

Universidad de las Ciencias Informáticas



Tesis para optar por el Título académico de Máster en
Informática Aplicada

Título:

**Propuesta de procedimiento para el
Mantenimiento de Software**

Autor: Ing. Dania Souchay Fábrega

Tutor:

Msc. Michael González Jorin

Ciudad de La Habana, 2009

Dedicatoria

A todos los que me han acompañado en el largo camino de mi aprendizaje.

Agradecimientos

Al profesor Francisco Ruiz quien me enseñó la diferencia de tocar en una puerta pequeña y una grande sobre todo cuando esta última se abre.

A mi tutor y amigo Michael por su genialidad incomparable!!!

A los profesores del CREA especialmente a Cañas por ser nuevamente mis compañeros de trabajo.

A quienes constituyeron pilares fundamentales en mi día a día en esta etapa de mi vida Yanier, Yoisy, Lily, Maypher.

"A" por mirar con ojo crítico el desarrollo de este trabajo porque me dio aliento para andar por el camino de la profesionalidad de manera rigurosa.

A mi familia y amigos de siempre, por constituir mi aliento constante, como en mi primera gran investigación, esta segunda también va para todos ustedes!!

Declaración de Autoría

Declaro que soy la única autora del presente trabajo. Autorizo a la Universidad de las Ciencias Informáticas (UCI) a darle al mismo el uso que estime pertinente. Para que así conste se firma el presente documento a los ____ días del mes de _____ del _____.

Ing. Dania Souchay Fábrega
Autora

Msc. Michael González Jorin
Tutor

Resumen

La actividad de Mantenimiento de Software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales de las mismas (y su supervivencia) dependen de los sistemas informáticos para su buen funcionamiento.

A pesar de la importancia que se le reconoce a esta etapa de evolución de las aplicaciones, por lo general es llevada a cabo sin orden formal, más bien al estilo libre del mantenedor. Muchos han sido los esfuerzos en estandarizar y definir procedimientos para normar esta actividad, pero por lo general se enfocan en la tercerización de la misma, como servicio a prestar a entidades que lo requieren y dirigido a una industria de software organizada, consolidada y multidisciplinaria, dejando al margen las pequeñas organizaciones que invierten esfuerzo propio en desarrollar sus sistemas y además mantenerlos.

Esta investigación se centra en la elaboración de una propuesta de procedimiento para la realización de las actividades de mantenimiento de software, dirigido pequeñas organizaciones y empresas, a fin de proporcionarle una guía de realización que permita lograr mejoras en operación y calidad de los sistemas que se encuentran en uso actualmente.

En el área de superación pedagógica y eventos del Centro de Referencia para la educación Avanzada (CREA), citado en la Ciudad Universitaria José Antonio Echeverría (CUJAE) Ciudad de la Habana, tiene lugar la gestión de la información de las actividades de postgrado que en el mismo se realiza, utilizando para ello el sistema GSP. El uso y puesta en práctica de la propuesta sobre esta aplicación de gestión permitirá extender su vida útil, conllevando al mejor aprovechamiento de las herramientas informáticas con que ya cuenta el centro, sin tener que desechar soluciones implementadas, sino reutilizarlas en función de las necesidades actualmente imperantes en la organización.

Palabras claves: Mantenimiento de software, Procedimiento, Mantenibilidad, Sistemas de gestión.

Índice

Introducción	1
Capítulo 1 Fundamentos Teóricos del Mantenimiento del Software	10
Introducción	10
1.1 Evolución del software	10
1.2 Gestionar la Evolución	12
1.2.1 Mantenimiento	13
1.2.2 Reingeniería.....	14
1.2.3 Abandono	16
1.3 Procesos del ciclo de vida del Software	16
1.4 Proceso de Mantenimiento.....	18
1.5 Tipos de mantenimiento.....	20
Conclusiones parciales	24
Capítulo 2 Mantenimiento de Software en la Industria	25
Introducción	25
2.1 Análisis de estándares y procedimientos para el mantenimiento de software	25
2.1.1 Estándar ISO 14764.....	25
2.1.2 Estándar IEEE 1219.....	28
2.1.3 Modelo de Madurez de Mantenimiento de Software (S3M) ...	Error! Marcador no definido.
2.1.4 Metodología MANTEMA.....	33
2.1.5 Ágil_MANTEMA	35
2.1.6 Otras metodologías y procedimientos	35
2.2 Características de las pequeñas organizaciones	36
2.3 Características generales de los Software de Gestión	37
2.4 Herramienta de Soporte.....	38
2.5 Calidad en el Mantenimiento de Software	39
Conclusiones Parciales	42
Capítulo 3 Propuesta de procedimiento para el Mantenimiento de Software.....	43
Introducción	43
3.1 Consideraciones generales.....	43
3.2 Alcance	43
3.3 Roles.....	44
3.4 Tipos de mantenimiento.....	45
3.5 Estructura general del procedimiento para el Mantenimiento de Software	45
3.6 Estructura detallada del Procedimiento de Mantenimiento	47
Conclusiones Parciales	57
Capítulo 4 Aplicación del Procedimiento Propuesto	58
Introducción	58

4.1	Descripción del Sistema GSP: Gestión de la Superación Pedagógica	58
4.2	Evaluación de la calidad de software antes de la ejecución de la propuesta	60
4.3	Ejecución de la propuesta de mantenimiento	63
4.4	Evaluación de la calidad de software después de la ejecución de la propuesta	73
4.5	Resultados Obtenidos	75
	Conclusiones Parciales	81
	Conclusiones	82
	Recomendaciones	83
	Referencias Bibliográficas	84
	Anexos	87

Introducción

En el mundo empresarial, marcado por el acelerado ritmo del desarrollo tecnológico y el constantemente cambio de sus procesos de negocios para sobrevivir en un mercado competitivo, las organizaciones son cada vez más conscientes de la importancia de los sistemas de gestión en la consecución de sus metas. Partiendo del hecho de que un sistema de gestión es una estructura probada para la propia gestión y mejora continua de las políticas, los procedimientos y procesos de la organización, éste ayuda a lograr los objetivos de la misma mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado como se plantea en (1). Debido a la necesidad de permitir aprovechar y desarrollar el potencial existente con que cuenta la organización, se ha hecho necesario, tanto en los países desarrollados como en los subdesarrollados, el auge de la utilización y aprovechamiento de los sistemas de gestión considerados como un activo corporativo que puede suponer una ventaja competitiva.

A menudo es necesario realizar mejoras y modificaciones a estos sistemas existentes y establecidos en la organización, debido a que los requisitos y las demandas de las necesidades propias de la organización y del usuario que interactúa con la aplicación cambian. Eso se traduce a la necesidad de la actividad de mantenimiento aplicado al software existente que puede llegar a ser una parte muy importante de los costes del ciclo de vida de su desarrollo.

Se entiende por “Mantenimiento del Software” según la definición de ISO/IEC: como “ *Un producto software soporta una modificación en el código y su documentación asociada para la solución de un problema o por la necesidad de una mejora. Su objetivo es mejorar el software existente manteniendo su integridad*”. (2)

El mantenimiento supone alrededor del 80% de los costes en el ciclo de vida de un producto software ⁱ y a pesar de que la fiabilidad del software depende en gran medida del rigor y la calidad con la que se efectúe su mantenimiento, la mayor parte de las organizaciones no prestan la suficiente atención al estudio de un proceso que a menudo reviste una especial criticidad para las mismas.

Múltiples estudios señalan que estadísticamente está comprobado que el coste de mantenimiento de un producto software a lo largo de toda su vida útil supone más del doble que los costes de su desarrollo. (3) La tendencia es creciente con el paso del tiempo.

ⁱ Planteado en las referencias (9), (3), (27)

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 (prev.)	90%

Fig. 1 Tendencia de la elevación del costo del Mantenimiento del Software

El esfuerzo de Mantenimiento de Software se podría reducir si se produce un software nuevo de mejor calidad. Igualmente, también se podrán reducir los costes futuros, si el mantenimiento se realiza utilizando técnicas que mejoren alguna de sus características y contribuyan a elevar la calidad del mismo. (3)

Se entiende en la investigación por Calidad de Software “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente” (4). Dentro de la Calidad de Software como característica fundamental a tener en cuenta en esta investigación se encuentra la mantenibilidad la cual se define como la capacidad de un producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requerimientos o en las especificaciones funcionales. Es el atributo de calidad del software que más directamente influye en los costes y necesidades del mantenimiento, debido a que a mayor mantenibilidad (facilidad de mantenimiento), menores costes de mantenimiento, y viceversa.

