de desarrollo de software del departamento de Sistemas de la Universidad Politécnica Salesiana que se base en la norma ISO/IEC 14764:2006

⁴ R. S. Pressman, Ingeniería de Software, Séptima ed., New York: Mac Graw Hill, 2010.

1.3.2. Objetivos específicos:

- Identificar los tipos de mantenimiento de software disponibles en la actualidad.
- Realizar un estudio del estado del arte de los estándares para mantenimiento de software.
- Diseñar un plan de mantenimiento para el área de desarrollo de la Universidad Politécnica.
- Evaluación de herramientas informáticas que automatice el mantenimiento.
- Llevar a cabo un caso de estudio en el Departamento de Sistemas de la Universidad Politécnica Salesiana.

1.4. Alcance y Limitaciones

El trabajo de tesis desarrollado proporciona un plan de mantenimiento de software en el cual se presentan los pasos y procesos para un caso de estudio dentro del departamento de sistemas de la Universidad Politécnica Salesiana. El plan de mantenimiento no se usará para productos software que son soluciones a corto plazo.

El plan de mantenimiento será usado en aquellos productos que se mantendrán un tiempo lo más largo posible dentro de la Universidad. El mantenimiento se aplicará a programas de ordenador, código, datos, y documentación de administración y más adelante en el tiempo a productos de software que sean creados durante el desarrollo de nuevo software.

Los tipos de mantenimiento que cubre el plan de gestión de mantenimiento son: correctivo, preventivo, adaptativo y perfectivo.

1.5. Estado del Arte

En la Ingeniería de Software hay un gran número de normas, sin embargo, tres de ellas son fundamentales para el mantenimiento del software: ISO/IEC 14767, IEEE 1219 e

ISO/IEC 12207. Estas normas se refieren al mantenimiento del software como actividades de desarrollo de software en áreas muy específicas, los encargados de mantenimiento deben asegurarse de que se adapten a sus necesidades específicas de desarrollo.

La norma ISO/IEC 12207⁵ describe el mantenimiento del software como uno de los cinco procesos primarios en el ciclo de vida del software. Esta norma aclara cuales son las actividades utilizadas por los desarrolladores y que deberían ser utilizadas por los encargados de mantenimiento tales como: gestión de la configuración, control de calidad, verificación, validación, revisiones y auditorías, la resolución de problemas, mejora de procesos, gestión de infraestructuras, la documentación, y la formación.

La norma internacional ISO/IEC12207 es un excelente documento para obtener una visión global del proceso de mantenimiento y sus relaciones con el desarrollo de software, Figura 1.

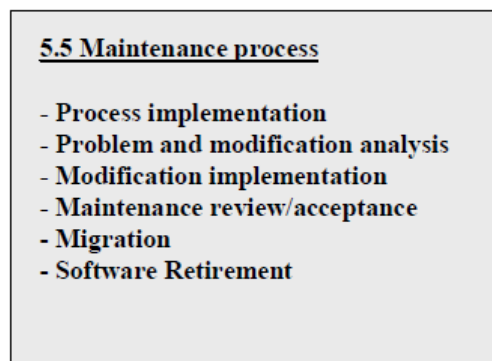


Figura 1. El mantenimiento del software como un proceso primario de la norma ISO/IEC 12207.

Fuente: <http://s3.amazonaws.com/publicationslist.org/data/a.april/ref-229/926.pdf>

Específicamente 2 normas se centran en el mantenimiento de software y son: ISO/IEC14764 e IEEE1219.

La norma ISO/IEC 14764:2006⁶ describe con mayor detalle la gestión del proceso de mantenimiento descrito en la norma ISO/IEC 12207, establece las definiciones de los

⁵ ISO/IEC 12207:2002. AMENDMENT 1: Information Technology - Software Life Cycle Processes Amendment 1.

⁶ 14764 - International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering -Software Life Cycle Processes – Maintenance.

distintos tipos de mantenimiento, además proporciona una guía para la planificación, ejecución y control, revisión y evaluación, y el cierre del proceso del mantenimiento. El ámbito de aplicación de la norma ISO/IEC 14764:2006 incluye el mantenimiento de múltiples productos de software con los mismos recursos de mantenimiento. Recalcar que el uso de la palabra "Mantenimiento" en la norma ISO/IEC 14764:2006 significa mantenimiento del software a menos que se indique lo contrario.

La norma ISO/IEC 14764:2006 proporciona el marco en el que los planes de mantenimiento de software genéricos y específicos pueden ser ejecutados, evaluados y adaptados al alcance y la magnitud de mantenimiento de productos de software dadas. Proporciona el marco, los procesos y la terminología precisa para permitir la aplicación coherente de la tecnología (herramientas, técnicas y métodos) para el mantenimiento del software.

ISO/IEC 14764:2006 proporciona una guía para el mantenimiento de software. La base para el proceso de mantenimiento y sus actividades proviene de las definiciones de la norma ISO/IEC 12207. Se definen las actividades y tareas de mantenimiento de software, y proporciona los requisitos de planificación de mantenimiento. No se refiere a la operación del software y las funciones operativas, por ejemplo, copia de seguridad, recuperación y administración del sistema, que normalmente se lleva a cabo por aquellos que operan el software.

ISO/IEC 14764:2006 está dirigido principalmente a los encargados del mantenimiento de software y adicionalmente para los responsables del desarrollo y control de calidad. También puede ser utilizado por los adquirentes y usuarios de sistemas que contienen software que puede proporcionar insumos para el plan de mantenimiento.

La norma IEEE 1219⁷ se define al mantenimiento como: “la modificación de un producto de software después de haber sido entregado a los usuarios o clientes con el fin de corregir los defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio de entorno”.

⁷ IEEE Std 1219-1998, IEEE Standard for Software Maintenance, IEEE, 1998.

IEEE 1219 Standard for Software Maintenance, hasta el año de 1998 fue el único estándar que describía el proceso de mantenimiento de software. Detalla un proceso iterativo para la gestión y ejecución de las actividades del proceso. Aunque sólo menciona las fases de desarrollo y de producción de un producto de software, éstas cubren todo su ciclo de vida, cualquiera que sea su tamaño o complejidad.

Esta norma define cambios en un producto de software a través de un proceso de mantenimiento dividido en fases, el proceso es iterativo y en cascada, con una gran semejanza al ciclo de vida del desarrollo clásico, como se menciona a continuación: Identificación del problema, análisis, diseño, implementación, pruebas del sistema, pruebas de aceptación, puesta en producción o liberación de versión. Para cada una de estas fases, el estándar define una serie de procedimientos que se han de llevar a cabo y con los que se identifica la documentación, las personas y productos de software que intervienen.⁸

Esta norma plantea un proceso de mantenimiento con gran nivel de detalle y documentación a llevar para su desarrollo, haciéndolo muy útil y necesario sobre todo en los lugares que se realiza mantenimiento del software, aquí es fundamental la traza que marca el estado y evolución de cada una de las fases pero pudiera resultar excesivo para pequeñas organizaciones que deseen aplicar dicho estándar en el mantenimiento de sus sistemas internos.

Por otro lado, las normas han desencadenado trabajos de implantar el mantenimiento de software en las organizaciones. Es el caso de MANTEMA⁹, el carácter investigativo de esta metodología parte de una tesis doctoral en el año 2000 y ha provocado otros trabajos

⁸ Estándares para el mantenimiento del software EcuRed, marzo 2015. URL: http://www.ecured.cu/index.php/Est%C3%A1ndares_para_el_mantenimiento_del_software

⁹ Macario Polo, Mario Piattini, Francisco Ruiz, and Coral Calero. MANTEMA: A Complete Rigorous Methodology for Supporting Maintenance Based On The ISO/IEC 12207 Standard. In Proc. of the 3rd European Conference on Software Maintenance and Reengineering (CSMR), pages 178–181, Amsterdam (The Netherlands), March 1999.

para seguir adaptándola a los cambios durante el tiempo¹⁰. Los trabajos a nivel de ingeniería del software en el tema de mantenimiento pasan por temas de tesis¹¹ hasta implementaciones por parte de personal de organizaciones.

El desarrollo de software va a tener un gran impacto en el mantenimiento del producto. Uno de los desarrollos es el desarrollo iterativo, el cual es particularmente útil en el mantenimiento debido a una nueva iteración se puede hacer para cada error que se fija y cada función que se añade. Esto es más eficiente que hacer todo el mantenimiento a la vez. Esto hace que sea más fácil de codificar y depurar cada parte. Además, nos permite disponer de un producto de trabajo al final de cada iteración. Dado que los proyectos se hacen más grandes, se hace más difícil para el cliente para definir los requisitos desde el principio. Agile es una forma de desarrollo iterativo que se centra en la adaptación a los requerimientos del usuario durante todo el proceso de desarrollo. Esto se logra mediante la liberación de un producto de trabajo al final de cada iteración. Algunos métodos populares de desarrollo ágil son Extreme Programming, SCRUM, Crystal, y FDD.¹² La complejidad del software hace su cambio a través del mantenimiento y evolución inevitablemente, y esto intensifica el problema de hacer frente a su complejidad a través de sus largas vidas.

1.6. Organización de Capítulos

Los capítulos están divididos en 4 partes, en el capítulo 1 se da una revisión hacia los detalles del porque realizar este proyecto y una revisión del estado del arte con respecto al mantenimiento del software. En el capítulo 2 se da una presentación teórica acerca del mantenimiento del software: normativas, metodologías, sus aplicaciones dentro del desarrollo del software y casos de estudio. En el capítulo 3 se revisa aspectos a tomar en cuenta sobre la gestión de mantenimiento de software para el área de desarrollo de software del departamento de Sistemas de la Universidad Politécnica Salesiana que se

¹⁰ Pino, F., Ruiz, F., Salas, S.: Agil mantema, technical report, Instituto de Tecnologías de Información y sistemas, Universidad de Castilla-La Mancha, (2008).

¹¹ Balseca, Caiza, "ELABORACIÓN DE UNA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE", Tesis, Escuela Politécnica Nacional, Quito, 2014

¹² Erdil, Finn, Keating, "Software Maintenance As Part of the Software Life Cycle", Department of Computer Science Tufts University, December 16, 2003.

basa en la norma ISO/IEC 14764:2006. Los resultados de este desarrollo del plan de gestión se analizan en el capítulo 4. Las conclusiones y recomendaciones en base de la finalización del proyecto se presentan en esta parte.

CAPÍTULO 2

MANTENIMIENTO DEL SOFTWARE

2.1. Características del mantenimiento del software

A través de los años algunas definiciones de Ingeniería del software¹³ contemplan la palabra y el proceso de mantenimiento dentro del ciclo de vida del software:

Definición 1:

“Ingeniería de Software es el estudio de los principios y metodologías para desarrollo y **mantenimiento** de sistemas de software”¹⁴

Definición 2:

“Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y **mantenerlos**. Se conoce también como desarrollo de software o producción de software”¹⁵

Definición 3:

“Ingeniería del Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales”¹⁶

¹³ R. S. Pressman, Ingeniería de Software, Séptima ed., New York: Mac Graw Hill, 2010.

¹⁴ ZELKOVITMZ., V ., CHAW,A . C. y GANNON,J . D.:F Principles of software Engineering and Design. Prentice-Hall, Englewoods Ciif, 1979.

¹⁵ BOEHM, B. W.: Software Engineering., IEEE Transactions on Computers, C-25, núm. 12, diciembre, pp. 1226.1241.

¹⁶ BAUER, F. L.: “Software Engineering”, Information Processing, 71, North Holland Publishing Co., Amsterdarn, 1972

Definición 4:

“La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y **mantenimiento** del software; es decir, la aplicación de ingeniería al software”¹⁷

De las anteriores definiciones se puede afirmar que el mantenimiento del software no sólo es "arreglar errores" sino que es parte fundamental del ciclo de vida del software.

La norma ISO 12207 de Procesos del Ciclo de Vida del Software define al mantenimiento como: “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”.¹⁸

Mientras que la norma ISO/IEC 14764 enfatiza la planeación del mantenimiento de software: “Conjunto de actividades destinadas a proporcionar soporte económicamente rentable para un determinado producto software. Estas actividades se realizan tanto antes de la entrega del producto como después de la entrega del mismo. Las actividades previas a la entrega incluyen las actividades destinadas a planificar, anticipar y preparar actividades de mantenimiento posteriores. Las actividades posteriores a la entrega incluyen modificaciones del producto software, formación y asistencia al usuario”¹⁹

¹⁷ IEEE: Standards Collection: Software Engineering, IEEE Standard 610.12-1990, IEEE, 1993.

¹⁸ Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología. URL: <http://ingenieria.uatx.mx/labastida/files/2011/08/MANTENIMIENTO-DE-SOFTWARE.pdf>

¹⁹ <http://ocw.unican.es/enseñanzas-tecnicas/ingenieria-del-software-ii/materiales/tema8-mantenimientoSistemasSoftware.pdf>

Mientras que la norma IEEE 1219 establece como: “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”.²⁰

2.1.1. Necesidad del mantenimiento²¹

El mantenimiento es necesario para que el software continúe satisfaciendo los requerimientos del usuario, el mantenimiento es aplicable al software desarrollado. El sistema cambia debido a las acciones correctivas y no correctivas sobre el software. El mantenimiento debe ser realizado con el objetivo de:

- Corregir fallas.
- Mejorar el diseño.
- Implementar mejoras.
- Definir interfaces con otros sistemas.
- Adaptar programas a diferentes tipos de hardware, software, características del sistema y capacidad de las redes de telecomunicaciones.
- Migrar software heredado.
- Retirar el software.

2.1.2. Costos del mantenimiento

Los costos del mantenimiento consumen gran parte de los recursos financieros del ciclo de vida del software, Figura 2. La comprensión de los factores que influyen en el mantenimiento de un sistema puede ayudar a fijar adecuadamente los costos, algunos de estos factores son²²:

²⁰ <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-ii/materiales/tema8-mantenimientoSistemasSoftware.pdf>

²¹ Computer Society, marzo 2015. URL: <http://www.computer.org/portal/web/swebok/html/ch6>

²² Computer Society, marzo 2015. URL: <http://www.computer.org/portal/web/swebok/html/ch6>

- Tipo de aplicación.
- Disponibilidad del mantenimiento de software.
- Ciclo de vida del software.
- Características del hardware.
- Calidad de diseño del software, construcción, documentación y pruebas.

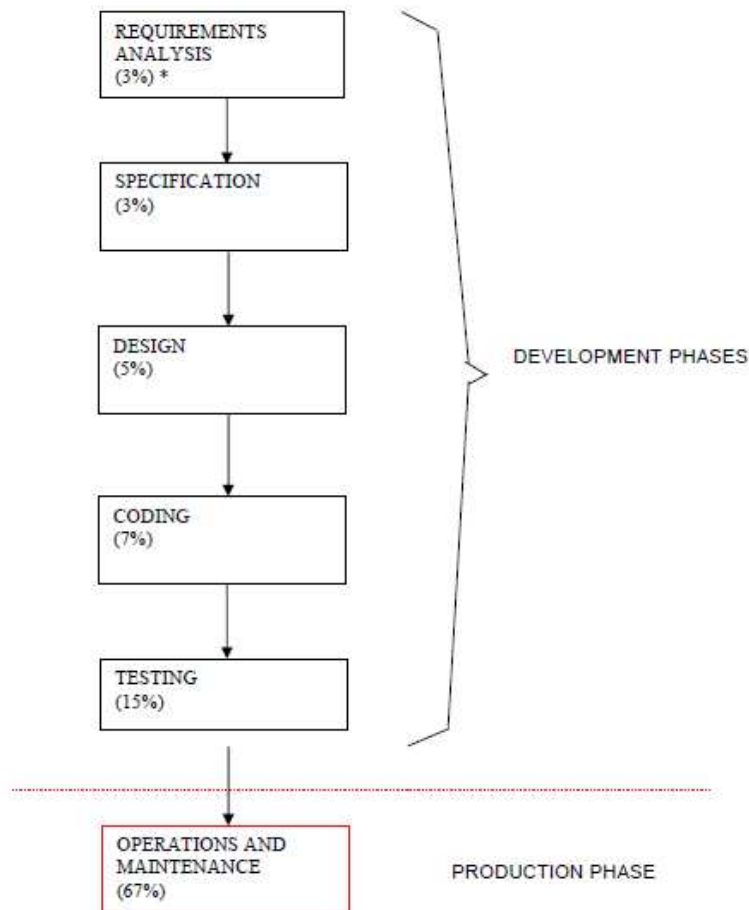


Figura 2. El mayor costo del ciclo de vida del software es la fase de mantenimiento.

Fuente: http://hepguru.com/maintenance/Final_121603_v6.pdf

Algunos estudios en base de costos del mantenimiento del software como parte del ciclo de vida del software indican que es la fase más costosa. El costo de mantenimiento del software en el tiempo es el doble que los costos de desarrollo. La tendencia es creciente con el paso del tiempo, Tabla 1.²³

²³ Ruiz y Polo, "Mantenimiento del Software", Grupo Alarcos, Dep. de Informática, Escuela Superior de Informática, Universidad de Castilla - La Mancha, Ciudad Real, 2001. URL: <http://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/s1.pdf>

Tabla 1. Costos históricos del mantenimiento. Fuente:
<http://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/s1.pdf>

Referencia	Fechas	% Mantenimiento
Pressman, 1993	Años 70	35 – 40 %
Lientz y Swanson, 1980	1976	60%
Pigoski, 1997	1980 - 1984	55%
Pressman, 1993	Años 80	60%
Rock – Evans y Hales, 1990	1987	67%
Schach, 1990	1987	67%
Pigoski, 1997	1985 - 1989	75%
Frazer, 1992	1990	80%
Pressman, 1993	Años 90	90%

2.1.3. La evolución del software

Desde 1969 se trató el tema de mantenimiento de software por parte de Lehman quien durante un periodo de 20 años formulo ocho "Leyes de la Evolución", Tabla 2. El punto que más resalta de estas leyes es abordar al mantenimiento como evolutivo.²⁴

²⁴ Computer Society, marzo 2015. URL: <http://www.computer.org/portal/web/swebok/html/ch6>

Tabla 2. Leyes de evolución. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014.

LEY	DESCRIPCIÓN
Cambio continuo, 1974	Un sistema que se utiliza en un entorno real debe cambiar o se volverá obsoleto con el pasar el tiempo.
Incremento de la complejidad, 1974	A medida que un programa en evolución cambia, su estructura tiende a ser cada vez más compleja.
Evolución prolongada del programa, 1974	La evolución de los programas es un proceso autorregulativo. Los atributos de los sistemas, como el tamaño, el tiempo entre entregas y el número de errores documentados, son aproximadamente invariantes para cada entrega del sistema.
Estabilidad organizacional, 1978	Durante el tiempo de vida de un programa, su velocidad de desarrollo es aproximadamente constante e independiente de los recursos dedicados al desarrollo del sistema.
Conservación de la familiaridad, 1978	Durante el tiempo de vida de un sistema, el cambio incremental en cada entrega es aproximadamente constante.
Crecimiento continuo, 1991	La funcionalidad de los sistemas tiene que crecer continuamente para mantener la satisfacción de los usuarios.
Decremento de la calidad, 1996	La calidad de los sistemas comenzará a disminuir a menos que dichos sistemas se adapten a los cambios en su entorno de funcionamiento.
Sistema de retroalimentación, 1971 - 1996	Los procesos de evolución incorporan sistemas de realimentación, para lograr una mejora significativa del producto.

2.1.4. Cuestiones claves en el mantenimiento de Software²⁵

Para asegurar que el mantenimiento de software sea eficaz, es importante comprender que el mantenimiento del software ofrece desafíos técnicos y de gestión únicos para los ingenieros de software. A continuación se presenta algunas de las cuestiones técnicas y de gestión relacionados con el mantenimiento del software. Se han agrupado en los siguientes apartados temáticos:

- Asuntos técnicos.
- Asuntos de gestión.
- Estimación de costos del mantenimiento
- Medidas específicas

2.1.4.1. Asuntos técnicos

a) Comprensión limitada

La comprensión limitada es la rapidez con la que un ingeniero de software puede entender dónde hacer un cambio o una corrección en el software que este individuo no desarrolló. Alrededor del 40% al 60% del esfuerzo de mantenimiento está dedicada a comprender el software a ser modificado. Por lo tanto, los ingenieros de software pueden tener inicialmente una comprensión limitada del software, y mucho tiene que hacer para remediar esto.

b) Pruebas

El repetir una prueba es significativo en tiempo y dinero. Las pruebas de regresión, la repetición de pruebas selectivas de un componente de software o para verificar que las modificaciones no han causado efectos no deseados, es importante para el mantenimiento. Cuando software realiza funciones críticas, puede que sea imposible llevarla fuera de línea para la prueba.

²⁵ Computer Society, marzo 2015. URL: <http://www.computer.org/portal/web/swebok/html/ch6>

c) Análisis de impacto

El análisis de impacto detalla cómo realizar técnicamente un análisis completo del impacto de un cambio en el software existente. Los encargados del mantenimiento deben poseer un conocimiento profundo de la estructura y el contenido del software, todo esto para poder realizar una estimación de los recursos necesarios para llevar a cabo el cambio y conocer el riesgo de hacer el cambio determinado. La solicitud de cambio se llama una petición de modificación (MR) y, a menudo llamado un informe de problemas (PR), primero debe ser analizado y traducido en términos de software. Los objetivos de análisis de impacto son:

- Determinación del alcance de un cambio con el fin de planificar y ejecutar el trabajo.
- Desarrollo de estimaciones precisas de los recursos necesarios para realizar el trabajo.
- El análisis de los costos / beneficios del cambio solicitado.
- Comunicación con otros de la complejidad de un cambio dado.

d) Mantenibilidad

La mantenibilidad en IEEE 1219 es definida como: la facilidad con la que el software se puede mantener, mejorar, adaptar, o ser corregido para satisfacer los requisitos especificados. La norma ISO 9126-01 define la mantenibilidad como una de las características de calidad.

La condición de mantenibilidad reduce los costos de mantenimiento, por lo tanto el proceso de mantenimiento del software mejorará. Muchas veces durante el desarrollo se ignora la mantenibilidad considerando que no es importante por parte de los desarrolladores siendo incluso muchas veces un requerimiento del encargado de mantenimiento. Esto se desencadena en la falta de documentación del sistema, que es la principal causa de las dificultades en la comprensión del programa y análisis de impacto.

También se ha observado que la presencia de procesos sistemáticos y maduros, técnicas y herramientas de ayuda para mejorar la capacidad de mantenimiento de un sistema.

2.1.4.2. Asuntos de gestión

a) Alineación con los objetivos de la organización

Los objetivos de la organización detallan el retorno de la inversión de las actividades de mantenimiento de software. No obstante, el mantenimiento del software a tiene el objetivo de alargar la vida del software. Las actividades de mantenimiento puede ser impulsado por la necesidad de satisfacer la demanda del usuario para las actualizaciones y mejoras del software. En ambos casos, el retorno de la inversión es mucho menos clara, de modo que la vista en alto nivel de gestión es a menudo de una de las principales actividades que consumen recursos significativos sin ningún beneficio cuantificable clara para la organización.

b) Contratación de personal

El mantenimiento a menudo no es visto como un trabajo glamoroso, en muchas organizaciones el personal de mantenimiento de software frecuentemente es mal visto. De allí que es importante atraer y mantener al personal de mantenimiento de software.

c) Proceso

El proceso de software es el conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener el software y los productos asociados. A nivel de proceso, las actividades de mantenimiento de software comparten mucho en común con el desarrollo de software, es así, que la gestión de configuración de software es una actividad crucial en ambos. El mantenimiento también requiere varias actividades que no se encuentran en el desarrollo de software como son: actividades únicas, de apoyo, de planificación del mantenimiento, Gestión de configuración de software y calidad del software.

d) Aspectos organizativos del mantenimiento

Los aspectos organizativos describen quien será responsable del mantenimiento del software, el equipo que desarrolla el software no debería estar asignado para mantener el software. Sin embargo, la decisión debe ser tomada en una base de caso. Lo que es importante es la delegación o asignación de la responsabilidad de mantenimiento a un solo grupo o persona independientemente de la estructura de la organización.

e) Externalización

Actualmente la externalización del mantenimiento es una gran industria, las grandes empresas están externalizando carteras enteras de sistemas de software, incluyendo el mantenimiento del software. Uno de los principales retos para los subcontratistas es determinar el alcance de los servicios de mantenimiento requeridos y los detalles contractuales. Las empresas de subcontratación suelen pasar varios meses evaluando el software antes de que entren en una relación contractual

2.1.4.3. Estimación de costos del mantenimiento

A efectos de planificación, estimación de costos es un aspecto importante del mantenimiento del software.

a) La estimación de costos

Las estimaciones de los costos de mantenimiento se ven afectados por muchos factores técnicos y no técnicos, la norma ISO/IEC 14764 establece que "los dos enfoques más populares a los recursos que estiman para el mantenimiento del software son el uso de modelos paramétricos y el uso de la experiencia". Muy a menudo, se utiliza una combinación de éstos.

Los modelos paramétricos:

Algunos trabajos se han realizado en la aplicación de modelos de costos paramétrico para el mantenimiento del software. Muy importante en este tipo de casos es poseer los datos de proyectos anteriores con el fin de utilizar los modelos.

Experiencia:

La experiencia, en forma de juicio de expertos, analogías, y una estructura de desglose del trabajo, varios enfoques que deberían utilizarse para aumentar los datos de modelos paramétricos. Es evidente que el mejor enfoque para la estimación de mantenimiento es combinar los datos y la experiencia empírica. Estos datos deben ser proporcionados como resultado de un programa de medición.

2.1.4.4. Medidas específicas

El encargado de mantenimiento debe determinar qué medidas son adecuadas para la organización, existe una lista de serie de medidas para cada uno de los mantenimientos:

Analizabilidad: Medidas de esfuerzo o recursos gastados en tratar de diagnosticar deficiencias o causas del fracaso del mantenedor, o en la identificación de las piezas que se desea modificar.

Mutabilidad: Medidas del esfuerzo del mantenedor asociados con la implementación de una modificación especificada

Estabilidad: Medidas del comportamiento inesperado de software, incluyendo que se encontró durante la prueba

Comprobabilidad: Medidas del mantenedor del esfuerzo y de los usuarios en el intento de probar el software modificado

Algunas de las medidas de la capacidad de mantenimiento de software se pueden obtener utilizando herramientas comerciales disponibles.

2.2. Tipos de mantenimiento del software

El tipo de mantenimiento está basado en función de la actividad y que se pretende conseguir con aquella acción. En la norma ISO/IEC 14764 para Mantenimiento de Software existen cuatro categorías de actividades de mantenimiento, y son²⁶:

- Correctivo
- Adaptativo
- Perfectivo
- Preventiva

Mantenimiento correctivo. El propósito del mantenimiento correctivo es corregir las fallas: errores de procesamiento y las fallas de rendimiento. Un programa que produce una salida incorrecta es un ejemplo de fracaso de procesamiento. Del mismo modo, un programa que no es capaz de cumplir con los requisitos en tiempo real es un ejemplo de falla de rendimiento. El proceso de mantenimiento correctivo incluye el aislamiento y la corrección de elementos defectuosos en el software. El producto de software es reparado para satisfacer los requisitos. Hay una variedad de situaciones que se pueden describir como el mantenimiento correctivo, como la corrección de un programa que aborta o produce resultados incorrectos. Básicamente, el mantenimiento correctivo es un proceso reactivo, lo que significa que el mantenimiento correctivo se realiza después de la detección de defectos con el sistema.

Mantenimiento adaptativo. El propósito del mantenimiento adaptativo es permitir que el sistema pueda adaptarse a los cambios en su entorno de datos o entorno de procesamiento. Este proceso modifica el software para interactuar correctamente con un cambio o entorno modificado. El mantenimiento adaptativo incluye cambios en el sistema, adiciones, supresiones, modificaciones, ampliaciones y mejoras para satisfacer las necesidades cambiantes del entorno en el que el sistema debe operar. Algunos ejemplos genéricos son:

²⁶ Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014.

- (i) el cambio del sistema para apoyar la nueva configuración de hardware;
- (ii) convertir un sistema de batch a operación en línea; y
- (iii) cambiar el sistema para que sea compatible con otras aplicaciones. Un ejemplo más concreto es: un software de aplicación en un teléfono inteligente puede ser mejorado para soportar la comunicación basada en WiFi, además de su actual tercera generación (3G) de comunicación celular.

Mantenimiento perfectivo. El propósito de mantenimiento perfectivo es hacer una variedad de mejoras, es decir, la experiencia del usuario, la eficiencia de procesamiento, y la mantenibilidad. Por ejemplo, los resultados del programa se pueden hacer más legible para una mejor experiencia del usuario; el programa puede ser modificado para hacerlo más rápido, aumentando así la eficiencia de procesamiento; y el programa puede ser reestructurada para mejorar su legibilidad, aumentando así su capacidad de mantenimiento. En general, las actividades de mantenimiento perfectivo incluyen la reestructuración del código, creación y actualización de la documentación, y ajustar el sistema para mejorar el rendimiento. También se le llama "el mantenimiento por el bien de mantenimiento" o "reingeniería".

El mantenimiento preventivo. El propósito del mantenimiento preventivo es evitar que ocurran problemas al modificar los productos de software. Básicamente, es mirar hacia adelante e identificar los riesgos futuros y problemas desconocidos, y tomar medidas para que no se produzcan estos problemas. Por ejemplo, los buenos estilos de programación pueden reducir el impacto del cambio, lo que reduce el número de errores. Por lo tanto, el programa puede ser reestructurado para lograr buenos estilos para hacer la comprensión programa más adelante más fácil. El mantenimiento preventivo es muy a menudo se realiza en sistemas de software críticos y de alta seguridad. El concepto de "renovación de software" es una medida de mantenimiento preventivo para evitar, o al menos posponer, las ocurrencias de fallas debido al funcionamiento continuo del sistema de software. Renovación de software es una técnica de gestión de fallos proactiva encaminada a limpiar el sistema de estado interno para evitar la ocurrencia de accidente más grave en el futuro. Se trata cuando concluido una aplicación o un sistema, se limpia su estado interno y se lo reinicia. El rejuvenecimiento puede aumentar el tiempo de inactividad de la aplicación; sin embargo, previene la aparición de fallas más graves y

costosas. En un entorno de seguridad crítica, la necesidad de llevar a cabo el mantenimiento preventivo es evidente por el ejemplo de software de control de misiles Patriot “mucho tiempo de funcionamiento” podría afectar a la precisión de la focalización. El propósito de las actividades de mantenimiento de software de mantenimiento preventivo de un sistema de seguridad de software crítico es para eliminar el riesgo o reducir su riesgo asociado a un nivel aceptable. Tenga en cuenta que un peligro es un estado de un sistema o de una situación física que, cuando se combina con ciertas condiciones ambientales, podría dar lugar a un accidente. Un peligro es un requisito previo para un accidente o una desgracia.

Un resumen de los tipos de mantenimiento se puede observar en la Figura 3.

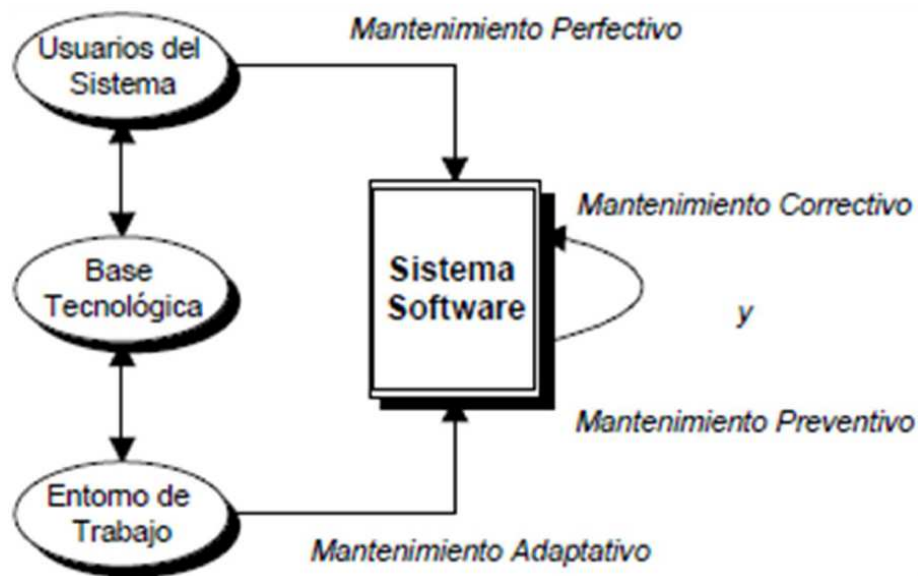


Figura 3. Tipos de mantenimiento Fuente:
<http://ingenieria.uatx.mx/labastida/files/2011/08/MANTENIMIENTO-DE-SOFTWARE.pdf>

2.3. Plan de mantenimiento de software²⁷

2.3.1. La norma ISO/IEC 14764 para el proceso de mantenimiento

El documento ISO / IEC 14764 es un estándar internacional para el mantenimiento del software, y que describe el mantenimiento utilizando los mismos conceptos como IEEE/EIA 1219, salvo que se representan de forma ligeramente diferente. Un proceso iterativo para ejecutar y gestionar las actividades de mantenimiento se describe en el documento.

La estructura básica de un proceso de ISO se compone de actividades, y una actividad se compone de tareas. Para cambiar un software operativo sin romper su integridad, las actividades necesarias se describen en el proceso de mantenimiento.

Tras la activación del proceso de mantenimiento, planes y procedimientos se desarrollan y se asignan los recursos para llevar a cabo el mantenimiento. En respuesta a una CR (change request), el código se modifica junto con la documentación pertinente. La modificación del software se ejecuta sin perder la integridad del sistema se considera que es el objetivo general de mantenimiento. El proceso de mantenimiento permite que el producto de software para migrar de su entorno inicial en su inicio a nuevos entornos. El proceso de mantenimiento se termina sobre la posible clausura del producto, conocido comúnmente como estar jubilado. El proceso de mantenimiento comprende las siguientes actividades de alto nivel:

1. Proceso de implementación.
2. Problema y análisis modificación.
3. Modificación aplicación.
4. opinión Mantenimiento y aceptación.
5. Migración.
6. Retiro.

²⁷ Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014, pp 99-111.

Las actividades del proceso de mantenimiento desarrollados por ISO/IEC se muestran en la Figura 4.