En Cuba con el esfuerzo que se realiza para llevar a cabo la informatización de la sociedad, comenzando fundamentalmente por las empresas y organizaciones, este aspecto se ha convertido en elemento primordial a tener en cuenta. En los centros de desarrollo de software cada vez más se aboga por establecer procedimientos pensados en la fase futura de mantenimiento del software. Adecuada documentación, estilos y comentarios de códigos, legibilidad en el diseño, gestión de cambio y configuración han convertido en prácticas habituales de los equipos de desarrollo de software.

Comúnmente, las actividades de mantenimiento a sistemas existentes se han realizado en las organizaciones de una manera ad hoc en un estilo libre establecido por el propio programador. No en todas las ocasiones esta situación es debido a la falta de tiempo para producir una modificación diseñada cuidadosamente. Prácticamente todas las metodologías se han centrado en nuevos desarrollos y no han tenido en cuenta la

importancia del mantenimiento. Por esta razón, son poco conocidos los métodos, técnicas y herramientas que proporcionan una solución global al problema del mantenimiento. (5)

Existen diversas definiciones y normas de organismos internacionales de estandarización que tienen relación directa o indirecta con el Mantenimiento del Software. Dentro de las mismas se encuentran para los procesos del ciclo de vida del software: IEEE 1074 e ISO 12207 y enmarcados en la fase de mantenimiento en el ciclo de desarrollo de software, la norma ISO/IEC 14764 y el estándar de la IEEE 1219. Generalmente dentro de las especificaciones estos se enuncian y se refieren a equipos de desarrollo compuesto por varios miembros donde cada uno tiene muy bien identificado el rol que juega dentro del mismo, avalado y respaldado por una documentación que recoge desde el cuestionario inicial, la definición del contrato de la fase de mantenimiento entre otros artefactos intermedios hasta el plan de migración y retiro de la versión anterior del sistema. Utilizando la guía que establecen dichas normas se tiene un fundamento que sirve de base para realizar las distintas actividades asociadas al Mantenimiento de Software. Estas regulaciones de mantenimiento se enfocan de manera general dirigidas a la industria de software y empresas en las que su oportunidad de mercado consiste en prestar servicio de contratación de mantenimiento refiriéndose en mayor medida a la externalización de la actividad de Mantenimiento de Software.

En un sondeo realizado a profesionales de la informática en Cuba pertenecientes a organizaciones y empresas con una muestra de 12 entidades, entre ellas Terminal de Contenedores de la Habana (TCH), Polos Científicos (Inmuno Ensayos, ICID), Agencia de Comercio Electrónico Soy Cubano, empresas pertenecientes al ministerio de Informática y las Comunicaciones como Softel, Correos de Cuba, diferentes áreas la Universidad de las Ciencias Informáticas (UCI) entre otros, se identificaron elementos importantes relacionados con la actividad de mantenimiento, actividad a la cual todos estaban vinculados directamente o indirectamente en sus organizaciones. Como resultado de una encuesta (Anexo 1) que les fue aplicada, se pudo comprobar que todos los encuestados asignan un alto nivel de prioridad a la realización del Mantenimiento de Software dentro de sus organizaciones, y sin embargo sólo en el 5 de las organizaciones consideran que se obtienen buenos resultados relacionados con la calidad en la mejora la mantenibilidad del sistema modificado.

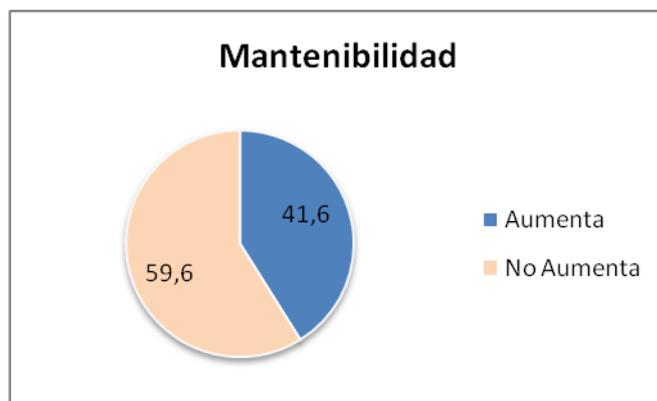


Fig. 2 Variación en la Mantenibilidad en el Mantenimiento del Software

Es de destacar que en estas instituciones la actividad de mantenimiento generalmente es realizada de manera individual aunque en algunas ocasiones puede llegar a incluir hasta 3 especialistas.

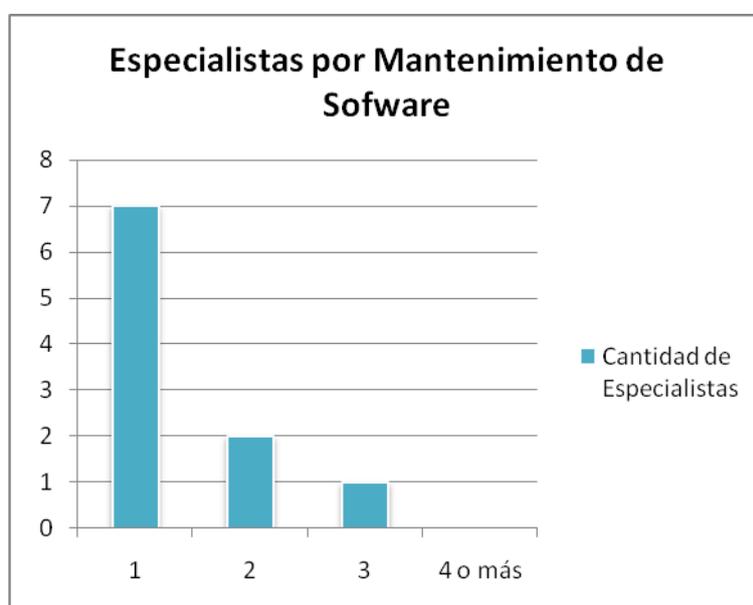


Fig. 3 Cantidad de especialistas vinculados al mantenimiento del Software

Esto último se relaciona con que no siempre el desarrollo de software está ligado a equipos multidisciplinarios donde cada miembro juega un rol bien identificado.

En la industria cubana del software ha habido muchos desarrollos de sistemas realizados a la medida de los procesos internos de varias organizaciones donde los desarrolladores de dichas aplicaciones integran generalmente equipos pequeños oscilando de uno a tres informáticos en plantilla en la organización. Según estudios realizados por el “Centro de Referencia de Ingeniería de Software” (CRIS) de la CUJAE descrito en (6), es muy común la constante actividad de mantenimiento a realizar sobre dichos software en explotación. A la hora de enfrentarse a la tarea de mantenimiento, y como fue corroborado con la

encuesta aplicada, generalmente designan a una persona para atender un determinado sistema. En la ejecución de esta actividad no existe una adecuada planificación del trabajo y además no se registra información de los cambios realizados en el sistema o cuando se hace no es suficiente para poder llevar una trazabilidad de la evolución del mismo. La planificación y ejecución de esta actividad se basan en la experiencia y el buen juicio de una persona que esté a cargo de ello y en su generalidad se adolece del conocimiento, de procedimientos o modelos que indiquen o guíen al mantenedor en cómo realizar el mantenimiento a estos sistemas desarrollados a la medida para un proceso particular vigente en la organización. (6)