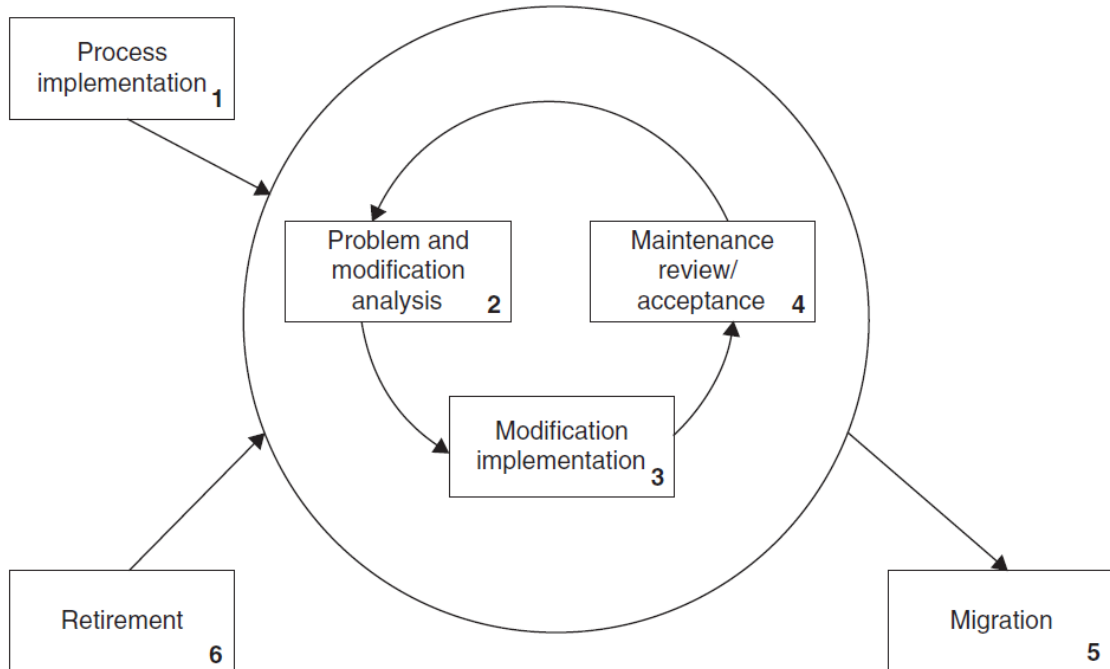


Figura 4. Proceso Iterativo de mantenimiento ISO/IEC 14764. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Cada una de estas actividades se compone de tareas, y cada tarea describe una acción específica con las entradas y salidas. Una tarea específica qué hacer, pero no cómo hacerlo. Las entradas se refieren a los elementos que se utilizan en la actividad de mantenimiento para generar salidas. Se necesitan controles efectivos para proporcionar una guía útil para que la actividad de mantenimiento produce los resultados deseados. Los resultados son los objetos generados por la actividad de mantenimiento. El soporte se refiere a los elementos que apoyan la actividad de mantenimiento.

Implementación del proceso. Esta actividad establece planes y procedimientos que deben seguirse. Un plan de mantenimiento se realiza simultáneamente con el plan de desarrollo. Figura 5 resume gráficamente la actividad de implementación de procesos con los elementos de entrada, salida, control y apoyo. La actividad de implementación proceso consiste en tres tareas principales, como se explica en el siguiente:

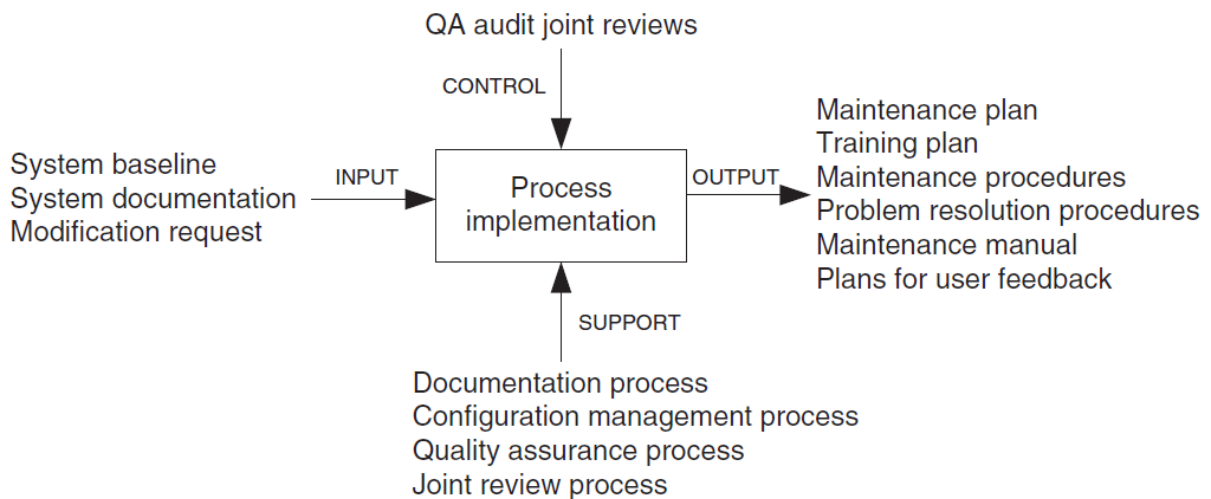


Figura 5. Actividad del proceso de implementación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Plan de mantenimiento: El plan de mantenimiento describe una estrategia para mantener el sistema, mientras que los procedimientos de mantenimiento describen en detalle cómo lograr realmente el mantenimiento. El plan también se describe cómo:

- (i) organizar y personal del equipo de mantenimiento;
- (ii) asignar responsabilidades entre los miembros del equipo; y
- (iii) los recursos de programación. La idea principal es proporcionar apoyo económico para el equipo de mantenimiento.

Las solicitudes de modificación: Los usuarios envían modificación (o modificar) solicita comunicarse con el mantenedor. El mantenedor establece procedimientos para recibir, registrar y peticiones pista usuario para modificaciones y darles retroalimentación. El proceso de resolución del problema se inicia cuando se recibe un MR (Modification request). MR se clasifican por el mantenedor ya sea como informes de problemas (correctivo) o la mejora peticiones (adaptativo y perfectivo). El proceso de mantenimiento prioriza y seguimiento de estas solicitudes individualmente como diferentes tipos de mantenimiento están allí.

La gestión de configuración CM (Configuration management): El producto de software y los cambios realizados en él durante su vida útil de mantenimiento necesitan ser controlados. Básicamente, el control de cambio se lleva a cabo mediante la aplicación e implementación de un proceso de SCM (software configuration management) aprobado.

El proceso SCM se implementa mediante el desarrollo y seguimiento de un plan de gestión de configuración CMP (configuration management plan) y los procedimientos correspondientes.

Problema y análisis de modificación. Esta actividad se invoca después de las transiciones del sistema de software de la etapa de desarrollo a la etapa de mantenimiento, y se llama iterativa cuando la necesidad de la modificación se plantea, como se muestra en la Figura 4. El mantenedor analiza la RM para identificar su impacto en la organización, el sistema existente y los sistemas de interfaz. Además, el mantenedor (i) desarrolla y documenta posibles soluciones y (ii) obtiene la aprobación de la alta gerencia para implementar las soluciones. Figura 6 resume gráficamente el problema y modificación de la actividad de análisis con los elementos de entrada, salida, control y apoyo. Esta actividad comprende cinco tareas como se indica más adelante en este apartado.

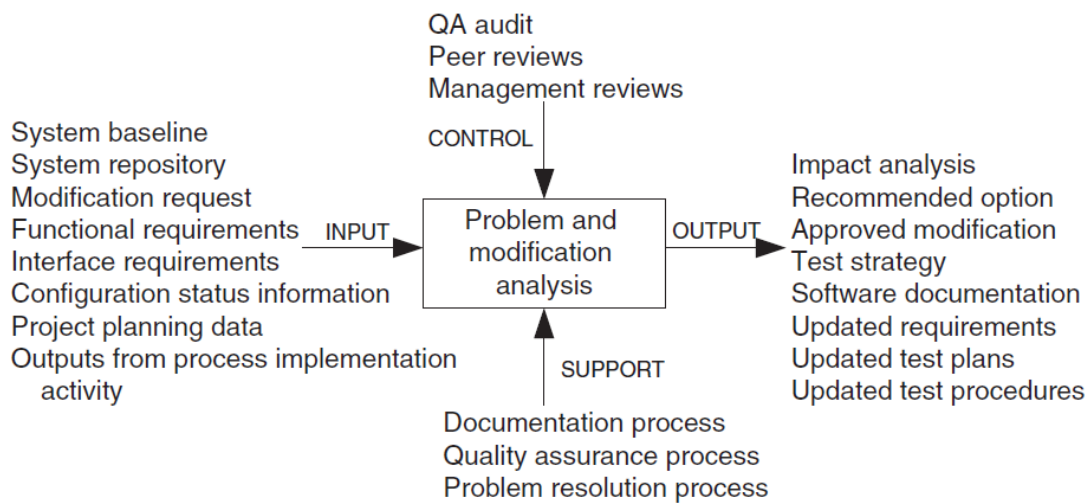


Figura 6. Actividades para problema y modificación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Análisis MR: El mantenedor analiza el MR para determinar el impacto en la organización, el hardware, el sistema existente, otros sistemas de interconexión, la documentación, las estructuras de datos, y los seres humanos (operadores, mantenedores y usuarios). El objetivo general de análisis de impacto es determinar todas las entidades que van a ser modificados y/o afectadas si el MR se va a implementar. Los pasos del análisis de impacto se dan en la Tabla 3.

Tabla 3. Pasos para la tarea de solicitud de modificación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

1. Decida si el mantenedor tiene personal suficiente para realizar los cambios propuestos.
2. Decida si o no el programa de mantenimiento ha recibido presupuesto adecuado.
3. Decida si o no hay suficientes recursos están disponibles y si el cambio propuesto afectará a algunos proyectos actuales o futuros.
4. Determinar las cuestiones operativas que se deben considerar.
5. Determinar la prioridad de manejo.
6. Clasificar el tipo de mantenimiento.
7. Determinar el impacto en los usuarios actuales y futuros.
8. Determinar las implicaciones de seguridad y protección.
9. Identificar efecto dominó.
10. Determinar las limitaciones de hardware o software que pueden resultar de los cambios propuestos.
11. Estimación de los valores de los beneficios de hacer los cambios.
12. Determinar el impacto en las listas existentes.
13. Documentar los riesgos resultantes del análisis de impacto.
14. Estimación de la evaluación que se realizará.
15. Estimar el coste de la gestión para ejecutar la modificación.
16. Coloque desarrollado artefactos menores de CM.

Verificación: El mantenedor deberá reproducir el problema y documentar los resultados de las pruebas en el entorno de laboratorio si el MR es correctivo con el fin de determinar la validez de la MR. Para las tareas de mantenimiento adaptativo y perfectivo, no se requiere verificación. El mantenedor diseña una estrategia de prueba para verificar y replicar el problema.

Opciones: El mantenedor debe describir dos o más soluciones alternativas a la MR en base a los análisis realizados. El informe de soluciones alternativas debe incluir el costo, el esfuerzo, y el calendario para la aplicación de diferentes soluciones. El mantenedor deberá realizar los pasos de la tarea que se muestran en la Tabla 4 para identificar soluciones alternativas al MR.

Tabla 4. Pasos para la tarea de opciones. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

1. El MR se le asigna una prioridad de trabajo.
2. Explorar una solución temporal para el problema. Si existe alrededor de una obra, proporcionar al usuario.
3. Identificar las necesidades concretas de la modificación prevista.
4. Calcular la magnitud y dimensiones de la modificación prevista.
5. Identificar una variedad de opciones para ejecutar la modificación prevista.
6. Estimar los impactos de las opciones en el hardware de los usuarios y del sistema.
7. Analizar los riesgos de cada opción.
8. Documento de los resultados de análisis de riesgos para cada una de las opciones propuestas.
9. Desarrollar un plan ampliamente aceptable para aplicar la modificación.

Documentación: El mantenedor documenta los MRs, los resultados del análisis y el informe opción de ejecución después de terminado el análisis y las soluciones alternativas se identifican. El mantenedor puede utilizar los pasos de la tarea que se muestran en la Tabla 5 para escribir este documento.

Tabla 5. Pasos para la tarea de documentación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

1. Asegurarse de los análisis de resultados se han completado y la documentación actualizada. Si no existen documentaciones, desarrollar nueva documentación.
2. Para mayor precisión, revise la estrategia prevista para llevar a cabo las pruebas y revisar el calendario.
3. Revisión de los recursos estimados para la precisión.
4. Revisar la base de datos para almacenar el estado de la contabilidad.
5. Describir un procedimiento para decidir si aprueba o no el MR.

Aprobación: El mantenedor presenta el informe de análisis a la autoridad competente en la organización para buscar su aprobación para la opción de cambio seleccionado. Una vez aprobado, el mantenedor actualiza los requisitos si el MR es una mejora.

Aplicación de la Modificación. En esta actividad, los mantenedores deben: (i) identificar los elementos para ser modificados y (ii) ejecutar un proceso de desarrollo a aplicar en la práctica las modificaciones. El mantenedor determina el tipo de documentación, unidades de software, y la versión del software que han de ser cambiado. Aunque el desarrollo se convierte en parte de la actividad de modificación, que se adapta a eliminar las actividades que no se aplican al esfuerzo de mantenimiento, como la obtención de requisitos y diseño arquitectónico. Para asegurar que lo modificado o los requisitos recién añadidos se implementan correctamente, los planes de prueba y los procedimientos están incluidos en el proceso de desarrollo. Además, se asegura que los requisitos que no han sido modificados no se ven afectados por la nueva implementación. Las entradas a esta actividad incluyen todo el trabajo de análisis realizado en las actividades anteriores, y la salida es una nueva línea de base de software. La Figura 7 muestra la actividad de implementación modificación.

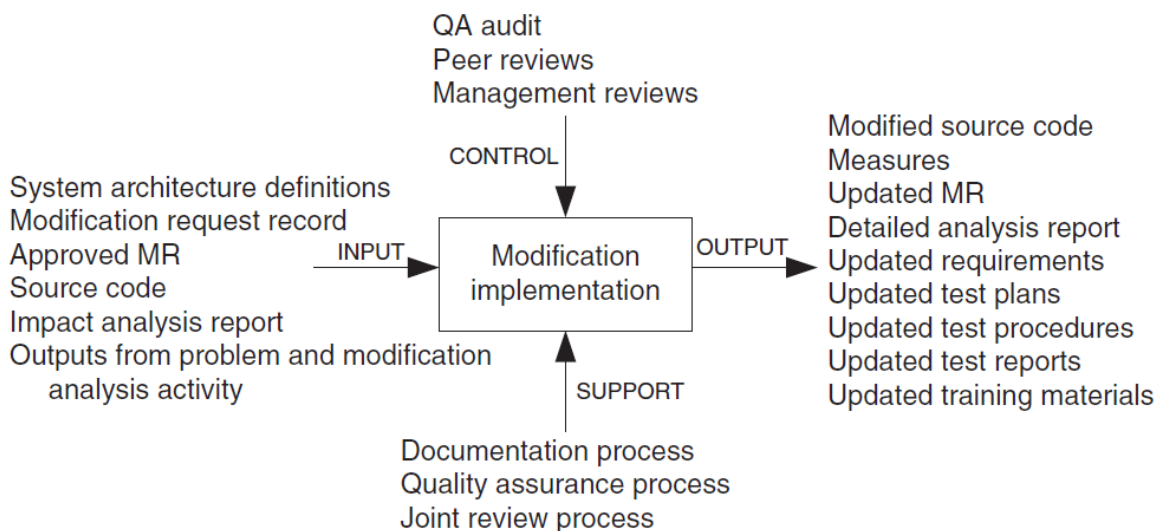


Figura 7. Actividades para la implementación de la modificación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Aceptación/Revisión del mantenimiento. Por medio de esta actividad, se garantiza: (i) los cambios realizados en el software son correctos y (ii) se realizan cambios en el software de acuerdo con las normas y metodologías aceptadas. A la actividad se añade

los siguientes procesos: (i) un proceso para la gestión de la calidad; (ii) un proceso para verificar el producto; (iii) un proceso para validar el producto; y (iv) un proceso de revisión del producto. El plan de mantenimiento debería haber sido documentado cómo se adaptaron estos procesos de apoyo para hacer frente a las características del producto de software específico. Las entradas a esta actividad incluyen el software modificado y los resultados de las pruebas. La Figura 8 resume la actividad de aceptación del mantenimiento con los elementos de entrada, salida, control y apoyo. La actividad de implementación del proceso consta de dos tareas principales: la revisión y aprobación. Los pasos de la tarea, tanto para su revisión y aprobación se enumeran en la Tabla 6.

Tabla 6. Pasos para la tarea de Revisión y Aprobación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Pasos para la tarea de Revisión.
1. Seguir los MRs de especificación de requisitos para la codificación.
2. Asegúrese de que el código es comprobable.
3. Asegúrese de que el código se ajusta a los estándares de codificación.
4. Asegúrese de que sólo los componentes de software necesarios fueron cambiados.
5. Asegúrese de que el nuevo código se integra correctamente con el sistema.
6. Asegúrese de que la documentación se actualizan con precisión.
7. El personal de CM construir elementos de software para la prueba.
8. Realizar las pruebas por una organización de pruebas independiente.
9. Realice la prueba del sistema en un sistema totalmente integrado.
10. Desarrollar informe de la prueba.
Pasos para la tarea de aprobación.
1. Obtener la aprobación de aseguramiento de calidad.
2. Verifique que el proceso se ha seguido.
3. CM prepara el paquete de entrega.
4. Llevar a cabo la auditoría de configuración funcional y física.
6. notificar a los operadores.
7. Realizar la instalación y la capacitación en las instalaciones del operador.

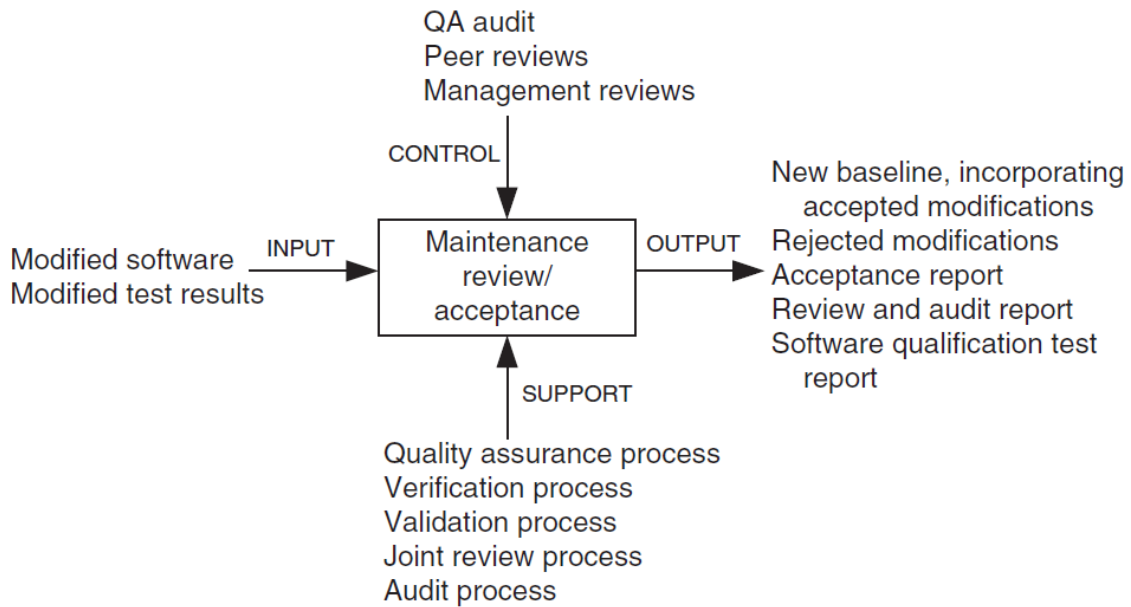


Figura 8. Actividades para la revisión/aceptación del mantenimiento. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Migración. Esto se refiere al proceso de mover un sistema de software de un entorno tecnológico a una diferente que se considera que es mejor. La migración se realiza en dos grandes fases: (i) identificar las acciones necesarias para lograr la migración y (ii) diseñar y documentar las medidas concretas que se deben ejecutar para llevar a cabo la migración. Figura 9 resume la actividad de la migración con los elementos de entrada, salida, control y apoyo. Esta actividad consta de siete tareas a explicar en los siguientes párrafos.