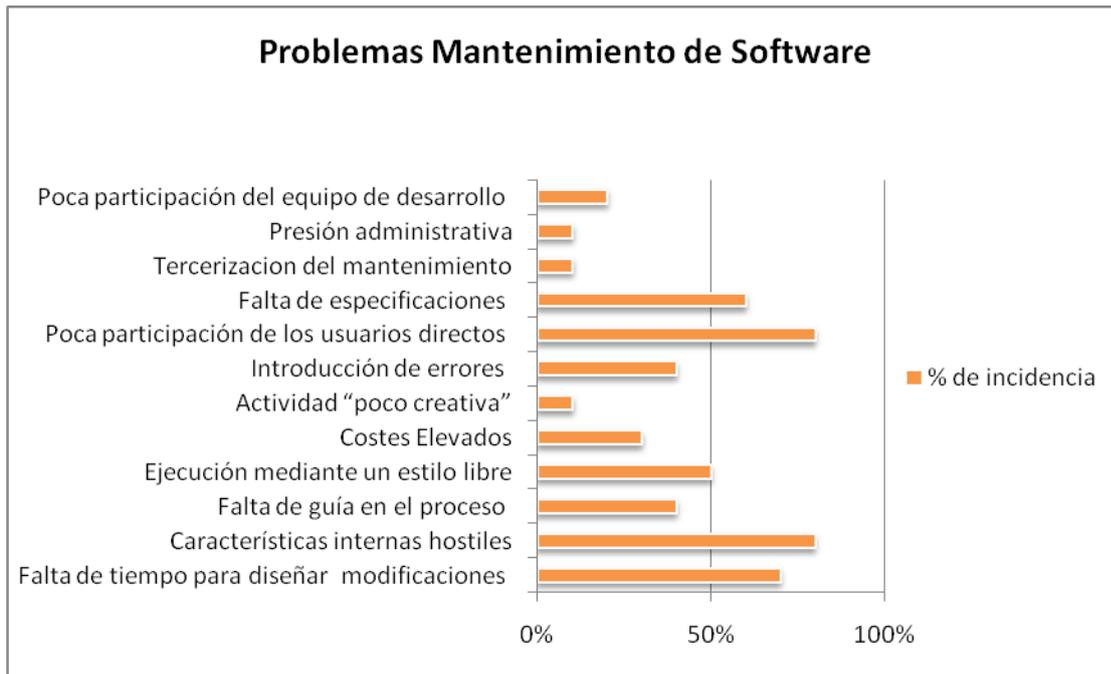


Fig. 4 Problemas relacionados con la actividad del Mantenimiento del Software

Las causas detectadas en el análisis de la encuesta, asociadas al bajo indicador de calidad en el mantenimiento con más de un 50% de reincidencia en selección, están asociadas fundamentalmente a:

- La falta de tiempo para diseñar las modificaciones cuidadosamente.
- Falta de disponibilidad de documentación y especificaciones de diseño o la existente es incomprensible, incorrecta o insuficiente.
- Poca participación de los usuarios directos del sistema en el proceso de mantenimiento.
- La ejecución del mismo mediante un estilo libre establecido por el propio programador.
- Características internas del sistema considerablemente hostiles como diseño pobre de las estructuras de datos, mala codificación, lógica defectuosa.

Aunque directamente la falta de guía en la realización del proceso en la encuesta no tuvo un índice de selección superior al 50%, los factores que sí se seleccionaron inciden en este elemento directa o indirectamente.

Solo 4 de los encuestados notifica utilizar procedimiento para la realización del mantenimiento enmarcado no propiamente en el Mantenimiento de Software sino como políticas de sus instituciones para cumplir con la norma de calidad ISO 9000:1.

Como se puede evidenciar a pesar de que la actividad de Mantenimiento de Software en una entidad resulta necesaria para la supervivencia de la misma, los resultados de su ejecución son relativamente bajos dados fundamentalmente por la falta de un procedimiento o guía definida, que permita realizar y documentar el desarrollo de dicha actividad.

Estos elementos identificados anteriormente también se encuentran reflejados en el ya mencionado Centro de Referencia para la educación avanzada (CREA). Este centro forma parte de las organizaciones que también se preocupa por el adecuado funcionamiento y la actualización constante de los soportes de tecnologías de información que automatizan sus procesos internos guiados por la fuerte motivación de lograr la excelencia y declararse apto para cumplir exitosamente el proceso de acreditación que se desarrolla hoy en día en el mismo.

En el área de superación pedagógica de dicho centro se desarrolla el proceso de control y gestión de las actividades de superación posgraduadas que imparten los profesores del centro. Esta actividad esta soportada por un sistema de gestión sobre plataforma Web en explotación, que automatiza lo referente a las necesidades de información de superación pedagógica que tiene lugar en la organización. Según entrevista realizada al usuario directo de la aplicación y encargado de la actividad de superación en el centro, actualmente el sistema presenta algunas deficiencias de funcionamiento en determinadas funcionalidades y no cubre algunos elementos nuevos que han surgido dentro del propio proceso de control en el área. Dentro de dichos elementos están la creación de proyectos productivos, la vinculación de los temas de investigación de los maestrantes a los mismos, la necesidad de búsqueda de registros debido al gran cúmulo de información con que ya cuenta el sistema, la necesidad de nuevos requerimientos de información por cada una de las actividades de superación entre otros elementos son los que expresan la necesidad de reajuste del sistema actual a estos nuevos requerimientos. La dirección del CREA, y el usuario directo de la aplicación están muy interesados en la realización del mantenimiento que actualmente necesita el sistema de gestión de la superación pedagógica (GSP) por ser una de las aplicaciones fundamentales que soporta procesos internos de dicha organización.

Del análisis de esta situación se desprende como **problema de investigación** *la insuficiencia en la adaptabilidad del proceso de mantenimiento dentro del ciclo de vida del software de gestión en pequeñas organizaciones influye negativamente en la mantenibilidad de los mismos.*

Para dar respuesta a este problema se plantea la siguiente **hipótesis** *la adaptación del proceso de mantenimiento dentro del ciclo de vida del software de gestión para pequeñas organizaciones mejorará la mantenibilidad en los mismos.*

Se define así el **objeto de estudio** de la investigación como: *proceso de mantenimiento dentro del ciclo de vida del software*, y como **campo de acción**: *Actividades enmarcadas en el proceso de mantenimiento de los sistemas de gestión.*

Para dar respuesta al problema de la investigación, se formuló el **siguiente objetivo general**:

Desarrollar un procedimiento de Mantenimiento de Software para sistemas de gestión ajustado a las necesidades y características de pequeñas organizaciones.

Como resultado del análisis del problema de investigación y para dar cumplimiento al objetivo general enunciado, se plantearon los siguientes **objetivos específicos**:

- Evaluar el estado del arte de elementos teóricos asociados a la fase de Mantenimiento de Software, sus fronteras dentro de la evolución del software, tipos de mantenimiento, características de las pequeñas organizaciones y los sistemas de gestión, así como la mantenibilidad como atributo de calidad.
- Elaborar la propuesta de procedimiento para realizar el Mantenimiento de Software en sistemas de gestión en pequeñas organizaciones.
- Aplicar el procedimiento obtenido para la formalización del proceso de mantenimiento en el sistema de gestión GSP del CREA.
- Evaluar los resultados durante la aplicación del procedimiento propuesto en el proceso de mantenimiento del sistema de gestión GSP del CREA.

Para cumplir estos objetivos se realizan las siguientes **tareas**:

- Estudio sobre el estado del arte de la fase mantenimiento, conceptos, características y principales clasificaciones, delimitación entre otros procesos del ciclo de vida y evolución del software para definir y enmarcar el objeto de estudio de la investigación.
- Revisión de estándares, metodologías, modelos comprendidos dentro del proceso de Mantenimiento de Software para identificar las actividades y tareas a realizar en esta actividad.

- Análisis y evaluación de las mejores prácticas que proponen metodologías y estándares que abordan el tema del Mantenimiento de Software que puedan ser tomadas en cuenta para realizar la propuesta del procedimiento.
- Identificación y descripción de los flujos de actividades a realizar, los roles involucrados y propuesta de herramientas que soporten el procedimiento basándose en las mejores prácticas utilizadas internacionalmente.
- Aplicación del procedimiento propuesto en el sistema de gestión GSP del CREA, para la realización del mantenimiento haciendo uso de las herramientas propuestas.
- Evaluación de los resultados obtenidos después de realizado el mantenimiento al sistema GSP.

Entre los métodos de trabajo científico utilizados en esta investigación se destacan los siguientes:

- El uso de métodos estadísticos con empleo de técnicas de muestreo intencional para la selección de organizaciones con procesos principales automatizados, que sería la población definida para la investigación donde el proceso de mantenimiento de los sistemas de gestión serían la unidad de estudio.
- Métodos lógicos: El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos de forma independiente, para luego sintetizarlos en la solución de la propuesta.
- Métodos empíricos: El método de la entrevista para obtener las principales dificultades existentes hoy en el sistema GSP y el método encuesta para obtener información sobre el estado de la actividad de Mantenimiento de Software de gestión en organizaciones que tienen procesos soportados por sistemas que necesitan ser modificados.

La **Novedad Científica** de la investigación está expresada en que hasta el momento no se conoce de un procedimiento ajustado a las necesidades y requerimientos de las pequeñas organizaciones y empresas nacionales, basado en los estándares internacionales enmarcados en el área, que permita guiar el proceso de cómo ejecutar Mantenimiento de Software en sistemas de gestión.