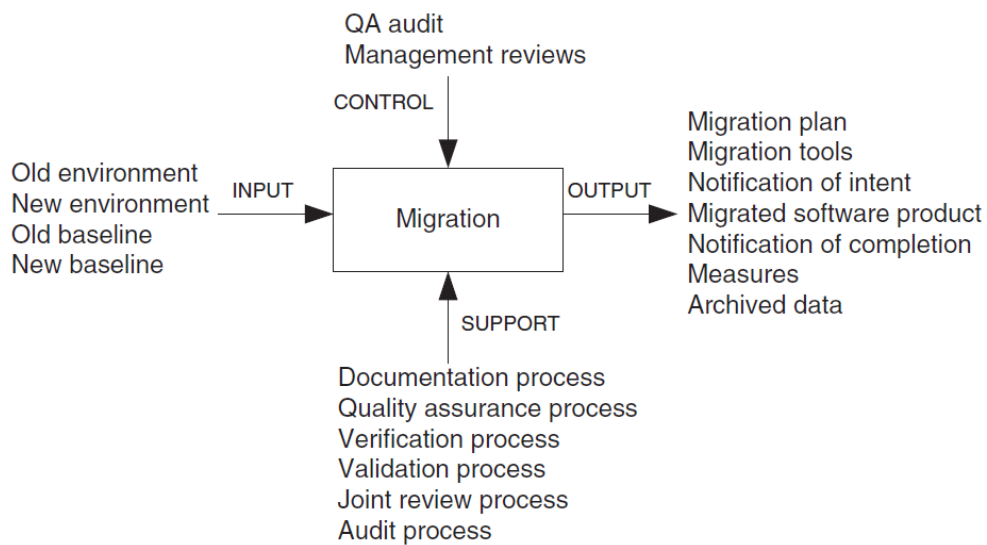


Figura 9. Actividades para la migración. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Norma de migración: Durante la migración de un producto de software de un antiguo a un nuevo entorno operativo, el mantenedor debe garantizar que cualquier producto de software o datos adicionales producidos o modificados se adhieran a la norma ISO/IEC 12207. Como parte de las tareas estándar, el mantenedor (i) identifica todos los elementos de software o datos que han sido modificados o añadidos y (ii) garantiza que las tareas se realizaron de acuerdo con la norma ISO/IEC 12207.

Plan de migración: Para la migración exitosa, un plan debe ser desarrollado, documentado, revisado y ejecutado. El mantenedor realiza los pasos de la tarea que se muestran en la Tabla 7 para escribir este documento. El plan se desarrolla en colaboración con los clientes y se ocupa de lo siguiente:

- Análisis de requerimientos y definición de la migración.
- Desarrollo de herramientas de migración.
- La conversión de producto de software y datos.
- Ejecución de la migración.
- Verificación de la migración.
- Compatibilidad con versiones anteriores con el entorno de ejecución antiguo.

Tabla 7. Pasos para la tarea de migración. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

1. Analizar los requisitos para la migración.
2. Llevar a cabo un análisis del impacto de la migración del sistema de software.
3. Haga un horario para ejecutar la migración.
4. Determinar todos los requisitos para la recopilación de datos para llevar a cabo la revisión posterior a la operación.
5. Identificar y registrar el esfuerzo de migración.
6. Identificar y reducir los riesgos.
7. Identificar las herramientas necesarias para apoyar la migración.
8. Determinar cómo el ambiente antiguo va a ser apoyado.
9. Adquirir y / o diseñar nuevas herramientas de apoyo a la migración.
10. Los productos y datos de software de partición para la conversión de forma incremental.
11. Priorizar las actividades de conversión de datos y productos de software.
12. Ejecutar los productos de software y conversiones de datos.
13. Lleve a cabo la migración de productos de software y los datos al nuevo entorno.
14. Haga funcionar el sistema migrado y el viejo sistema en paralelo tanto como sea posible.
15. Realizar pruebas para asegurar el éxito de la migración.
16. En caso de que sea necesario, seguirá prestando soporte para el entorno antiguo.

Notificación de intención: El mantenedor explica a los usuarios de: (i) por qué el soporte para el entorno antiguo ha sido discontinuado; (ii) el nuevo entorno y cuándo va a ser objeto de soporte; y (iii) la disponibilidad de otras opciones, si hay alguna, tras la eliminación del entorno antiguo.

Implementación de operaciones y entrenamiento: Una vez que un producto de software se ha mejorado mediante la modificación y probado por el mantenedor, se instala en un entorno operativo para funcionar simultáneamente con el sistema antiguo. Mediante la ejecución de lo antiguo y el nuevo sistema en paralelo, los usuarios tienen la oportunidad de familiarizarse con el nuevo sistema, por lo que la transición de lo antiguo al nuevo sistema se vuelve más paulatino. Además, se creará un entorno para el

mantenedor a fin de comparar y entender las relaciones entrada/salida del antiguo y nuevo sistema. Durante este periodo de formación también debe ser proporcionada a los usuarios. Los pasos que se indican en la Tabla 8 se pueden realizar por el mantenedor en este paso.

Tabla 8. Pasos para la tarea de Operación y Formación. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, "Software Evolution and Maintenance", Wiley, USA, 2014

Pasos para la tarea de operaciones paralelas.
1. Encuesta del sitio.
2. Instale el equipo de hardware.
3. Instale el sistema de software.
4. Ejecute las pruebas básicas para asegurar que el hardware y el software se han instalado correctamente.
5. Ejecutar tanto los nuevos y viejos sistemas en paralelo, con la carga operativa deseada.
6. Recopilar datos de los antiguos y los nuevos sistemas.
7. Analizar los datos recogidos.
Pasos para la tarea de formación
1. Identificar los requisitos para la formación de la migración.
2. Programe los requisitos para la formación de la migración.
3. Revisar la formación de migración.
4. Actualizar el plan para proporcionar capacitación.

La notificación de finalización: El mantenedor notifica a todos los sitios que el nuevo sistema entrará en funcionamiento y que el antiguo sistema se discontinúa y será desinstalado, después de la finalización de la formación y el funcionamiento en paralelo de ambos el nuevo y antiguo sistema de un número adecuado de horas. En esencia, los siguientes pasos de la tarea son realizadas por el mantenedor:

1. Anunciar la migración.
2. Documento de los aspectos específicos del centro y hacer un plan para resolverlos.
3. Archivar el antiguo sistema, incluidos los datos y el software.

4. Retirar el equipo antiguo.

Revisión después de la operación: Después de la instalación y operación de un sistema modificado, se realiza una revisión para evaluar el impacto de cambiar el sistema en el nuevo entorno. Los informes de revisión se envían a las partes competentes para la información, orientación, y demás acciones. El mantenedor ejecuta los siguientes pasos, como parte de la tarea:

1. Analizar los resultados de la ejecución de los dos sistemas simultáneamente.
2. Identificar las áreas potenciales de riesgo.
3. Resumir las lecciones aprendidas.
4. Elaborar un informe sobre el análisis de impacto.

Archivos de datos: Los datos asociados con el antiguo entorno se hacen accesibles para cumplir con los requisitos contractuales de protección de datos y auditoría. El mantenedor realiza los siguientes pasos como parte de la tarea:

1. Los datos y software antiguo se archivan.
2. Los datos y el software antiguo se ponen en múltiples respaldos.
3. Los respaldos se guardan en lugares seguros.

Retiro. Un producto de software se retira cuando se ve que ha llegado al final de su vida útil. Un análisis económico se realiza para retirar el producto y se incluye en el plan de retiro. A veces, el trabajo realizado por el producto ya no es necesaria; por lo tanto, no se reemplaza el producto retirado. En otros casos, un nuevo producto de software ya se ha desarrollado para reemplazar el sistema actual. En cualquier caso, el sistema de software debe ser retirado del servicio de una manera ordenada. Además, las consideraciones son dadas a acceder a los datos producidos por el software para ser retirado. La Figura 10 resume la actividad de retiro con los elementos de entrada, salida, control y apoyo. Todos los objetos de la actividad de retiro se controlan con CM. Esta actividad comprende cinco tareas discutidas en los párrafos siguientes.

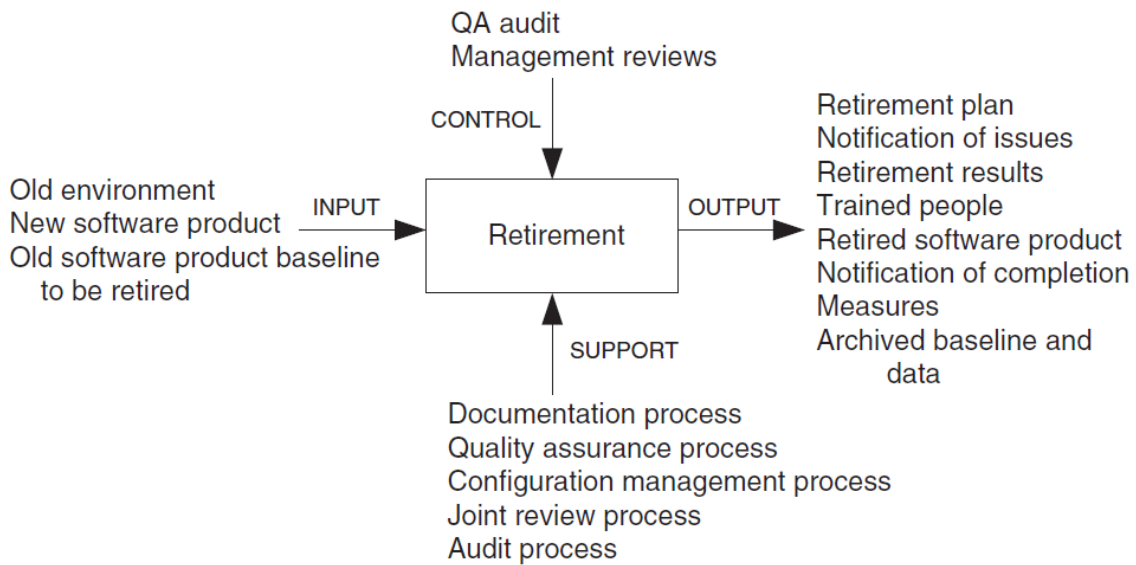


Figura 10. Actividades para el retiro. Fuente: Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014

Plan de retiro: A fin de garantizar un retiro exitoso, un plan de retiro es desarrollado, documentado, revisado, y ejecutado. El mantenedor realiza los pasos de la tarea que se muestran en la Tabla 9 para escribir este documento. El plan se desarrolla en colaboración con los clientes para hacer frente a lo siguiente:

- La transición a un nuevo sistema de software.
- Retiro de soporte parcial o total después de un período de gracia.
- La responsabilidad de ningún soporte contractual futuro.
- Archivar el sistema de software, incluyendo toda la documentación.
- La accesibilidad a los datos archivados.

Tabla 9. Pasos para la tarea del plan de retiro.

1. Analizar los requisitos para retiro de los sistemas.
2. Determine qué impactos del software de retiro tendrá.
3. Identificar un producto que sustituirá el software que se retiró.
4. Haga un horario para retirar el software.
5. Determinar la necesidad de soporte residual en el futuro.
6. Identificar y describir el esfuerzo de retiro.

Notificación de intención: El mantenedor transmite a los usuarios: (i) la razón para discontinuar el apoyo para el producto; (ii) una nota sobre el reemplazo o actualización para el sistema a eliminar, con una fecha de disponibilidad; y (iii) una lista de las otras opciones disponibles, si hay alguna, tras la eliminación del entorno antiguo.

Implementar operaciones paralelas y formación: Si hay un sistema de reemplazo para el producto de software que se retiró, se instala en un entorno operativo para funcionar simultáneamente con el sistema antiguo. Mediante la ejecución de lo nuevo y lo del antiguo sistema en paralelo, los usuarios tendrán la oportunidad de familiarizarse con el nuevo sistema de modo que la transición de antiguo al nuevo sistema se vuelve más suave. Además, se creará un entorno para el mantenedor para comparar y entender las relaciones de entrada/ salida entre el nuevo sistema y el sistema antiguo. Además, se proporciona formación a los usuarios durante este período.

Notificación de finalización: El mantenedor notifica a todos los sitios que el nuevo sistema entrará en funcionamiento y que el antiguo sistema puede cerrarse. El antiguo sistema se suele cerrar después de que el nuevo sistema está en funcionamiento durante un cierto período de tiempo. El mantenedor realiza los pasos siguientes como parte de la tarea:

1. Hacer un anuncio sobre los cambios.
2. Identificar los problemas específicos a los sitios individuales y describir cómo los va a resolver.
3. Almacenar los antiguos datos y software en un archivo.
4. Desconectar y mover la antigua infraestructura de hardware.

Archivos de datos: Los datos asociados o utilizados con el entorno anterior se pondrá a disposición de acuerdo con los requisitos contractuales relacionadas con la protección de datos y auditoría. El mantenedor ejecuta los siguientes pasos como parte de esta tarea:

1. Archiva datos y software antiguos.
2. Realiza varias copias de datos y software antiguos.
3. Mantenga los respaldos en lugares seguros.

2.4. Antecedentes del mantenimiento y casos de éxito

Caso 1:

Entre junio de 1985 y enero de 1987 operaba una máquina de radioterapia llamada Therac-25 que se utilizaba para tratar a pacientes con cáncer, pero debido a un mal funcionamiento de software terminó matando a 3 personas debido a cantidades extremadamente altas de radiación. La máquina utilizaba algunos de los mismos programas que su predecesor Therac-6, pero debido a que el productor quería reducir los costes que no incluía los mecanismos de seguridad a nivel de hardware que se habían utilizado anteriormente. El control de la máquina fue realizada mediante software con el aporte de los usuarios y por los fallas en el software esto conllevó en pacientes recibiendo una dosis más alta o varias dosis porque mostraba al usuario y operador de que no se había dado ninguna dosis. El software de Therac-25 conllevaba una mayor responsabilidad en el mantenimiento para la seguridad que el software de máquinas anteriores. Si los mecanismos de seguridad a nivel de hardware hubiesen estado presentes se habría impedido que tres personas perdieran la vida.²⁸

Caso 2:

Ariane 5 fue un cohete europeo diseñado para entregar cargas útiles en la órbita de la tierra y fue el sucesor de Ariane 4. La mayoría del software se volvió a utilizar a partir de Ariane 4, pero debido a la diferencia física entre los dos cohetes, que no se tuvo en cuenta, el cohete dirigía su curso con las señales equivocadas enviadas a los motores. Esto dio

²⁸ Nancy Leveson and Clark S. Turner. An Investigation of the Therac-25 Accidents. URL: http://courses.cs.vt.edu/cs3604/lib/Therac_25/Therac_1.html.

lugar a una trayectoria de salida equivocada de la atmósfera de la Tierra y el cohete quemó. Se produjo una falla de software cuando una conversión de punto flotante de 64 bits en un entero de 16 bits provocó un desbordamiento que no se daba en Ariane 4 porque la variable nunca podría alcanzar un valor lo suficientemente alto para que eso suceda.²⁹

Caso 3:

Durante la guerra fría un satélite soviético señaló que los Estados Unidos habían disparado cuatro misiles balísticos en Rusia. El software en el satélite había señalado por un mal cálculo de la reflexión del sol sobre las nubes como misiles y el operador del sistema creó la alerta ante este evento, luego Rusia hubiera lanzado un contraataque con sus propios misiles. Sin embargo, la alarma fue afortunadamente cuenta y una posible nueva guerra mundial fue evitada.³⁰

²⁹ Alan Dix. The Ariane 5 Launcher Failure. URL:
http://www.comp.lancs.ac.uk/computing/research/cseg/projects/tracker/sss_iceis04.pdf

³⁰ Tony Long. Sept. 26, 1983: The Man Who Saved the World by Doing ... Nothing. URL:
http://www.wired.com/science/discoveries/news/2007/09/dayintech_0926

CAPÍTULO 3

CRITERIOS PARA EL DISEÑO DEL PLAN DE MANTENIMIENTO PARA EL DEPARTAMENTO DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

3.1. Apreciaciones sobre el mantenimiento del software

Factores importantes a tomar en cuenta en la metodología de ingeniería del software es el modelo para el ciclo de vida y el proceso de mantenimiento. Dos normas que van a la par de estos factores son las normas ISO/IEC 12207 e ISO/IEC 14764 para el ciclo de vida y el proceso de mantenimiento respectivamente.

La norma ISO/IEC 12207 describe el ciclo de vida global de software, es un documento clave para tener una visión global del proceso de mantenimiento y las relaciones con el desarrollo de software. Las normas para el mantenimiento del software más usadas son: ISO/IEC 14764 e IEEE1219, no obstante se escoge la norma ISO/IEC 14764 por que utiliza y adapta los procesos de desarrollo de software cuando la necesidad de modificar el software se presenta durante su mantenimiento, y por ser una extensión de la norma ISO/IEC 12207.