El **Valor Práctico** del trabajo está dado en que se ofrece un procedimiento que indica cómo llevar a cabo el proceso de Mantenimiento de Software en la industria nacional, considerándose su aplicación en pequeñas organizaciones que posean sistemas de gestión con necesidad de mantenimiento. Se ofrece además, planillas como herramientas que facilitan el registro de la actividad de mantenimiento para cumplimentar la documentación existente de la evolución del sistema al que se le aplica mantenimiento.

El documento de tesis está compuesto en su estructura por cuatro capítulos. El Primer capítulo de *“Fundamentos Teóricos del mantenimiento del software”*, se exponen definiciones, clasificaciones y características de esta actividad. El segundo capítulo sobre: *“Mantenimiento de Software en la Industria”* se realiza una evaluación del estado del arte del mantenimiento de software. Además se describen y analizan algunos estándares y metodologías existentes para llevar a cabo la fase de mantenimiento en el ciclo de vida de software. El tercer capítulo sobre: *“Propuesta de procedimiento para el mantenimiento de Software”*, aborda la descripción de la propuesta a realizar del que objeto este capítulo, con una descripción de cada una de las actividades a realizarse. También se hace un estudio sobre la calidad de software fundamentalmente la característica de mantenibilidad y como evaluarla. El capítulo cuatro *“Aplicación del Procedimiento Propuesto”* se lleva a la práctica la aplicación del procedimiento de Mantenimiento de Software propuesto sobre la aplicación GSP del CREA y se realiza un análisis de los resultados una vez culminada la fase de mantenimiento en la nueva versión obtenida del sistema.

Capítulo 1 Fundamentos Teóricos del Mantenimiento del Software

Introducción

Para desarrollar una investigación científica debe tenerse en cuenta una base teórica que sustente y respalde el trabajo a desarrollar en el curso investigativo. En este capítulo se exponen elementos relacionados con el mantenimiento de software, se precisan términos importantes para la comprensión del objeto de estudio y campo de acción donde se enmarca la investigación asociados proceso de mantenimiento entre otras aclaraciones necesarias para llevar a cabo esta actividad.

1.1 Evolución del software

Los sistemas que fueron desarrollados y puestos en práctica para su uso en estos años ,se fueron convirtiendo en parte vital de muchas organizaciones, por lo que se ve la constante necesidad de corregirlos, para así poder alargar su vida útil y garantizar su continuidad.

Las necesidades cambiantes del negocio donde se usan y brindan su utilidad los sistemas de gestión demandan que estos vayan a la par de dichas transformaciones acompañándolos en el avance y desarrollo propio de cada organización. La realidad ha demostrado que existe una gran dificultad a la hora de lograr este propósito y particularmente a la hora de mantener los programas. (7) Dentro de las principales dificultades en ello se encuentran:

- Dificultad para que el software evolucione
- Cambios frecuentes en los requisitos
- Diseño pobre / rígido
- Problemas de obsolescencia
- Cambios tecnológicos (8)

Objetivos del Mantenimiento y la Evolución

La evolución de software consiste en un conjunto de actividades necesarias para mantener operativo un software. (8) Estas actividades incluyen:

- Corregir los defectos (mantenimiento)
- Mejorar la funcionalidad del software (evolución)
- Mejorar la calidad del software existente (mantenimiento) (8)

Estas actividades permiten que el sistema que evoluciona continúe su vida útil. Al realizar cambios, ampliar y mejorarse las prestaciones y funcionalidades que brinda el software permite mantener contentos a los usuarios. Mejorar el sistema con que se cuenta aprovechando la experiencia de su uso, permite maximizar la inversión y reducir costes y

en gran medida permite alargar la vida del software posibilitando su adaptación a los nuevos cambios o requisitos.

Leyes del mantenimiento del Software

Lehman (1974) formuló las primeras “leyes de la mantenimiento del software” por primera vez a partir de un estudio del proceso de programación. En el ámbito de ciencias de la ingeniería, una ley debe entenderse como una característica común a muchos fenómenos o que se presenta con regularidad. (3)

A continuación se hace referencia a las Leyes de Lehman (9) para el mantenimiento del software:

- *Continuidad del Cambio:” Un programa utilizado en un entorno del mundo real está destinado a cambiar, ya que, en caso contrario, será utilizado cada vez menos en dicho entorno (tan pronto como un programa ha sido escrito, está ya desfasado).”*

Las razones que conducen a esta afirmación son varias:

- A los usuarios se les ocurren nuevas funcionalidades cuando comienzan a utilizar el software;
 - Nuevas características en el hardware pueden permitir mejoras en el software;
 - Se encuentran defectos en el software que deben ser corregidos;
 - El software debe instalarse en otro sistema operativo o máquina
 - El software necesita ser más eficiente.
- *Incremento de la Complejidad:” A la par que los cambios transforman los programas, su estructura se hará progresivamente más compleja salvo que se haga un esfuerzo activo para evitar este fenómeno.”*
 - *Evolución del Programa:” La evolución de un programa es un proceso autorregulado. Las medidas de determinadas propiedades (tamaño, tiempo entre versiones y número de errores) revelan estadísticamente determinadas tendencias e invariantes.”*
 - *Conservación de la Estabilidad Organizacional:” A lo largo del tiempo de vida de un programa, la carga que supone el desarrollo de dicho programa es aproximadamente constante e independiente de los recursos dedicados.”*
 - *Conservación de la Familiaridad:”Durante todo el tiempo de vida de un producto de software, el incremento en el número de cambios incluidos con cada versión (release) es aproximadamente constante”*

Estas leyes referenciadas de (9) tributan a problemas y dificultades asociadas a la actividad de mantenimiento de software dentro de las cuales pueden sumarse otros

problemas de gestión y técnicos descritos en la bibliografíaⁱⁱ dentro de los que se encuentran:

1. Ausencia metodológica.
2. Tendencia a la desestructuración.
3. Disminución de la comprensibilidad.
4. Poca participación de los usuarios.
5. Muchos programadores consideran el trabajo de mantenimiento como una actividad con bajo componente creativo.
6. Realizar mantenimiento, en vez de emplear una estrategia sistemática, las correcciones tienden a ser realizadas con precipitación, sin pensarse de forma suficiente, no documentadas adecuadamente y pobremente integradas con el código existente.
7. El propio mantenimiento conduzca a la introducción de nuevos errores e ineficiencias que conducen a nuevos esfuerzos de mantenimiento con posterioridad.

Como propuesta para resolver estos problemas según (3) , desde el punto de vista de las soluciones de gestión (organizativas) se recomienda un buen respaldo por parte de la organización en la mejora de los recursos dedicados al mantenimiento, la gestión estructurada del proceso, documentación de los cambios y desde la óptica de las soluciones técnicas el uso de herramientas y metodologías. Es por ello que la propuesta a realizar se enmarcará dentro de este ámbito de solución.

1.2 Gestionar la Evolución

La evolución de un sistema es un concepto amplio; abarca desde una simple modificación para corregir un “bug” de un programa hasta una reimplantación completa del sistema. Las actividades de evolución que se pueden realizar sobre un sistema son de tres tipos: mantenimiento, reingeniería y abandono (7). La figura 5(9) ilustra cómo son aplicadas cada una de las actividades de evolución durante el ciclo de vida de un sistema.

ⁱⁱ Descritos en la referencias (9), (3), (5) , (8)

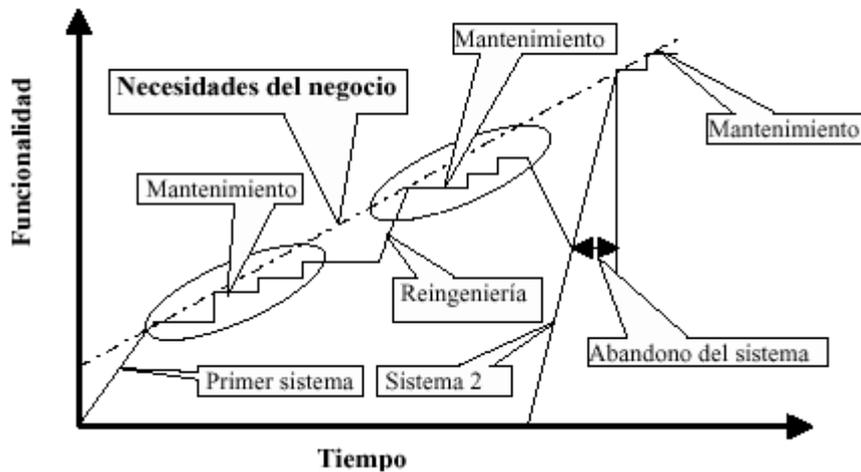


Fig. 5 Actividades de evolución en el ciclo de vida de un sistema.