Estas normas indican que el mantenimiento del software se refiere a las actividades de desarrollo de software en áreas muy específicas y los encargados de mantenimiento deben asegurarse de que se adapten a sus necesidades.

3.1.1. Los problemas de mantenimiento de software.

El mantenimiento del software no está presente en los programa de enseñanza de las escuelas³¹, esto ha provocado una falta de cultura, conocimientos, técnicas disponibles y las herramientas para los empleados que trabajan en este campo.

³¹ Pérez-Castillo, García, García Rodríguez de Guzmán y Piattini, “La Reingeniería como tópico en la docencia de la Ingeniería del Software: una Experiencia Práctica”, Instituto de tecnologías y Sistemas de Información, Universidad de Castilla – La Mancha, España, 2012.

Los empleados tienen una perspectiva interna, mientras que los usuarios y los clientes tienen una perspectiva externa. En la Tabla 10³² se enumeran 19 problemas principales del mantenimiento de software.

Tabla 10. Encuesta sobre el mantenimiento de software problemas y percepciones. Fuente: Alain A. April, 2004

Rango	Problema de Mantenimiento
1	No seguimiento de las prioridades cambiantes.
2	Técnicas de pruebas inadecuadas.
3	Es difícil medir el desempeño.
4	La documentación del software es incompleta o faltante.
5	Adaptarse a los cambios rápidos de las organizaciones de usuarios.
6	Gran atraso o peticiones.
7	Difícil de medir/demostrar la contribución de mantenimiento de software.
8	La baja moral debido a la falta de respeto y comprensión.
9	Pocos profesionales de mantenimiento con experiencia.
10	Poca metodología; pocas normas, procedimientos o herramientas específicas.
11	El código fuente del software existente es compleja y no estructurada.
12	Integración, solapamiento, y la incompatibilidad de los sistemas existentes.
13	Bajo nivel de capacitación para el personal de mantenimiento.
14	No hay planes estratégicos para el mantenimiento del software.
15	Difícil de entender y responder a las peticiones de los usuarios finales.
16	Poca comprensión y el apoyo de la Información Gestión de la Tecnología.
17	Software de sistemas de bajo mantenimiento que operan en entornos obsoletos.
18	Poca intención de rediseñar el software existente.
19	Pérdida de conocimientos cuando los empleados abandonan el equipo.

El mantenimiento del software es, por sí mismo, de gran importancia económica. Una serie de encuestas en los últimos 15 años han demostrado que para la mayoría de software, el mantenimiento ocupa entre el 40 y el 90 por ciento de los costos totales del ciclo de

³² Alain A. April, "SM^{mm} Model to Evaluate and Improve the Quality of Software Maintenance Process", Institut für Verteilte Systeme Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg Postfach 4120 39016 Magdeburg, Germany, 2004.

vida de acuerdo a lo publicado por Foster³³, el 75% de acuerdo con Rand P. Hall³⁴, 50-80% de acuerdo con Tony Scott³⁵ y más de 60% de acuerdo con Hanna³⁶. Este tipo de encuesta confirma la percepción de los usuarios de que los costos de mantenimiento son altos.

Existe la idea equivocada de que la mayoría del trabajo de mantenimiento es correctivo y no representa claramente la importancia y el valor añadido de las otras categorías de mantenimiento.

*“La parte más sustancial de los costes de mantenimiento se dedica a acomodar cambios funcionales en el software necesario para seguir el ritmo de las necesidades cambiantes de los usuarios. Con base en datos revisados, se señaló también que los sistemas con software bien estructurado eran mucho más capaces de adaptarse a esos cambios”.*³⁷

Pigoski³⁸ afirma que el mantenimiento del software es mano de obra intensiva y que la mayoría de los costos están asociados con el componente de recursos humanos. Debido a las economías de escala y nuevos procesos de producción, el hardware ya no es un costo representativo para la mayoría de los sistemas de software modernos.

³³ Foster, J.R., Munro, M. (1987) A Documentation Method Based on Cross-Referencing, Proceedings of the IEEE Conference on Software Maintenance, IEEE Computer Society Press, Los Alamitos, California, pp 181-185.

³⁴ Hall, P.V.A. (1987) Software components and reuse: Getting more out of your code. Information Software Technology, Vol. 29, No. 1, Feb, pp. 38-43.

³⁵ Scott, T. Farley, T. (1988) Slashing Software Maintenance Costs, Business Software Review, Indianapolis, March.

³⁶ Hanna, M. (1993) Maintenance Burden Begging for a Remedy, Datamation, April, pp.53-63.

³⁷ Fornell, G.E. (1992) cover letter to report, “Process for Acquiring Software Architecture,” July 10, http://www.stsc.hill.af.mil/resources/tech_docs/gsam2/chap_2.DOC (link tested May 11th 2004).

³⁸ Pigoski T.M. Practical software maintenance: Best practice for managing your software investment. John Wiley & Sons: New York, NY, 1997; 384 pp.

Banker³⁹ afirma que el tamaño y la complejidad del software influyen en gran medida en sus costos de mantenimiento y los esfuerzos de modificación. A tal punto que Boehm⁴⁰ publica que por cada dólar invertido en el desarrollo de software habrá un gasto de 2 dólares en mantenimiento. Los costos de mantenimiento se pueden plantear como una función del número de instrucciones en el código fuente del software.

Lehman⁴¹ indica que la estructura del código de software que se somete a las actividades de mantenimiento sucesivos, se hace cada vez más complejo debido al elevado número de cambios. Como resultado deberá tener asignados recursos (humanos y otros) para mantener el software.

Pigoski⁴² observó que a mayor número de cambios durante los primeros tres años de mantenimiento de software, se recomienda que la administración debe asignar más recursos de mantenimiento durante los primeros años en servicio y después de ese tiempo, se requerirá cada vez menos recursos ya que el software se estabiliza y gradualmente caerá en desuso.

3.2. Actividades y Procesos del Mantenimiento de Software

Es importante explicar y describir el alcance de las actividades de mantenimiento de software y el contexto en que los mantenedores trabajan diariamente, Figura 11. De hecho, existen varias interfaces en un contexto organizacional de mantenimiento del software típico en el que el jefe de mantenimiento debe mantener sus aplicaciones funcionando sin problemas. Él tiene que reaccionar rápidamente para restaurar el orden cuando hay problemas de producción. Debe proporcionar el nivel de servicio acordado.

³⁹ Banker RD, Datar SM, Kemerer CF, Zweig D. (1993) Software complexity and maintenance costs. *Communications of the ACM*; 36(11):81–94.

⁴⁰ Boehm, B.W. (1987). Industrial software metrics top 10 list, *IEEE Software*, 4(5), September issue, 84-85.

⁴¹ Lehman, M.M. Belady, L.A. (1985) *Program Evolution – Processes of Software Change*, Publisher: London; Orlando: Academic Press Inc. Ltd. ISBN: 0124424406 0124424414.

⁴² Pigoski T.M. *Practical software maintenance: Best practice for managing your software investment*. John Wiley & Sons: New York, NY, 1997; 384 pp.

Debe mantener la comunidad de usuarios seguros de que tienen un equipo de apoyo dedicado y competente a su disposición, que está actuando dentro del presupuesto acordado.⁴³

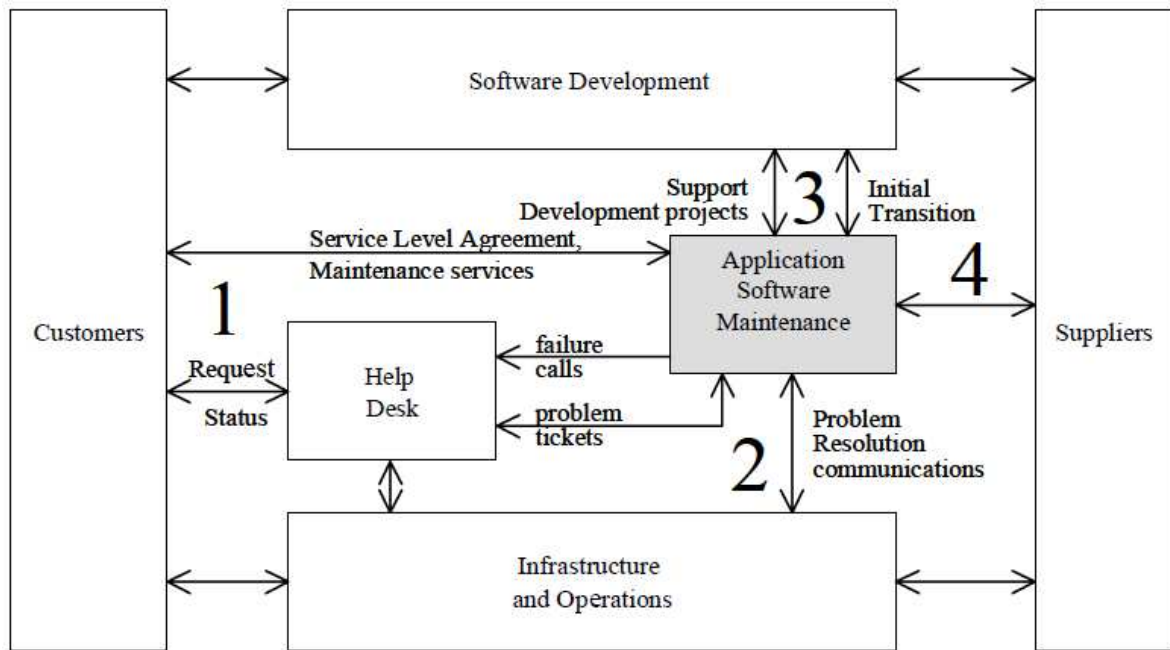


Figura 11. Diagrama de contexto para encargados de mantenimiento. Fuente: Alain A. April, 2004.

La interfaz con el usuario es una función clave y se refiere a las comunicaciones diarias que requieren: a) las respuestas operativas rápidas a los informes de problemas; b) la capacidad de respuesta a las preguntas sobre una regla de negocio, pantalla o informe específico; y c) los informes sobre un gran número de solicitudes de modificación.

Estas interfaces de usuario pueden ser directas o accesibles a través de un servicio de asistencia, y en las mejores prácticas, son apoyadas por un sistema de tratamiento de tickets, que documenta, controla y acelera la carga de trabajo. Otras actividades de la interfaz de usuario, menos intensas y menos frecuentes, consisten en negociaciones y discusiones sobre las prioridades de petición individual, acuerdos de nivel de servicio, planificación, presupuestario, fijación de precios y las actividades relacionadas con la satisfacción del usuario.

⁴³ A.Abran, H. Nguyenkim, (1993), "Measurement of the Maintenance Process from a Demand-based Perspective", *Journal of Software Maintenance: Research and Practice*, 5 (2), 63-90.

Una segunda interfaz de mantenimiento con a) Help Desk; y b) la infraestructura y las operaciones de la organización. El usuario no suele ser consciente de, o participar en, los detalles de los procesos de ingeniería de software. No es consciente de las muchas interacciones diarias entre estas dos organizaciones. Internamente los ingenieros de software deben tener un proceso de resolución de problemas adecuado y comunicaciones eficientes para asegurar la resolución rápida y eficaz de las fallas.

Existe la tercera interfaz clave entre los desarrolladores de software y los mantenedores de software, y se inicia durante el desarrollo de un nuevo producto. La causa raíz de varios problemas de mantenimiento se puede remontar al desarrollo, y se reconoce que los mantenedores deben participar y ejercer algún tipo de control durante esta transición⁴⁴. Esta interfaz de desarrollo de mantenimiento también ilustra las contribuciones hechas por los mantenedores de ayudar y apoyar, y a veces estar involucrado en una serie de grandes proyectos de desarrollo al mismo tiempo. Algunas de las actividades clave serían, por ejemplo: a) el desarrollo de estrategias de transición para reemplazar el software existente; b) ayudar en el diseño de interfaces temporales o nuevas; c) la verificación de las reglas de negocio o ayuda para comprender los datos de los programas informáticos existentes; y d) ayuda para la migración de datos y de corte y cambio de nuevo software o interfaz.

La cuarta interfaz, Figura 11, aborda las relaciones con un número creciente de proveedores, subcontratistas y proveedores. La interfaz de mantenedores con ellos en todo tipo de relaciones, por ejemplo: a) con los proveedores que desarrollan un nuevo software o configuración de software; b) con subcontratistas que son parte del equipo de mantenimiento, para ayudar con conocimientos específicos y mano de obra adicional durante las horas pico; c) con los proveedores de contratos de mantenimiento que prestan servicios de apoyo específicos para su software ya con licencia; y d) con empresas externas que podrían reemplazar parcial o totalmente las funciones de la organización de ingeniería de software (desarrollo, mantenimiento, operaciones e infraestructura). Para garantizar un buen servicio a sus usuarios, mantenedores de software deben desarrollar cierta comprensión de los muchos tipos de contrato, y gestionarlos de manera eficiente,

⁴⁴ S.M. Dekleva. (1992), "Delphi Study of Software Maintenance Problems", ICSM – International Conference on Software Maintenance, IEEE Computer Society Press, 10-17.

para asegurar el desempeño del proveedor, que a menudo afecta a los acuerdos de los resultados a nivel de servicios.

Dependiendo de la fuente de las solicitudes de mantenimiento, las actividades se manejan a través de procesos distintos; esto se ilustra en la Tabla 11 con unos pocos ejemplos. Para cada fuente de solicitud, un servicio/proceso clave de mantenimiento, junto con el debido registro de las categorías de mantenimiento relacionados de trabajo, se inicia. Por ejemplo, si los usuarios son la fuente de las solicitudes, a continuación, una solicitud de cambio en relación con el uso operativo del software y el trabajo que se lleva a cabo se pueden clasificar dentro de uno de los tres servicios de mantenimiento: la corrección, la evolución o el apoyo operacional. En algunos casos, será necesario un proceso de soporte. Un caso típico es la necesidad de información de acuerdo de nivel de servicio como parte de las actividades de apoyo operacional.

Tabla 11. Actividades y categorías de los trabajos de mantenimiento. Fuente: Alain A. April, 2004.

Fuente de Solicitudes	Ejemplo de un Servicio/Proceso clave de Mantenimiento.	Asignación a una categoría de Mantenimiento para el trabajo de mantenimiento.
Jefes de Proyectos	Gestión de la transición desde el desarrollo hasta el mantenimiento.	Apoyo operacional para el proyecto.
Jefes de Proyectos	Proporcionar conocimiento de los sistemas heredados existentes.	Apoyo Operacional para el proyecto.
Usuarios	Pedir un nuevo informe o consulta compleja.	Apoyo Operacional para usuarios.
Usuarios	Solicitar una nueva funcionalidad.	Adaptativo.
Usuarios	Informar un problema operativo.	Correctivo.
Usuarios	Reunión de administración de cuentas trimestrales con los usuarios.	Apoyo operacional para los usuarios y los acuerdos de nivel de servicios.
Operaciones del software	Cambio de una utilidad del sistema.	Perfectivo.
Estudios de renovación	Análisis del impacto de Software.	Si es extremadamente grande, puede ser asignado a mantenimiento preventivo, a menudo supone un proyecto o para el redesarrollo, ambos de los cuales están fuera del alcance de las actividades de pequeños mantenimientos.

Mientras se hace el inventario de las actividades de mantenimiento y revisión de la literatura, se confirmó que algunos de los procesos y actividades de mantenimiento son exclusivos de mantenedores y no están presentes en la función de desarrollo de software, Tabla 12.

Tabla 12. Áreas de proceso clave de mantenimiento de software (P = presente, A = ausente). Fuente: Alain A. April, 2004.

Algunos Procesos Clave de Mantenimiento	Software de gestión (mantenimiento)	Desarrollo de software (creación)
Gestión de problemas (resolución de problemas de interfaz con un Help Desk).	P	A
La aceptación del software.	P	A
La gestión de la transición desde el desarrollo hasta el mantenimiento.	P	A
Acuerdos de Nivel de Servicio.	P	A
Actividades de planificación de Mantenimiento (versiones, acuerdos de nivel de servicio, y el análisis del impacto).	P	A
De eventos y gestión de solicitudes de servicio.	P	A
La administración de software (apoyo operacional).	P	A
Renovación de software.	P	A

3.3. Herramientas Informáticas para el mantenimiento⁴⁵

El uso de herramientas para el mantenimiento del software simplifica las tareas y aumenta la eficiencia y la productividad.

Existen varios criterios para la selección de la herramienta adecuada para la tarea. Estos criterios son la capacidad, características, costo/beneficio, la plataforma, lenguaje de

⁴⁵ Erdil, Finn, Keating, "Software Maintenance As Part of the Software Life Cycle", Department of Computer Science Tufts University, December 16, 2003. URL: http://hepguru.com/maintenance/Final_121603_v6.pdf

programación, facilidad de uso, la apertura de la arquitectura, la estabilidad del proveedor, y la cultura organizacional.

La herramienta debe ser analizada por los beneficios que aporta en contra de su costo. Los indicadores de beneficio de una herramienta son la calidad, la productividad, capacidad de respuesta y reducción de costos. El entorno que la herramienta se ejecuta se llama plataforma. El lenguaje del código fuente se llama lenguaje de programación. Es importante seleccionar una herramienta que apoye un lenguaje estándar de la industria.

La herramienta debe tener una sensación similar a los que los usuarios ya están familiarizados. La herramienta debe tener la capacidad de ser integrada con herramientas de diferentes proveedores. La apertura de la arquitectura juega un papel importante cuando el problema de mantenimiento es complejo. Por lo tanto, no siempre es suficiente utilizar sólo una herramienta, sino varias herramientas funcionando paralelamente.

También es importante considerar la credibilidad del vendedor. El proveedor debe ser capaz de dar soporte la herramienta en el futuro. Si el vendedor no es estable podría quedarse sin negocio y no ser capaz de dar soporte a la herramienta. Otro factor importante es la cultura de la organización. Cada cultura tiene su propio patrón de trabajo. Por lo tanto, es importante tener en cuenta si la herramienta va a ser aceptada por los usuarios.