La figura ilustra las 3 actividades que pueden estar presente durante la evolución del sistema, donde el mantenimiento es la que más presencia tiene de manera iterativa durante el período evolutivo.

1.2.1 Mantenimiento

El mantenimiento históricamente se puede considerar como la primera forma de evolución de los sistemas. La mayoría de las empresas que realizaron sus grandes desarrollos tiempos atrás, necesitan, sobre todo si la aplicación sigue resultando efectiva, la realización de estas actividades.

El ANSI/IEEE da la siguiente definición de mantenimiento:

Las modificaciones de los productos software después de su entrega para corregir fallos, mejorar rendimiento u otros atributos o adaptar el producto a un cambio de entorno.

Una definición similar es dada por ISO/IEC:

Un producto software soporta una modificación en el código y su documentación asociada para la solución de un problema o por la necesidad de una mejora. Su objetivo es mejorar el software existente manteniendo su integridad.

Otra definición de Pressman [1998]:

La fase de mantenimiento se centra en el cambio que va a asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente.

Todas estas definiciones de una forma u otra coinciden en que la esencia del Mantenimiento de Software se refiere al cambio o modificación a realizar sobre un producto de software, con diferentes fines ya sea para corregir una falla, mejorar algún elemento o

realizar una adaptación al entorno cambiante, elementos que serán tomados de base en la investigación para enmarcar la propuesta en el marco del mantenimiento de software.

El mantenimiento representa el porcentaje más alto del coste de todo el ciclo de vida de un sistema (8). El problema fundamental es que, el elevado número de cambios que el software soporta a lo largo de su ciclo de vida, hace imposible una previsión desde el diseño del sistema. Por ello, es previsible que la investigación en el campo del Mantenimiento del Software vaya en aumento progresivamente como muestra la necesidad de esta tesis.

Los principales métodos empleados en el Mantenimiento del Software como se describen en (3) son la reingeniería, la ingeniería inversa y la reestructuración:

- Reingeniería consiste en el examen y modificación de un sistema para mejorar la comprensión del software, preparar o mejorar el propio software, normalmente para incrementar su facilidad de mantenimiento, reutilización o evolución.
- Ingeniería Inversa es el proceso de analizar un sistema para identificar sus componentes y las interrelaciones que existen entre ellos, así como para crear representaciones del sistema en otra forma o en un nivel de abstracción más elevado.
- Reestructuración del software consiste en la transformación de un producto software a otra forma de representación, pero sin cambiar de nivel de abstracción.

Estas tres técnicas tienen relación entre ellas y a por lo general se utilizan conjuntamente.

1.2.2 Reingeniería

Posterior a las técnicas de microevolución del mantenimiento, que tienden a moverse a los niveles de abstracción más bajos del sistema, surge la reingeniería para cubrir la necesidad de una evolución más amplia del sistema. La reingeniería trabaja todos los niveles de abstracción (desde la implementación hasta el diseño) para conseguir realizar cambios más drásticos preservando los valores de negocio del sistema. (8)

Para entender en qué consiste la reingeniería se revisará la definición dada por el Reengineering Center del Software Engineering Institute de la Universidad Carnegie Mellon (1) que plantea:

Reingeniería es la transformación sistemática de un sistema existente a una nueva forma para realizar mejoras de la calidad en operación, capacidad del sistema, funcionalidad, rendimiento o capacidad de evolución a bajo coste, con un plan de desarrollo corto y con bajo riesgo para el cliente.

La actividad de reingeniería es una de las alternativas para la actualización de los sistemas legados.

Las concepciones alrededor de la reingeniería de software son muy variadas, van desde suponer mejoras en la comprensibilidad y Mantenimiento del Software, diagnosticar y reconstruir el software, aplicación de ingeniería inversa e ingeniería directa al código existente, pero quizás la más amplia y aceptada es considerarla una forma de reutilización, que mejora la calidad y reduce el esfuerzo de llevar un sistema existente a un nuevo contexto.

A continuación se hará referencia breve a cada una de estas perspectivas que se describen más en detalle en (10):

- 1) *Solución de un problema delimitado*: Asume como un problema de ingeniería que considera un sistema actual y busca mediante un proceso de comprensión de la aplicación y un plan a desarrollar, evolucionar hasta conseguir el sistema deseado.
- 2) *Perspectiva de sistema*: Toma como referencia un modelo que guía el proceso de reingeniería. El modelo propone actividades para la planeación, identificación descripción, comprensión, fortalecimiento y evaluación de factores técnicos, difundiendo las lecciones aprendidas al respecto.
- 3) *Perspectiva administrativa*: Se establece un plan con tres campos de acción: definición de objetivos, que marcan el rumbo del proyecto, la priorización de estos y definición de estrategias para alcanzarlos. Luego se determinan y recuperan los activos del sistema legado que con el tiempo se han deteriorado y se establece un plan para alcanzarlos.
- 4) *Perspectiva evolutiva*: Basado en una continua evolución que supone los siguientes principios: el producto de software nunca está terminado. El diseñador del software toma como referencia una serie de activos existentes. Supone que el usuario final está directamente ligado al proceso de desarrollo de software.
- 5) *Reingeniería desde una perspectiva del software*: Propone reescribir el software legado para hacerlo mantenible y reutilizable. Define tareas como: organizar, reestructurar y remodularizar el software, hacer revisiones, mediciones e ingeniería inversa sobre el código, traducir a un lenguaje de programación moderno y flexible, revisar el formato y el contenido de los datos que se manejan y reestructurar la documentación existente o generar una nueva. Puede hacerse utilizando varias alternativas:
 - Modernización de caja blanca
 - Modernización de caja negra o Wrapping

El propósito de la reingeniería es la reconstrucción de sistemas heredados para incrementar su calidad y mantenibilidad. La utilidad de la misma va desde suponer mejoras en la comprensibilidad y Mantenimiento del Software, diagnosticar y reconstruir el software, aplicación de ingeniería inversa e ingeniería directa al código existente, pero quizás la más

amplia y aceptada es considerarla una forma de reutilización, que mejora la calidad y reduce el esfuerzo de llevar un sistema existente a un nuevo contexto.

1.2.3 Abandono

Un sistema se abandona cuando no es capaz de seguir el ritmo de las necesidades del negocio y su reingeniería no es posible o no se puede hacer con un coste efectivo. Normalmente, estos sistemas adolecen de documentación, de actualización y de capacidad de expansión.

El abandono de un sistema comporta riesgos:

- Es una actividad que consume muchos recursos y que implica grandes esfuerzos a los mantenedores actuales del sistema. Estos recursos estarán ocupados en el mantenimiento del sistema y, seguramente, no conocerán las nuevas tecnologías que van a ser aplicadas en el desarrollo del nuevo sistema.
- Abandonar el sistema no significa empezar un desarrollo desde cero. No se puede pensar que los datos que contiene el legacy van a desecharse y que los valores que conservan los expertos del sistema van a perderse. El abandono del sistema debe venir precedido por un análisis exhaustivo que permitan decidir el método que se va a utilizar para la migración de la información del sistema actual al nuevo, incluyendo datos, funciones de negocio y arquitectura del sistema. (7)

1.3 Procesos del ciclo de vida del Software

La complejidad del proceso de producción de software se intenta abordar mediante la descomposición en diversas etapas. Esta descomposición ha recibido el nombre de Ciclo de Vida del Software.

En el contexto general de los procesos de software es necesario desarrollar una arquitectura donde se definan dichos procesos. La norma ISO/IEC 12207 *International Standard for Information Technology – Software Life Cycle Processes* describe la arquitectura de los procesos del ciclo de vida del software. Este estándar establece un marco de referencia común para los procesos del ciclo de vida software, con una terminología bien definida, que puede ser referenciada por la industria software. Dicha norma no especifica los detalles de cómo implementar o llevar a cabo las actividades y tareas incluidas en los procesos.

A continuación se presenta de forma resumida la norma ISO, en la que se basará para exponer las actividades y tareas del proceso de mantenimiento a proponer en la investigación. Esta norma agrupa, como puede verse en la Figura 6, las actividades que pueden ser realizadas durante el ciclo de vida del software en cinco procesos principales, ocho procesos de soporte, y cuatro procesos organizativos, así como el proceso de adaptación.



Fig. 6 Procesos del ciclo de vida software según la norma ISO 12207.

El “Proceso de Mantenimiento”, contiene las actividades propias de este proceso y del mantenedor en sí. La norma plantea que este proceso se activa cuando el producto de 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. (11) Entre las actividades de mantenimiento asociadas al proceso que propone la norma están:

- Implementación del proceso.
- Análisis de problemas y modificaciones.
- Implementación de las modificaciones.
- Revisión/aceptación del mantenimiento.
- Migración.
- Retirada del software.

Es necesario destacar que también existe el estándar de la IEEE 1074 -1995, *Developing Software Life Cycle Processes*, detalla el conjunto de actividades que aparecen obligatoriamente en el desarrollo y Mantenimiento del Software. La clasificación que propone se realiza dependiendo de que los procesos sean de gestión de proyectos, antes del desarrollo, durante el desarrollo, después del desarrollo o durante todo el ciclo de desarrollo de un producto software por lo que se consideró que la ISO/IEC 12207 tenía un enfoque más general y abarcador y es por ello que fue objeto de análisis.

Destacar que según se plantea en la referencia 41, las actividades vinculadas con el soporte de software se incluye dentro de sus procesos principales, como servicios vinculados a la operación y mantenimiento de productos de software. Plantea que en estos

se describe la manera de realizar las actividades relacionadas con la atención a las organizaciones en la búsqueda de solución a los problemas que ocurran durante la operación de los productos de software. Estas actividades están abordadas desde los puntos de vistas del operador o usuario del software y los desarrolladores y mantenedores.

Se plantea en esta tesis que el proceso de operación del software tiene como propósito la operación y prestación de servicios de soporte a los clientes del producto de software y de esta manera incluye en el conjunto de tareas para la actividad de Soporte al Usuario.

1.4 Proceso de Mantenimiento

Como se representa en la figura 6, la norma de ISO considera el mantenimiento como uno de los procesos principales del ciclo de vida del software, ya que “define las actividades de la organización (mantenedor) que proporciona el servicio de mantener el producto software; es decir, gestionar las modificaciones al producto software con el fin de mantenerlo actualizado y adecuado a su uso” (12).

La figura 7 precisa la estructura del proceso de mantenimiento según ISO/IEC:

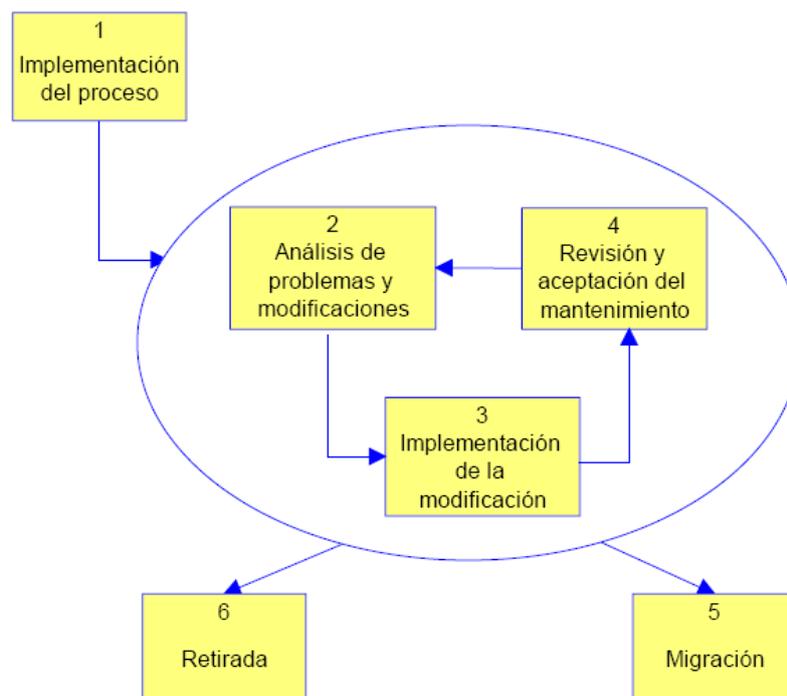


Figura 7. Proceso de mantenimiento de ISO/IEC.

En la definición de ISO, el mantenimiento incluye las siguientes actividades:

1. *Implementación del proceso*

En esta actividad se desarrollan los planes correspondientes para llevar a cabo las actividades y tareas del mantenimiento. También debe definir los procedimientos necesarios para la gestión de problemas y petición de modificaciones.

II. Análisis de problemas y modificaciones

Esta actividad consiste en analizar los problemas o peticiones de modificación con el fin de evaluar su impacto en el sistema y la organización existentes, determinando el tipo de modificación, su alcance y su criticidad .

III. Implementación de las modificaciones

En esta actividad se incluyen todas las tareas relativas a determinar qué documentación, unidades de software y versiones deben modificarse, y se utiliza el proceso de desarrollo para implementar las modificaciones.

IV. Revisión y aceptación del mantenimiento

Esta actividad consiste en la revisión de la integridad del sistema modificado para obtener la aprobación de terminación satisfactoria de la modificación.

V. Migración

ISO especifica que se deberá asegurar que cualquier software o dato producido o modificado durante la migración se ajuste a la norma 12207.

Si se trata de una migración, el estándar aconseja desarrollar un plan de migración en el que se especifiquen al menos las siguientes cuestiones:

- Análisis de requisitos y definición de la migración.
- Desarrollo de herramientas de migración.
- Conversión del software y de los datos.
- Ejecución de la migración.
- Verificación de la migración.
- Soporte del entorno antiguo en el futuro.

También se insiste en la necesidad de notificar a los usuarios la intención de llevar a cabo la migración (describir el nuevo entorno, la fecha en que estará operativo, etc.), así como de ejecutar de forma paralela los dos entornos y de informar a los usuarios cuando se realice la migración prevista.

Por último, en el estándar se insiste en que se deberá poder acceder a los datos utilizados o asociados al antiguo entorno de acuerdo con los requisitos organizacionales para la protección y auditoría aplicables a los datos.

VI. Retirada de software

De acuerdo al estándar ISO es necesario desarrollar y documentar un “plan de retirada” que aborde cuestiones como las siguientes:

- Cese de soporte total o parcial después de un cierto tiempo.

- Archivo del producto software y su documentación asociada.
- Responsabilidad sobre cuestiones de soporte residual futuro.
- Transición al nuevo producto.
- Accesibilidad de copias de datos.

Es importante que se tenga en cuenta a los usuarios a la hora de planificar la retirada del software y que se les notifique el plan. Las notificaciones deberán incluir lo siguiente:

1. Descripción de la sustitución o actualización con su fecha de disponibilidad.
2. Informe de por qué no se soportará más el software.
3. Descripción de otras opciones de soporte disponibles una vez que se haya eliminado el soporte.

También se recomienda llevar a cabo operaciones paralelas entre el software nuevo y el retirado y proporcionar formación a los usuarios.

Cuando tenga lugar la retirada planeada, se deberá notificar a todos los involucrados. Se debe archivar, según sea apropiado, toda la documentación, ficheros y código.

Las actividades de implementación del proceso y de retirada se deben realizar obligatoriamente una sola vez por parte del mantenedor al inicio y final del servicio o proceso de mantenimiento. La actividad de migración podrá ocurrir varias veces o ninguna en momentos aleatorios, según se produzcan cambios en el entorno o no. Las tres actividades intermedias ocurren una vez por cada petición de modificación o informe de problemas recibido por el mantenedor. Estas tres actividades forman un ciclo en el sentido de que el resultado de una revisión y aceptación del mantenimiento puede conducir a nuevas peticiones de modificación o informes de problemas.

1.5 Tipos de mantenimiento

Si se analiza la definición de mantenimiento, tomando como base la que da IEEE, se puede ver que básicamente en ella aparecen reflejados los cuatro tipos de mantenimiento que existen. Cuando se refiere en la definición a corregir defectos se asocia al mantenimiento correctivo; mejorar el rendimiento u otros atributos con mantenimiento perfectivo y preventivo, y adaptar a cambios del entorno, a mantenimiento adaptativo quedando referidos los 4 tipos de mantenimiento existentes, que son: ⁱⁱⁱ

- Mantenimiento CORRECTIVO
- Mantenimiento PERFECTIVO
- Mantenimiento ADAPTATIVO

ⁱⁱⁱ Citado en (21) (5) (29)

- Mantenimiento PREVENTIVO

A continuación se hará un breve desglose sobre la finalidad que persigue cada uno de ellos según la literatura consultada.