Existen comercialmente numerosos productos en el mercado disponibles para el mantenimiento del software. Un tipo de producto son las herramientas de seguimiento de errores, que desempeñan un papel importante en el mantenimiento. Bugzilla por Mozilla Foundation es un ejemplo de una herramienta de este tipo. Otros productos de control de errores son Test Director de Mercury Interactive, Silk Radar de Segue Software, SQA Manager de Rational Software, y QA director de Compuware.

Proteus III Expert CMMS de Eagle Technology, Inc. es un paquete de software que permite a los usuarios programar el mantenimiento preventivo, generar órdenes de trabajo automático, historial de mantenimiento de equipos, seguimiento de los activos, inventario, personal, crear órdenes de compra, y generar informes.

Microsoft Visual Source Safe es una herramienta de control del sistema de origen que es utilizada para la administración de configuración.

Productos que son específicos de los lenguajes de programación son CCFinder y JAAT que está diseñado específicamente para los programas Java. CCFinder identifica clones de código en el programa JAVA. JAAT ejecuta análisis de programas Java. Para los programas en C++, existe OCL que es una herramienta de depuración de programas C++ mediante consultas formuladas en el lenguaje de restricción de objetos, OCL.

La tarea de mantenimiento de software se ha convertido en un proceso tan vital y complejo que se requiere soporte automatizado para hacerlo con eficacia. El uso de herramientas simplifica las tareas, la eficiencia y aumento de productividad. Existen numerosas herramientas disponibles en el mercado para el mantenimiento.

MANTOOL es una herramienta para apoyar el proceso de mantenimiento del software y que permite automatizar su gestión de acuerdo a MANTEMA, una metodología rigurosa para el mantenimiento. Este software permite a los usuarios gestionar las solicitudes de modificación siguiendo las diferentes etapas de la metodología. Los datos guardados en MANTOOL se pueden utilizar para extraer diferentes tipos de informes y para hacer estimaciones de futuras intervenciones de mantenimiento.

CAPÍTULO 4

PLANTILLA PARA EL PLAN DE MANTENIMIENTO DE SOFTWARE

4.1. Introducción

En este capítulo se detalla el proceso de mantenimiento de software basado en la norma ISO/IEC 14764. Las fases que engloban el proceso de mantenimiento son:

1. Proceso de implementación.
2. Análisis de modificación y problemas.
3. Implementación de la modificación.
4. Aceptación y revisión del mantenimiento.
5. Migración.
6. Retiro.

Estas fases están detallados en la Figura 12, en la que además se puede observar las actividades con sus respectivas las plantillas descritas en el presente capítulo.

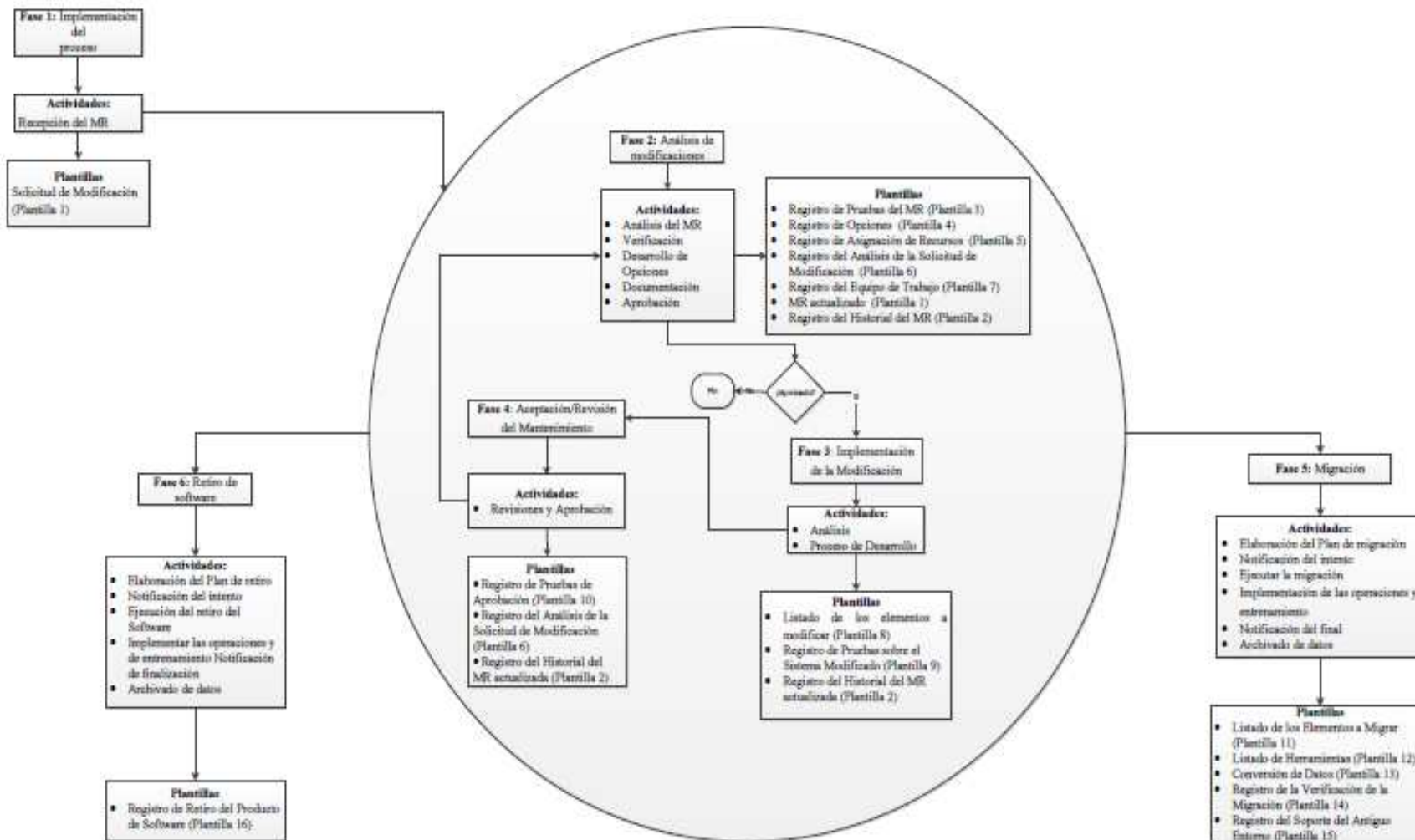


Figura 12. Fases del Proceso de Mantenimiento. Fuente: Autor

4.2. Alcance y Propósito

4.2.1. Alcance

Dar los lineamientos necesarios para realizar un mantenimiento sobre productos de software de una organización.

4.2.2. Propósito

Brindar los criterios y la dirección apropiada para las actividades de mantenimiento en cada una de sus fases. Este plan debe ser citado en el plan de gestión de proyectos de la Dirección de Sistemas.

4.3. Documentos Aplicables

Los documentos necesarios con los que se debe trabajar este plan de mantenimiento son los siguientes:

- **Misión de declaración** (elaborado por el presidente de la organización / CEO)
- **Política de la organización para la mejora continua de procesos** (elaborado por la alta dirección)
- **Manual de proceso de organización de mantenimiento de software** (elaborado por el mantenedor, departamento de mantenimiento de software)
- **Manual de operación de sistemas de gestión de la calidad** (preparado por el director de aseguramiento de la calidad)
- **Norma IEEE 610, 12** (glosario estándar de ingeniería de software)
- **ISO/IEC 14764: 2006** (mantenimiento de software)
- **ISO / IEC 12207: 2008** (los procesos del ciclo de vida del software)

4.4. Definiciones

Definiciones: la terminología utilizada para describir el plan de mantenimiento de software y las definiciones se encuentran en el estándar IEEE 610,12 a 1990, ISO/IEC 14764 e ISO/IEC 12207 y normas de la organización.

Mantenimiento adaptativo: La modificación de un producto de software, realizado después de su entrega, para mantener el producto de software en funcionamiento en un entorno diferente o cambiante.

Mantenimiento correctivo: La modificación de un producto de software realizada después de la entrega para corregir problemas descubiertos.

Mantenimiento de emergencia: Una modificación no programada realizada para mantener temporalmente funcionando un sistema, teniendo pendiente realizar un mantenimiento correctivo.

Mantenibilidad: la capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a los cambios en el entorno y en requisitos y especificaciones funcionales.

Mantenedor: Una organización o persona que realiza actividades de mantenimiento.

Mejora de Mantenimiento: Una mejora de mantenimiento es un cambio en el software que comprende un nuevo requisito.

Plan de mantenimiento: Un documento en el que especifica las prácticas de mantenimiento, los recursos y la secuencia de las actividades pertinentes para dar mantenimiento a un producto de software.

Solicitud de Modificación (MR): Un término genérico que se utiliza para identificar los cambios propuestos a un producto de software.

Mantenimiento Perfectivo: La modificación de un producto de software después de su entrega para detectar y corregir fallas latentes en el software antes de que se manifiestan como fallas.

Mantenimiento Preventivo: La modificación de un producto de software después de su entrega para detectar y corregir fallas latentes en el producto de software antes de que se convierta en fallas operacionales.

Informar de un problema (PR): Un término que se utiliza para identificar y describir los problemas detectados en un producto de software.

Mantenimiento de Software: El mantenimiento del software es el conjunto de actividades necesarias para proporcionar un soporte eficiente a un sistema de software. Dichas actividades se deben llevar a cabo antes y después de la entrega del software. Las actividades previas a la entrega incluyen la planificación de las operaciones posteriores a la entrega, compatibilidad y determinación logística. Las actividades post-entrega incluyen la modificación de software, capacitación y el funcionamiento de un centro de ayuda.

La migración del software: Es el proceso de transición de un producto de software desde un antiguo a un nuevo entorno operativo. Un ejemplo común es la migración desde un entorno mainframe a un entorno cliente / servidor; o un software puede migrar de una tecnología (por ejemplo, archivos y COBOL) a otra (por ejemplo, DBMS relacionales y 4GL).

Baja de un Software: Es el proceso de eliminación de un producto de software en funcionamiento una vez que el producto ha llegado al final de su vida útil.

Transición del software: Una secuencia controlada y coordinada de acciones en el que el desarrollo de software pasa de la organización ejecutante desarrollo inicial del software a la organización de realizar el mantenimiento del software.

4.4.1. Contenido

NOMBRE Y LOGOTIPO DE LA EMPRESA
QUE APLICARA LA GUIA

Plan de mantenimiento de Software

Nombre del programa o producto

Versión

Elaborado por:

Fecha

Figura 13. Portada del Plan de Mantenimiento. Fuente: Autor.

4.4.2. Descripción del sistema

En esta sección se explica el funcionamiento del sistema a darse mantenimiento.

4.5. Fase 1: Proceso de implementación

Entrada:

- Una solicitud de modificación (MR)

Actividad 1: Recepción del MR

Realizar las siguientes tareas:

Tarea 1: Recibir el MR (Plantilla 1) llenado por el solicitante y asignar un número de identificación único.

Tarea 2: Establecer una prioridad de mantenimiento tomando en cuenta la información proporcionada por el solicitante y las políticas establecidas por la organización.

La siguiente tabla es una referencia de prioridades que puede ser utilizada.

.

Tabla 13. *Tabla de prioridades. Fuente: Thomas Pigoski, 2008.*

Tabla de Prioridades	
Prioridad	Se aplica si un problema:
1	<ul style="list-style-type: none"> ✓ Evita la realización de una actividad esencial para el funcionamiento del software. ✓ Pone en peligro la seguridad del software
2	<ul style="list-style-type: none"> ✓ Afectar negativamente a la realización de una actividad esencial de funcionamiento y no se conoce ninguna solución temporal
3	<ul style="list-style-type: none"> ✓ Afectar negativamente a la realización de una actividad esencial para el funcionamiento del software pero existe una solución temporal ✓ Puede resultar inconveniente o molesto para el técnico encargado del sistema pero se conoce una solución temporal.
4	<ul style="list-style-type: none"> ✓ Presenta molestias al usuario pero no afecta la ejecución de actividades esenciales y no impide el cumplimiento de sus responsabilidades.
5	Cualquier otro efecto.

SOLICITUD DE MODIFICACIÓN			
Sección I			
Nombre del solicitante:	Fecha de recepción:	Sistema:	Creador:
Descripción del problema		Prioridad 1 2 3 4 5	
Sección II			
Número de MR:		Mantenedor:	
Prioridad		Tipo de mantenimiento:	
Sección III			
Id opción	Estado del MR	Fecha:	
Resultados del análisis:			
Aprobado por:			

Figura 14. Solicitud de Modificación (Plantilla 1). Fuente: Autor.

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de Implementación del Proceso.

Soporte

La fase de Implementación del Proceso usa los siguientes procesos:

- Documentación
- Aseguramiento de la Calidad
- Revisión Conjunta
- Gestión

Estos procesos deben estar definidos por la organización.

Salida

La salida de ésta fase es:

- MR aceptado

4.6. Fase 2: Análisis de modificaciones

Entradas

Las entradas para la fase de Análisis de Modificaciones son:

- MR
- Documentación del sistema

A continuación el mantenedor procederá a realizar las siguientes actividades:

Actividad 1: Análisis del MR

Tarea 1: Determinar el tipo de mantenimiento a realizarse y registrarlo en el MR, basándose en la Figura 15

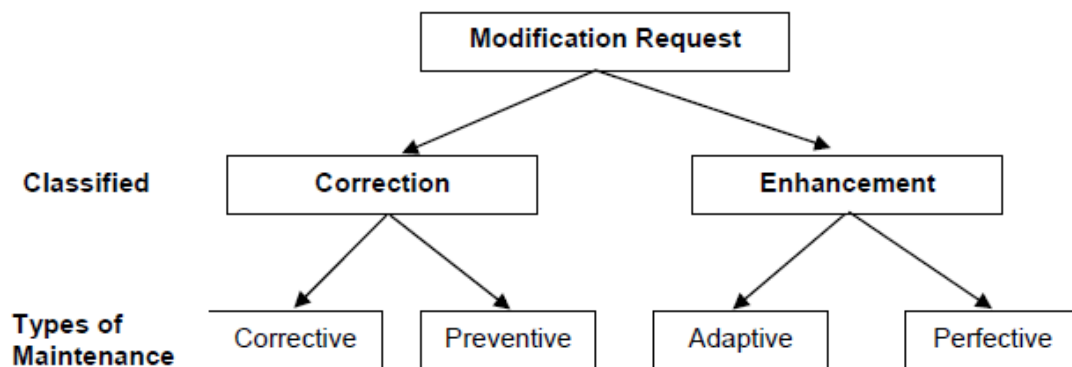


Figura 15. Tipos de Modificaciones. Fuente: Thomas Pigoski, 2008.

Actividad 2: Verificación

Tarea 1: Llenar el Registro del Historial del MR (Plantilla 2). En este se registra la información generada desde que se recibe hasta que se resuelve el MR.

REGISTRO DEL HISTORIAL DEL MR				
ID MR	Fecha de recepción	Fecha de análisis	Estado de MR	Documentos Afectados

Figura 16. Registro del Historial del MR (Plantilla 2). Fuente: Autor.

Tarea 2: Obtener el software afectado e instalarlo.

Tarea 3: Realizar pruebas sobre el software para comprobar la falla reportada y documentar los resultados obtenidos en el Registro de Pruebas del MR (Plantilla 3).

REGISTRO DE PRUEBAS DEL MR		
ID prueba:	ID MR:	Nombre del Sistema:
Responsable de la Prueba:		
Objetivo:		
Resultado Obtenido:		

Figura 17. Registro de Pruebas del MR (Plantilla 3). Fuente: Autor.

Actividad 3: Desarrollo de Opciones

Después de realizar las pruebas, el mantenedor deberá:

Tarea 1: Asignar una prioridad al MR. Ésta prioridad dependerá de la política de la organización y se registrara en el MR. A diferencia de las prioridades de la fase 1 la asignación de esta prioridad depende de quien la entrega.

Tarea 2: Definir los requerimientos de la organización y proponer al menos tres opciones para realizar la modificación.

Tarea 3: De cada opción se deberá hacer una estimación de extensión y magnitud de la modificación, los impactos que tendrán en el hardware del sistema y un análisis de los riesgos que se pueden presentar.

Tarea 4: El mantenedor recomienda cuál sería la opción más viable. Todas estas tareas se deberán registrar en la plantilla 4.

REGISTRO DE OPCIONES				
ID MR:	Nombre del sistema:		Responsable:	
Requerimientos de la organización :				
ID Opción	Descripción	Extensión	Impacto	Riesgos
Recomendación:				

Figura 18. Registro de Opciones (Plantilla 4). Fuente: Autor.

La extensión, el impacto y los riesgos se determinaran de acuerdo a las políticas de la organización. En algunos casos se puede asignar valores numéricos a estos campos facilitando así el análisis de las opciones presentadas.

Tarea 5: Estimar los recursos humanos y de costo que requiere cada opción y documentarlos en el Registro de Asignación de Recursos (Plantilla 5). Esta estimación también puede influir en la elección de la solución a implementar.

REGISTRO DE ASIGNACIÓN DE RECURSOS				
Id MR:	Nombre del sistema:		Responsable:	
Id Opción	Nº Personas	Nº Horas	Costo	Observación

Figura 19. Registro de Asignación de Recursos (Plantilla 5). Fuente: Autor.

Actividad 4: Documentación

Tarea 1: Verificar que todas las pruebas y opciones propuestas estén debidamente documentadas en las plantillas y actualizar el Registro del Historial del MR (Plantilla 2).

Actividad 5: Aprobación

Antes de realizar la modificación al sistema el mantenedor debería obtener la aprobación del MR. Para esto se deberá realizar las siguientes tareas.

Tarea1: Presentar el MR, Registro de Pruebas del MR, Registro de Opciones y Registro de Asignación de Recursos, para su análisis por parte de la Dirección de Sistemas.

Tarea 2: El mantenedor participara en las discusiones acerca de la modificación.

Tarea 3: Documentar el resultado del análisis de MR en el Registro del Análisis de la Solicitud de Modificación (Plantilla 6). Además se deberá actualizar la Sección II del MR (Plantilla 1) y el Registro del Historial del MR (Plantilla 2). Si el MR fue aprobado se continuara con el resto de actividades, sin embargo si el MR fue negado se registrará el motivo y el proceso de mantenimiento terminará.

REGISTRO DEL ANÁLISIS DE LA SOLICITUD DE MODIFICACIÓN			
Nombre del Sistema:			
Responsable(s):			
Fecha de recepción del MR:			
Fecha de aprobación/negado:			
Fecha de cierre:			
ID MR	Descripción de la solución	Estado del MR	Motivo del rechazo
		Aprobado Negado	
Responsable del Análisis:			

Figura 20. Registro del Análisis de la Solicitud de Modificación (Plantilla 6). Fuente: Autor.

Tarea 4: Detallar las personas que intervendrán en la modificación del sistema y el rol que desempeñaran y documentarlo en el Registro del Equipo de Trabajo (Plantilla 7).

REGISTRO DEL EQUIPO DE TRABAJO			
Nombre del Sistema:		Id MR:	
Mantenedor:			
Detalle del Mantenimiento	Nombre y Apellido	Rol	Observación

Figura 21. Registro del Equipo de Trabajo (Plantilla 7). Fuente: Autor.

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de Análisis de modificaciones.

Soporte

La Fase de Análisis de Modificaciones usa los siguientes procesos:

- Documentación
- Aseguramiento de la Calidad
- Información acerca de los Problemas

Estos procesos deben estar definidos por la organización.

Salidas

Las salidas de ésta fase son:

- Registro de Pruebas del MR
- Registro de Opciones
- Registro de Asignación de Recursos
- Registro del Análisis de la Solicitud de Modificación
- Registro del Equipo de Trabajo
- MR actualizado
- Registro del Historial del MR actualizado

4.7. Fase 3: Implementación de la Modificación

En esta fase el mantenedor desarrolla y prueba la modificación del producto de software.

Entradas:

Las entradas a la actividad de Implementación de la Modificación son:

- Salidas de la Fase de Análisis de Modificación
- Código fuente

Actividad 1: Análisis

El mantenedor y su equipo deberán realizar las siguientes tareas:

Tarea 1: Se identificarán de forma detallada los elementos del sistema que serán afectados por la modificación y quién será el responsable de realizarla. Todo esto deberá ser documentado en el Listado de los Elementos a Modificar (Plantilla 8)

LISTADO DE LOS ELEMENTOS A MODIFICAR					
Nombre del sistema:					
Mantenedor:					
Equipo de Mantenimiento					
N°	Elemento a ser modificado	Capa de la arquitectura	Descripción de la modificación a realizar	Responsable de la modificación	Duración

Figura 22. Listado de los Elementos a Modificar (Plantilla 8). Fuente: Autor.

Actividad 2: Proceso de Desarrollo

El mantenedor y su equipo deberán realizar las siguientes tareas:

Tarea 1: Desarrollar la modificación.

Tarea 2: Documentar y definir criterios de pruebas para la comprobación y evaluación de las partes modificadas y no modificadas. En el Registro de Pruebas sobre el Sistema Modificado (Plantilla 9) se deberá especificar el tipo de pruebas que se ejecutaran sobre el sistema modificado, estas pruebas pueden ser:

- ✓ Pruebas unitarias
- ✓ Pruebas de integración
- ✓ Pruebas de aceptación
- ✓ Pruebas de estrés
- ✓ Pruebas funcionales

El tipo de prueba se seleccionara basado en como la modificación realizada afecta al sistema, esto permitirá asegurar que el sistema esté funcionando correctamente después del cambio ejecutado. Los resultados de las pruebas deberían documentarse

en el Registro de Pruebas sobre el Sistema Modificado (Plantilla 9). Además se deberá actualizar Registro del historial del MR (Plantilla 2).

REGISTRO DE PRUEBAS SOBRE EL SISTEMA MODIFICADO				
Id MR:		Id Prueba:		Fecha de ejecución:
Nombre del sistema:				
Tipo de Prueba:				
Objetivo:				
Descripción:				
Responsable:				
Elemento a Probar	Precondición	Datos de Entrada	Resultados esperados	Resultados obtenidos

Figura 23. Listado de los Elementos a Modificar (Plantilla 9). Fuente: Autor.

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de Implementación de la Modificación.

Soporte

La Fase de Implementación de la Modificación utiliza los siguientes procesos:

- Documentación
- Aseguramiento de la Calidad
- Revisión Conjunta

Estos procesos deben estar definidas por la organización.

Salidas

Las salidas de ésta actividad deberían incluir:

- Listado de los elementos a modificar
- Código Fuente modificado
- Registro de Pruebas sobre el Sistema Modificado
- Registro del Historial del MR actualizada

4.7. Fase 4: Aceptación/Revisión del Mantenimiento

En esta fase se confirma que las modificaciones realizadas al sistema hayan sido ejecutadas correctamente.

Entradas

Las entradas a la actividad de Aceptación/Revisión del mantenimiento son:

- El Software Modificado
- Registro de Pruebas sobre el Sistema Modificado

Actividad 1: Revisiones y Aprobación

Tarea 1: El mantenedor deberá reunirse con el Coordinador de Desarrollo de la Dirección de Sistemas y realizar pruebas sobre el sistema para comprobar que la modificación se ha realizado adecuadamente.

Tarea 2: Documentar las pruebas realizadas en el Registro de Pruebas de Aprobación. Si las pruebas fueron exitosas se registrara un estado de aprobado, caso contrario se realizaran las observaciones necesarias y se retornara a la Fase 3

REGISTRO DE PRUEBAS DE APROBACIÓN						
Id MR:		Id Prueba:		Fecha de ejecución:		
Nombre del sistema:						
Tipo de Prueba:						
Objetivo:						
Descripción:						
Responsables:						
Elemento a Probar	Precondición	Datos de Entrada	Resultados esperados	Resultados obtenidos	Estado	Observaciones

Figura 24. Registro de Pruebas de Aprobación (Plantilla 10). Fuente: Autor.

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de Aceptación/Revisión del Mantenimiento.

Soporte

La Fase de Aceptación/Revisión del Mantenimiento usa los siguientes procesos:

- Aseguramiento de la Calidad
- Verificación
- Validación
- Revisión Conjunta
- Auditoría

Estos procesos deben estar definidos por la organización.

Salidas

Las salidas de ésta actividad son:

- Sistema modificado
- Registro de Pruebas de Aprobación
- Registro del Análisis de la Solicitud de Modificación

- Registro del Historial del MR actualizada
- Informes de revisión y auditoría

4.8. Fase 5: Migración

Durante el ciclo de vida de un sistema, puede que éste deba ser modificado para ejecutarlo en entornos diferentes. Para migrar un sistema a un nuevo entorno, el mantenedor necesita determinar las acciones necesarias para conseguir la migración y a partir de ahí desarrollar y documentar los pasos necesarios para efectuar la migración; el proceso de migración seguirá el mismo flujo que un MR normal pero la fase de implementación será reemplazada por la fase de migración, esto puede entenderse mejor observando la Figura 25.

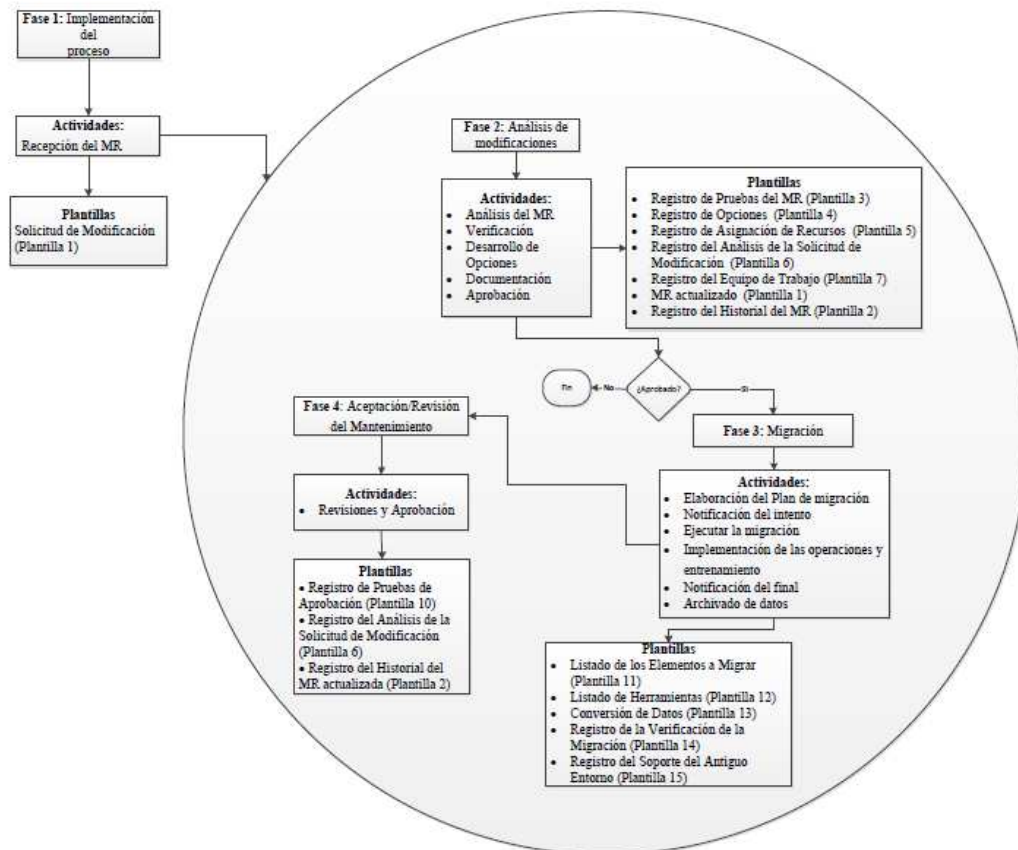


Figura 25. Proceso de Migración (Plantilla 10). Fuente: Autor.

Entradas

Las entradas a la actividad de Migración son:

- El Antiguo Entorno
- El Nuevo Entorno
- La Antigua Línea Base
- La Nueva Línea Base

Actividad 1: Elaboración del Plan de migración

Durante la migración de un sistema el mantenedor estará encargado de crear un plan de migración, documentarlo y ejecutarlo. Las actividades de planificación deberían incluir:

Tarea 1: Análisis de requerimientos y definición de la migración. El mantenedor analizará la petición de migración y valorará la factibilidad de los cambios a ser realizados.

LISTADO DE LOS ELEMENTOS A MIGRAR					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
N°	Elemento a ser migrado origen	Descripción elemento origen	Elemento a ser migrado destino	Descripción elemento destino	Responsable de la migración

Figura 26. Listado de los Elementos a Migrar (Plantilla 11). Fuente: Autor.

Tarea 2: Desarrollo de herramientas de ayuda a la migración. El mantenedor en base a los elementos definidos en la tarea 1 determinará las herramientas necesarias para llevar a cabo la migración. Se deberá llevar un registro de estas herramientas en la siguiente plantilla.

LISTADO DE HERRAMIENTAS					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
N°	Nombre	Tipo	Descripción	Elemento	Responsable

Figura 27. Listado de Herramientas (Plantilla 12). Fuente: Autor.

Tarea3: Conversión de datos. La mayor cantidad de información dentro de un producto de software suele encontrarse en una base de datos, las actividades de migración pueden representar modificación, consolidación o segregación de estos datos, el mantenedor deberá registrar estas alteraciones en la siguiente plantilla.

CONVERSIÓN DE DATOS				
Nombre del sistema:				
Mantenedor:				
Equipo de Migración:				
Elemento a migrar:				
Origen				
Código	Tablas	Columnas	Tipo de dato	Descripción
Destino				
Código	Tablas	Columnas	Tipo de dato	Descripción

Figura 28. Listado de Herramientas (Plantilla 13). Fuente: Autor.

Los datos que no se encuentren dentro de una base de datos deberán ser registrados como complemento a la plantilla anterior.

Actividad 2: Notificación del intento

Después de desarrollar el Plan de Migración se debe notificar a los usuarios de los planes y actividades de migración.

Esta notificación deberá incluir:

- Detallar las razones de ya no dar soporte al antiguo entorno
- Describir el nuevo entorno e indicar la fecha de arranque y disponibilidad
- Descripción de las nuevas opciones de soporte.

Además, el mantenedor proporcionara a los usuarios los procedimientos, y la planificación de la migración. Para esto el mantenedor debería llevar a cabo los siguientes pasos:

- Listar los puestos afectados por la migración
- Obtener la realimentación del puesto afectado
- Identificar aspectos específicos del puesto
- Socializar la planificación

Actividad 3: Ejecutar la migración

Tarea 1: Ejecución de la migración. En esta tarea es en donde el mantenedor realizara la migración.

Tarea 2: Verificación de la migración. Se realizaran las pruebas correspondientes a cada elemento a ser migrado y su registro individual se realizara utilizando la Plantilla 9 (que pertenece a la fase de implementación), los resultados generales de la verificación serán registrados en la siguiente plantilla.

REGISTRO DE LA VERIFICACIÓN DE LA MIGRACIÓN				
Id MR:		Fecha de ejecución:		
Nombre del sistema:				
Mantenedor:				
Equipo de migración:				
Elemento migrado	Resultado esperado de la migración	Resultado obtenido de la migración	Aprobado (si/no)	Fecha

Figura 29. Registro de la Verificación de la Migración (Plantilla 14). Fuente: Autor.

Tarea 5: Soporte para el antiguo entorno. Tras finalizar la migración se indicara el tipo de soporte o mantenimiento que se dará al sistema origen, este soporte puede ser dado a todo el sistema o por cada uno de los elementos que lo conforman, para ello se utilizara la siguiente plantilla:

REGISTRO DEL SOPORTE DEL ANTIGUO ENTORNO			
Nombre del sistema:			
Mantenedor:			
Equipo de Migración:			
Elementos migrados			
Elemento	Tipo de Soporte	Descripción de Soporte	Observación
Elementos no migrados			
Elemento	Tipo de Soporte	Descripción de Soporte	Observación

Figura 30. Registro del Soporte del Antiguo Entorno (Plantilla 15). Fuente: Autor.

Todo el proceso de migración requiere de adecuado control de los riesgos que estos representen para la organización, se deben determinar estos y buscar una estrategia para su mitigación. Por lo general, todo suele recaer en la ejecución paralela de los dos productos hasta que se comprueben el correcto funcionamiento del producto destino de la migración.

Actividad 4: Implementación de las operaciones y entrenamiento

Para facilitar al usuario la transición del antiguo entorno con el nuevo, el mantenedor deberá realizar los siguientes pasos:

- Mantener un puesto en su configuración original
- Instalar el hardware y el software
- Ejecutar pruebas preliminares para garantizar la correcta instalación del hardware y el software
- Ejecutar el software con una carga operativa en el entorno antiguo y en el nuevo
- Recolectar datos de los productos nuevos y viejos
- Analizar los datos

Además el mantenedor deberá proporcionar un entrenamiento a los usuarios para lo cual deberá realizar los siguientes pasos:

- Identificar y planificar los requisitos de entrenamiento
- Dirigir la revisión del entrenamiento
- Actualizar los planes de entrenamiento

Actividad 5: Notificación del final de la Migración

Al finalizar la migración se debe notificar a los usuarios involucrados.

Revisión post-Migración

Después de realizada la migración es importante determinar el impacto que ha generado el cambio a un nuevo sistema. Adicional a esto se debe notificar los resultados a las autoridades para su información, guía y actuación.

Para esto el mantenedor deberá:

- Revisar los resultados al operar paralelamente ambos entornos
- Localizar las áreas con un riesgo potencial
- Identificar aspectos específicos del puesto de trabajo
- Realizar un informe sobre el Análisis del Impacto

Actividad 6: Archivado de datos

Es importante para la organización mantener los datos, usados por el antiguo entorno, accesibles para lo cual se deberá realizar las siguientes tareas:

- Almacenar los datos y el software antiguo.
- Hacer copias de respaldo de los datos y del software antiguo.
- Almacenar las copias de respaldo en un lugar seguro.

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de migración.