Mantenimiento CORRECTIVO: A pesar de las pruebas y verificaciones que aparecen en etapas anteriores del ciclo de vida del software, los programas pueden tener defectos. El mantenimiento correctivo tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de causar un fallo. Este ocurre cuando el comportamiento de un sistema es diferente del establecido en la especificación. Este tipo de mantenimiento suele ser necesario en alguna de las siguientes ocasiones:

- Cuando el programa falla o aborta.
- Un programa produce un resultado que difiere con los requisitos.
- Los diseños y requisitos no están acordes con el software que los soporta.
- La documentación de usuario lleva a conclusiones erróneas al propio usuario hacia actividades que provoquen resultados incorrectos o fallos en el sistema.

La figura que se muestra a continuación realiza un análisis de los distintos las causas de los defectos según un estudio estadístico realizado. (3)

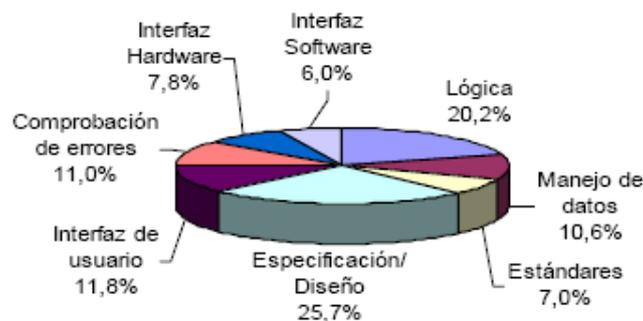


Fig. 8 Origen de los defectos del software.

Existen dos tipos básicos de mantenimiento correctivo:

- a) Reparaciones de emergencia: ejecutadas en cortos períodos de tiempo y generalmente sobre un único programa.
- b) Reparaciones planificadas: arreglan defectos que no requieren una atención inmediata y re-examinan todas las reparaciones de emergencia.

Mantenimiento ADAPTATIVO: Este tipo de mantenimiento consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el cual se ejecuta.

Estos cambios pueden afectar al sistema operativo (cambio a uno más moderno), a la arquitectura física del sistema informático (paso de una arquitectura de red de área local a Internet/Intranet) o al entorno de desarrollo del software (incorporación de nuevos elementos o herramientas como ODBC). La envergadura del cambio necesario puede ser muy diferente: desde un pequeño retoque en la estructura de un módulo hasta tener que reescribir prácticamente todo el programa para su ejecución en un ambiente distribuido en una red.

Mantenimiento PERFECTIVO: Comprende cambios en la especificación, normalmente debidos a cambios en los requisitos de un producto software. Algunos autores dividen este tipo de mantenimiento en dos:

- Mantenimiento de Ampliación: orientado a la incorporación de nuevas funcionalidades.
- Mantenimiento de Eficiencia: que busca la mejora de la eficiencia de ejecución.

Este tipo de mantenimiento aumenta cuando un producto software tiene éxito comercial y es utilizado por muchos usuarios, ya que cuanto más se utiliza un software, más peticiones de los usuarios se reciben demandando nuevas funcionalidades o mejoras en las existentes.

Mantenimiento PREVENTIVO: Este último tipo de mantenimiento consiste en la modificación del software para mejorar sus propiedades (por ejemplo, aumentando su calidad y/o su mantenibilidad) sin alterar sus especificaciones funcionales. Por ejemplo, se pueden incluir sentencias que comprueben la validez de los datos de entrada, reestructurar los programas para mejorar su legibilidad, o incluir nuevos comentarios que faciliten la posterior comprensión del programa. Este tipo de mantenimiento es el que más partido saca de las técnicas de ingeniería inversa y reingeniería.

En algunos casos se ha planteado el Mantenimiento para la Reutilización, consistente en modificar el software (buscando y modificando componentes para incluirlos en bibliotecas) para que sea más fácilmente reutilizable. En realidad este tipo de mantenimiento es preventivo, especializado en mejorar la propiedad de reusabilidad del software. (3)

El desconocimiento de las actividades que implica el Mantenimiento del Software puede inducir a minusvalorar su importancia, y se tiende a asociarlo con la corrección de errores en los programas. Por esta causa, la impresión más generalizada entre los gestores, usuarios, e incluso entre los propios informáticos, es que la mayor parte del mantenimiento que se realiza en el mundo es de tipo correctivo. Sin embargo, varios autores^{iv} exponen que esta impresión es equivocada, mostrando cómo los mayores porcentajes de esfuerzo se dedican a mantenimiento perfectivo como se muestra en la figura 9.

^{iv} Citado en (3), (3), (5)

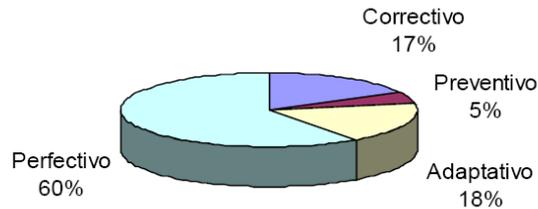


Fig. 9 Esfuerzo dedicado a cada tipo de mantenimiento.

Los tipos de mantenimiento anteriormente expresados son básicamente los que se identifican en la literatura y derivado de ellos se han propuesto algunas subdivisiones específicas.

Cuando el mantenedor recibe una petición de modificación (propuesta de cambio de un producto software que está siendo mantenido), según su naturaleza, implicará una corrección o una mejora en el software. Una corrección supone un mantenimiento de tipo correctivo, mientras que una mejora puede suponer un mantenimiento preventivo, adaptativo o perfectivo (3).

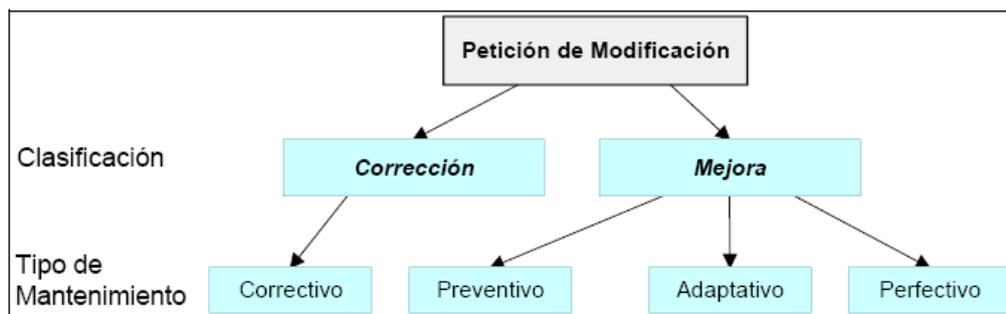


Fig. 10 Peticiones de modificación y tipos de mantenimiento

Dentro de las divisiones por tipos de mantenimiento también se pueden encontrar las categorías de mantenimiento Planificable y No Planificable como se hace uso de ello en la metodología MANTEMA (12).

El *mantenimiento No Planificable* incorpora al mantenimiento Correctivo con categoría urgente o de emergencia referido anteriormente y que consiste en localizar y eliminar los posibles defectos que bloquean el programa o los procesos de funcionamiento de la empresa. Mientras que el *mantenimiento Planificable* se refiere al mantenimiento Correctivo No Urgente o de correcciones planificadas que se refiere a localizar y eliminar los posibles defectos de los programas que no son bloqueantes. También esta categoría incluye al mantenimiento Perfectivo, que como se vio anteriormente consiste en añadir al software nuevas funcionalidades solicitadas por los usuarios. Al mantenimiento Adaptativo que modifica el software para adaptarlo a cambios en el entorno de trabajo (hardware o software) y al Preventivo que modifica el software para mejorar sus propiedades (calidad, mantenibilidad, etc.).

Conclusiones parciales

Del estudio realizado sobre la fase de Mantenimiento del Software, conceptos, características, los diferentes tipos que existen se puede concluir que:

Todo sistema software evoluciona y debe mantenerse a lo largo de su ciclo de vida. La actividad de mantenimiento estará presente durante toda la vida útil de un sistema; la mayor parte de las veces, por necesidades de mejora. Otras de las actividades durante la evolución de un sistema son la reingeniería y el abandono del software.