Soporte

La actividad de Migración usa los siguientes procesos:

- Documentación
- Gestión de la Configuración
- Aseguramiento de la Calidad
- Verificación
- Validación
- Revisión Conjunta
- Auditoría
- Informe sobre problemas
- Entrenamiento

Salidas

Las salidas de ésta actividad son:

- Plan de Migración

- Herramientas de Migración
- Notificación de Intentos
- Producto Software Migrado
- Notificación de Finalización
- Datos archivados

4.9. Fase 6: Retiro del software

Esta fase se realiza cuando un software ha llegado al final de su vida útil. Para esto la organización deberá realizar un análisis basado en el costo que generaría:

- Conservar software antiguo
- Cambiar a una nueva tecnología desarrollando un nuevo software
- Desarrollar un nuevo software para conseguir modularidad
- Desarrollar un nuevo software para facilitar el mantenimiento
- Desarrollar un nuevo software para lograr la estandarización
- Desarrollar un nuevo software para alcanzar la independencia del fabricante

Al igual que en la fase anterior el retiro será tratado como un MR normal pudiendo reemplazar las actividades de la implementación por las del retiro, esto se observa en la Figura 31

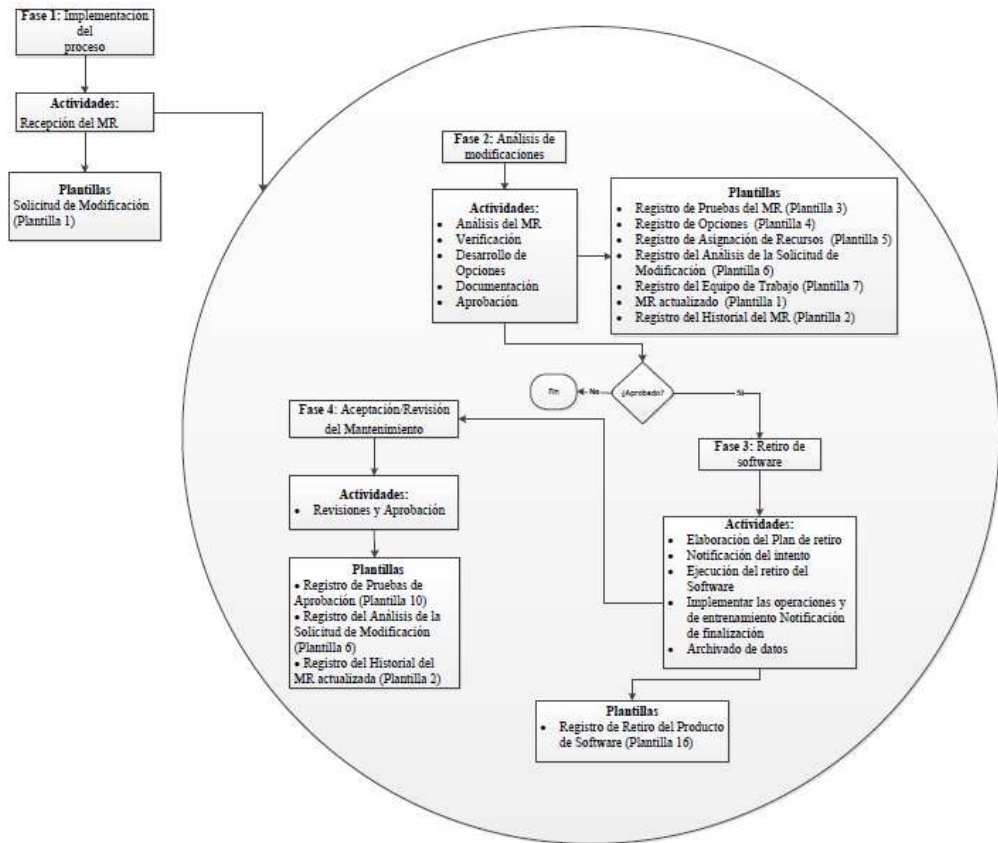


Figura 31. *Proceso del Retiro del Software. Fuente: Autor*

El producto de software podría reemplazarse por uno nuevo pero no siempre. Para retirar un producto de software, el mantenedor debería determinar las acciones necesarias para conseguir el retiro, y entonces desarrollar y documentar los pasos necesarios para efectuar el retiro junto con sus datos almacenados.

Entradas

Las entradas a la fase de retiro son:

- El producto software a retirar
- El nuevo producto software
- El antiguo entorno

Actividad 1: Elaboración del Plan de retiro

Se debería desarrollar y documentar un plan de retiro para eliminar el soporte por parte de las organizaciones que operan con el sistema y lo mantienen. Las actividades de planificación deberían incluir a los usuarios. El plan debería tener en cuenta los siguientes aspectos:

- Fin del soporte total o parcial
- Archivar software y su documentación
- Determinar el responsable del soporte de datos residuales
- Transición al nuevo producto software
- Accesibilidad a las copias archivadas de los datos

Como se puede observar los aspectos indicados coinciden con los de la fase de migración, esta similitud se presenta ya que se puede considerar al proceso de retiro como una migración en la cual no hay un producto nuevo.

Como parte de ésta tarea, el mantenedor debería:

- Analizar los requerimientos de retiro
- Determinar el impacto del retiro del producto de software
- Identificar claramente el producto de software a reemplazar, si lo hay (Es recomendable tratar esta actividad como un proceso de migración)
- Establecer una planificación para el retiro del producto software
- Identificar los responsables del soporte residual futuro
- Definir y documentar el esfuerzo de retiro

Todas estas tareas pueden ser registradas en la siguiente plantilla.

REGISTRO DE RETIRO DEL PRODUCTO DE SOFTWARE					
Nombre del sistema:					
Mantenedor:					
Equipo de Retiro:					
Elementos retirados					
Elemento	Tipo de Soporte residual	Descripción de Soporte residual	Impacto	Responsable	Tiempo

Figura 32. Registro de Retiro del Producto de Software (Plantilla 16). Fuente: Autor

Al igual que en el proceso de migración se debe llevar a cabo las siguientes actividades:

- Notificación del intento
- Implementar las operaciones y de entrenamiento (Esta actividad es necesario si el producto software retirado va a ser reemplazado por una nueva)
- Notificación de finalización
- Archivado de datos

Controles

Las revisiones conjuntas entre el mantenedor y el Director de Sistemas se deberían usar para controlar las salidas de la Fase de Retiro del Software.

Soporte

La Fase de Retiro del Software usa los siguientes procesos:

- Documentación
- Gestión de la Configuración
- Aseguramiento de la Calidad
- Revisión Conjunta
- Auditoría

- Entrenamiento

Salidas

Las salidas de ésta fase son:

- Plan de Retiro
- Notificación de Intento
- Resultados del Retiro
- Personas entrenadas
- Producto Software Retirado
- Notificación de finalización
- Línea base del producto retirado archivada.

CONCLUSIONES Y RECOMENDACIONES

El plan de mantenimiento de software propuesto contribuye de manera directa a hacer frente a los problemas de calidad en la función de mantenimiento del Departamento de Sistemas de la Universidad Politécnica Salesiana, sugiriendo una metodología de proceso en base de la norma ISO/IEC 14764, y con esto la mejora continua en la gestión de mantenimiento en requerimientos futuros.

Debido a la complejidad de las organizaciones el mantenedor y su equipo debe tener claras las actividades de mantenimiento y las interfaces (comunicaciones con otras personas y departamentos) con las que trabaja dentro de la organización, por ejemplo: clientes, usuarios, desarrolladores, proveedores, departamentos internos. Las interfaces estarán constantemente solicitando al departamento mantenimientos de diferente prioridad, por lo que el análisis del mantenedor para realizar mantenimientos o presentar otras soluciones, debe estar acorde al presupuesto asignado y/o como afectara este cambio a la producción en general de la organización.

La fase 5 “migración” y fase 6 “retiro del software” del proceso de mantenimiento, se realiza siguiendo el mismo proceso del plan de mantenimiento. Si bien, la migración y el retiro del software son fases, la norma las presenta como un proceso más dentro del ciclo de vida software y requieren un análisis paralelo con el plan de mantenimiento para su realización utilizando la norma ISO/IEC 12207.

La interpretación de la norma ISO/IEC 14746 en el proceso de generación de este plan de mantenimiento lograra resultados, calidad, y una productividad en base de las iteraciones y de los mantenimientos que se vayan produciendo. A lo anterior, se puede añadir analizar cuáles son los efectos de los cambios realizados en el mantenimiento sobre personal y aplicaciones involucradas durante el paso del tiempo.

Dados los enormes esfuerzos y costos dedicados al mantenimiento de software, toda empresa debe evaluar y considerar las mejores prácticas para el mantenimiento. Es así, que la creación de una sección/departamento de mantenimiento o contratar el mantenimiento es un tema de análisis para UPS.

Sobre la base del presente trabajo se recomienda involucrar para otros temas de tesis los siguientes aspectos:

- Inclusión de la norma ISO/IEC 9126 de calidad de software, en el desarrollo de nuevo software con énfasis en la mantenibilidad posterior.
- Realizar las valoraciones de métricas del software existente e involucrarlo en el proceso de mantenimiento.
- Selección de herramientas de mantenimiento en el mercado que ayuden al proceso diseñado.
- Elaboración de un programa que contenga el plan de mantenimiento para registro de información.
- Realizar una auditoría sobre los datos generados de diferentes mantenimientos, con el fin de evaluar el nivel de madurez del plan de mantenimiento.
- Departamento/Generador de pruebas como agente de validación de cambios.
- Análisis financiero sobre la creación de la sección/departamento de mantenimiento de software en el departamento de sistemas de la UPS.

REFERENCIAS BIBLIOGRÁFICAS

Zvegintzov Nicholas & Parikh Girish, “60 years of Software Maintenance: Lessons Learned”, Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on Software Maintenance, Page(s): 726 - 727.

14764 - International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering -Software Life Cycle Processes – Maintenance.

ISO/IEC 12207:2002. AMENDMENT 1: Information Technology - Software Life Cycle Processes Amendment 1.

R. S. Pressman, Ingeniería de Software, Séptima ed., New York: Mac Graw Hill, 2010.

IEEE Std 1219-1998, IEEE Standard for Software Maintenance, IEEE, 1998.

Estándares para el mantenimiento del software – EcuRed, marzo 2015.
http://www.ecured.cu/index.php/Est%C3%A1ndares_para_el_mantenimiento_del_software

Macario Polo, Mario Piattini, Francisco Ruiz, and Coral Calero. MANTEMA: A Complete Rigorous Methodology for Supporting Maintenance Based On The ISO/IEC 12207 Standard. In Proc. of the 3rd European Conference on Software Maintenance and Reengineering (CSMR), pages 178–181, Amsterdam (The Netherlands), March 1999.

Pino, F., Ruiz, F., Salas, S.: Agil mantema, technical report, Instituto de Tecnologías de Información y sistemas, Universidad de Castilla-La Mancha, (2008).

Balseca, Caiza, “ELABORACIÓN DE UNA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE”, Tesis, Escuela Politécnica Nacional, Quito, 2014

Erdil, Finn, Keating, “Software Maintenance As Part of the Software Life Cycle”, Department of Computer Science Tufts University, December 16, 2003. URL: http://hepguru.com/maintenance/Final_121603_v6.pdf

ZELKOVITMZ., V ., CHAW,A . C. y GANNON,J . D.:F Principles of software Engineering and Design. Prentice-Hall, Englewoods Ciif, 1979.

BOEHM, B. W.: Software Engineering., IEEE Transactions on Computers, C-25, núm. 12, diciembre, pp. 1226.1241.

BAUER, F. L.: “Software Engineering”, Information Processing, 71, North Holland Publishing Co., Amsterdarn, 1972

IEEE: Standards Collection: Software Engineering, IEEE Standard 610.12-1990, IEEE, 1993.

Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología. URL: <http://ingenieria.uatx.mx/labastida/files/2011/08/MANTENIMIENTO-DE-SOFTWARE.pdf>

Computer Society, marzo 2015. URL: <http://www.computer.org/portal/web/swebok/html/ch6>

Ruiz y Polo, “Mantenimiento del Software”, Grupo Alarcos, Dep. de Informática, Escuela Superior de Informática, Universidad de Castilla - La Mancha, Ciudad Real, 2001. URL: <http://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/s1.pdf>

Priyadarshi Tripathy & Kshirasagar Naik, “Software Evolution and Maintenance”, Wiley, USA, 2014.

Nancy Leveson and Clark S. Turner. An Investigation of the Therac-25 Accidents. URL: http://courses.cs.vt.edu/cs3604/lib/Therac_25/Therac_1.html.

Alan Dix. The Ariane 5 Launcher Failure. URL:
http://www.comp.lancs.ac.uk/computing/research/cseg/projects/tracker/sss_iceis04.pdf

Tony Long. Sept. 26, 1983: The Man Who Saved the World by Doing ... Nothing. URL:
http://www.wired.com/science/discoveries/news/2007/09/dayintech_0926

Pérez-Castillo, García, García Rodríguez de Guzmán y Piattini, “La Reingeniería como tópico en la docencia de la Ingeniería del Software: una Experiencia Práctica”, Instituto de tecnologías y Sistemas de Información, Universidad de Castilla – La Mancha, España, 2012.

Alain A. April, “SMmm Model to Evaluate and Improve the Quality of Software Maintenance Process”, Institut für Verteilte Systeme Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg Postfach 4120 39016 Magdeburg, Germany, 2004.

Foster, J.R., Munro, M. (1987) A Documentation Method Based on Cross-Referencing, Proceedings of the IEEE Conference on Software Maintenance, IEEE Computer Society Press, Los Alamitos, California, pp 181-185.

Hall, P.V.A. (1987) Software components and reuse: Getting more out of your code. Information Software Technology, Vol. 29, No. 1, Feb, pp. 38-43.

Scott, T. Farley, T. (1988) Slashing Software Maintenance Costs, Business Software Review, Indianapolis, March.

Hanna, M. (1993) Maintenance Burden Begging for a Remedy, Datamation, April, pp.53-63.

Fornell, G.E. (1992) cover letter to report, “Process for Acquiring Software Architecture,” July 10, http://www.stsc.hill.af.mil/resources/tech_docs/gsam2/chap_2.DOC (link tested May 11th 2004).

Banker RD, Datar SM, Kemerer CF, Zweig D. (1993) Software complexity and maintenance costs.

Communications of the ACM; 36(11):81–94.

Boehm, B.W. (1987). Industrial software metrics top 10 list, IEEE Software, 4(5), September issue, 84-85.

Lehman, M.M. Belady, L.A. (1985) Program Evolution – Processes of Software Change, Publisher: London; Orlando: Academic Press Inc. Ltd. ISBN: 0124424406 0124424414.

Pigoski T.M. Practical software maintenance: Best practice for managing your software investment. John Wiley & Sons: New York, NY, 1997; 384 pp.

A. Abran, H. Nguyenkim, (1993), "Measurement of the Maintenance Process from a Demand-based Perspective", Journal of Software Maintenance: Research and Practice, 5 (2), 63-90.

S.M. Dekleva. (1992), "Delphi Study of Software Maintenance Problems", ICSM – International Conference on Software Maintenance, IEEE Computer Society Press, 10-17.

ANEXOS

**ANEXO A. PLANTILLAS PARA EL PLAN DE MANTENIMIENTO DE
SOFTWARE DE LA UNIVERSIDAD POLITÉCNICA SALESIANA**

**UNIVERSIDAD POLITÉCNICA
SALESIANA**


Plan de mantenimiento de Software

Nombre del programa o producto

Versión

Elaborado por:


Fecha


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
SOLICITUD DE MODIFICACIÓN					
Sección I					
Nombre del solicitante:		Fecha de recepción:	Sistema:		Creador:
Descripción del problema			Prioridad 1 2 3 4 5		
Sección II					
Número de MR:			Mantenedor:		
Prioridad			Tipo de mantenimiento:		
Sección III					
Id opción	Estado del MR		Fecha:		
Resultados del análisis:					
Aprobado por:					

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
REGISTRO DE HISTORIALES DEL MR					
ID MR	Fecha de recepción	Fecha de análisis	Estado de MR	Documentos Afectados	

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
REGISTRO DE PRUEBAS DEL MR					
ID MR:			Nombre del Sistema:		
Responsable de la Prueba:					
Objetivo:					
Resultado Obtenido:					


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código
REGISTRO DE OPCIONES				
ID MR:	Nombre del sistema:		Responsable:	
Requerimientos de la organización :				
ID Opción	Descripción	Extensión	Impacto	Riesgos
Recomendación:				


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código
REGISTRO DE ASIGNACIÓN DE RECURSOS				
Id MR:	Nombre del sistema:		Responsable:	
Id Opción	N° Personas	N° Horas	Costo	Observación


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código
REGISTRO DEL ANÁLISIS DE LA SOLICITUD DE MODIFICACIÓN				
Nombre del Sistema:				
Responsable(s):				
Fecha de recepción del MR:				
Fecha de aprobación/negado:				
Fecha de cierre:				
ID MR	Descripción de la solución	Estado del MR	Motivo del rechazo	
		Aprobado Negado		
Responsable del Análisis:				


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
REGISTRO DEL EQUIPO DE TRABAJO					
Nombre del Sistema:			Id MR:		
Mantenedor:					
Detalle del Mantenimiento		Nombre y Apellido		Rol	Observación


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software			Código	
LISTADO DE LOS ELEMENTOS A MODIFICAR						
Nombre del sistema:						
Mantenedor:						
Equipo de Mantenimiento						
N°	Elemento a ser modificado	Capa de la arquitectura	Descripción de la modificación a realizar	Responsable de la modificación	Duración	

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
REGISTRO DE PRUEBAS SOBRE EL SISTEMA MODIFICADO					
Id MR:		Id Prueba:		Fecha de ejecución:	
Nombre del sistema:					
Tipo de Prueba:					
Objetivo:					
Descripción:					
Responsable:					
Elemento a Probar	Precondición	Datos de Entrada	Resultados esperados	Resultados obtenidos	


		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software			Código	
REGISTRO DE PRUEBAS DE APROBACIÓN						
Id MR:		Id Prueba:		Fecha de ejecución:		
Nombre del sistema:						
Tipo de Prueba:						
Objetivo:						
Descripción:						
Responsables:						
Elemento a Probar	Precondición	Datos de Entrada	Resultados esperados	Resultados obtenidos	Estado	Observaciones

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
LISTADO DE LOS ELEMENTOS A MIGRAR					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
N°	Elemento a ser migrado origen	Descripción elemento origen	Elemento a ser migrado destino	Descripción elemento destino	Responsable de la migración

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
LISTADO DE HERRAMIENTAS					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
N°	Nombre	Tipo	Descripción	Elemento	Responsable

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software			Código
CONVERSIÓN DE DATOS					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
Elemento a migrar:					
Origen					
Código	Tablas	Columnas	Tipo de dato	Descripción	
Destino					
Código	Tablas	Columnas	Tipo de dato	Descripción	

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código
REGISTRO DE LA VERIFICACIÓN DE LA MIGRACIÓN				
Id MR:		Fecha de ejecución:		
Nombre del sistema:				
Mantenedor:				
Equipo de migración:				
Elemento migrado	Resultado esperado de la migración	Resultado obtenido de la migración	Aprobado (si/no)	Fecha

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software		Código	
REGISTRO DEL SOPORTE DEL ANTIGUO ENTORNO					
Nombre del sistema:					
Mantenedor:					
Equipo de Migración:					
Elementos migrados					
Elemento		Tipo de Soporte	Descripción de Soporte		Observación
Elementos no migrados					
Elemento		Tipo de Soporte	Descripción de Soporte		Observación

		Universidad Politécnica Salesiana Dirección de Sistemas Plan de Mantenimiento de Software			Código	
REGISTRO DE RETIRO DEL PRODUCTO DE SOFTWARE						
Nombre del sistema:						
Mantenedor:						
Equipo de Retiro:						
Elementos retirados						
Elemento	Tipo de Soporte residual	Descripción de Soporte residual	Impacto	Responsable	Tiempo	