La reingeniería cubre necesidades de evolución más amplia de un sistema, y trabaja todos los niveles de abstracción. Esta se puede aplicar durante el Mantenimiento de Software, y busca reimplementar los sistemas para aumentar su mantenibilidad. Esta estrategia se asocia con alguna forma de mantenimiento preventivo y puede comprender la redocumentación; la organización y reestructura; la traducción a un lenguaje de programación más moderno o la modificación y actualización de la estructura y los valores de los datos del sistema. Cada una de las estrategias tiene diferentes propósitos y son complementarias entre sí. La actividad de abandono comprende desechar el software operativo que abarca determinadas funcionalidades del negocio. En el Mantenimiento de Software puede ser de tipo correctivo, preventivo, adaptativo y el que por lo general predomina es el perfectivo siendo este último donde más se hace uso de la técnica de reingeniería.

Capítulo 2 Mantenimiento de Software en la Industria

Introducción

Este capítulo está dedicado a revisar los temas del Mantenimiento de Software y su vinculación con la industria. Se analizarán algunas metodologías, procedimientos y estándares, evidenciando que el mismo, es un tema recurrente y que preocupa a cada una de las empresas que perfeccionan sus sistemas de gestión hoy en día. También se abordan elementos referentes a las características de las pequeñas organizaciones y de las aplicaciones de gestión. Se revisa la mantenibilidad como atributo de calidad más destacado a tener en cuenta durante el Mantenimiento de Software, así como otros elementos de interés. De manera general se analiza el estado del arte del Mantenimiento de Software en las organizaciones y las tendencias actuales.

2.1 Análisis de estándares y procedimientos para el mantenimiento de software

El Mantenimiento de Software es un tema que ha preocupado desde mucho tiempo a los empresas de software y usuarios que necesitan que se les adapten sus herramientas de trabajo cotidianas a sus necesidades continuamente cambiantes. Como se analizó en las definiciones de proceso de Mantenimiento de Software, este puede verse como el conjunto de todas las operaciones necesarias a realizar sobre el software para implementar las modificaciones solicitadas. Pero para dotar a este conjunto de operaciones de una base metodológica, es preciso definir con anterioridad el propio proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se lleve a cabo, sea conforme a un proceso de mantenimiento predefinido. Son varios los procedimientos, metodologías y estándares que se desarrollan en función de ello, a continuación se hará referencia brevemente a algunos de ellos, sus características y peculiaridades. Se abordarán particularmente los dos estándares definidos y conocidos dedicados al mantenimiento del software IEEE 1219 e ISO/IEC 14764. Posteriormente se hará referencia a procedimientos y metodologías que también se enmarcan en el ámbito del Mantenimiento de Software.

2.1.1 Estándar ISO 14764

Primeramente se hará referencia al procedimiento que establece la norma ISO 14764. Éste estándar internacional aclara los requerimientos para el Proceso de Mantenimiento del Software y hace referencia actividades y tareas del mantenedor dando una pequeña guía. También describe la gestión del Proceso de Mantenimiento definido en ISO/IEC 12207, y además establece definiciones para los distintos tipos de mantenimiento, y proporciona una guía aplicable a la planificación, ejecución y control, mantenimiento, revisión y evaluación y de forma cercana al proceso de mantenimiento. La limitación fundamental del mismo radica

en que describe el esqueleto del Proceso de Mantenimiento Software pero no especifica los detalles de cómo implementar o ejecutar las actividades y tareas incluidas en el proceso. La norma da una base de identificación de mantenimiento temprano como base para establecer un plan de mantenimiento temprano o elementos y requerimientos a tener en cuenta para garantizar el futuro mantenimiento del sistema desde la misma concepción del software.

Una vez definido dicho plan, propone establecer una guía para desarrollar el Mantenimiento. Describir el sistema al que se le dará soporte: se especifican todos los detalles del sistema a mantener. Identificar el estado inicial del software: eso para saber cuáles son los cambios nuevos realizados. Describir qué soporte es necesario: esto para facilitar el comienzo del desarrollo del Mantenimiento de Software. Identificar la organización que debe hacer el soporte/mantenimiento: para contemplar el objetivo del mantenimiento en el proceso de desarrollo del software. Describir cualquier acuerdo entre cliente y vendedor: Se debe tener claro lo que quiere el cliente pero por escrito, de este modo el vendedor sabe lo que es lo que tiene que hacer para satisfacer el cliente.

Este estándar define los siguientes tipos de mantenimiento: correctivo, perfectivo, adaptativo y preventivo.

Estos son los aspectos fundamentales en cuanto a la estrategia de mantenimiento que propone la norma. Ahora se revisará lo concerniente al proceso de mantenimiento que plantea la norma que es elemento de interés fundamental. Las actividades que comprende el Proceso de Mantenimiento son:

- Implementación del Proceso
- Análisis de Modificaciones y Problemas
- Implementación de Modificaciones
- Revisión/Aceptación del Mantenimiento
- Migración
- Retiro

Detallando brevemente cada una de estas actividades, las cuales están enfocadas con una visión de proceso, se tiene que:

Implementación del Proceso: Se refiere a como el encargado del mantenimiento establece los planes y procedimientos a ejecutar durante el Proceso de Mantenimiento. El encargado del mantenimiento debería establecer también los nexos necesarios con la entidad u organización que desea que se le realice el mantenimiento. Esta actividad define como entradas necesarias (La Línea seguida anteriormente, Documentación del Sistema, Una solicitud de modificación (MR) o informe de problema (PR)), las tareas a realizar (Desarrollar Planes y Procedimientos de Mantenimiento, Establecer procedimientos para MR/PR, Implementar la gestión de la configuración), controles, soporte, Salidas (El Plan de

Mantenimiento, Procedimientos para el mantenimiento, Procedimientos para la resolución de problemas, Planes para la realimentación del usuario, El Plan de Transición, Plan para la Gestión de la Configuración).

Análisis de Modificaciones y Problemas: Esta actividad se destaca como crítica debido a la importancia que tiene entender el problema, desarrollar una solución y obtener la aprobación para poder desarrollarla. Como entrada necesaria de esta actividad se destaca un informe de problema o petición de modificación validada, además de documentación sobre el Proyecto/Sistema y la documentación de requerimientos. Las tareas a seguir, (Análisis de las MR/PR's: Determinar el tipo de cada una de las peticiones y modificaciones, alcance y criticidad). Se plantea la verificación como un medio de validación de los reportes de problemas. Las opciones como alternativas a las solicitudes de modificaciones y problemas presentados, documentar dicha propuesta de opciones, posteriormente aparece la aprobación de las opciones presentadas anteriormente, indicando los controles y el soporte para ello, y por último las salidas que incluyen un análisis de Impactos de los diferentes cambios en forma de gasto de tiempo/dinero, la Opción Recomendada, la modificación Aprobada. Documentación Actualizada.

Implementación de la Modificación: En esta actividad el mantenedor desarrolla y prueba la modificación del producto software. Como entradas de esta actividad se definen línea base a seguir, la MR/PR aprobada, la documentación de la modificación aprobada. En las tareas a realizar el mantenedor realiza un análisis, y después lleva a cabo el proceso de desarrollo del mantenimiento. También se desarrollan actividades de control y soporte y como salidas de esta actividad Planes y Procedimientos para las Pruebas Actualizados, Documentación Actualizada, Código Fuente modificado, Informe de Pruebas, Métricas.

Aceptación/Revisión del Mantenimiento: Esta actividad asegura que las modificaciones al sistema se han hecho de forma correcta y de acuerdo a los estándares aprobados dentro del uso de una metodología adecuada. Como entradas se tienen el software modificado y los resultados de las pruebas de la modificación. Dentro de las tareas están las revisiones que se dirigen de forma que se asegure que las modificaciones son correctas y que se obtenga un final satisfactorio de la modificación y la aprobación para completar de forma satisfactoria la modificación tal como se especifica en el contrato. Hay actividades de control y de soporte (este último asociado a Proceso de Aseguramiento de la Calidad, Proceso de Verificación, Proceso de Validación, Proceso de Revisión Conjunta, Proceso de Auditoría). Como salidas de esta actividad se tienen: nueva línea base, incorporando las modificaciones aceptadas, modificaciones rechazadas, informe de aceptación, informes de revisión y auditoría, informe de pruebas de cualificación del software.

Migración: Se refiere a determinar las acciones necesarias para conseguir la migración y a partir de ahí desarrollar y documentar los pasos necesarios para efectuar la migración. Como entradas de esta actividad se tienen el antiguo entorno, el nuevo entorno, la antigua

línea base, la nueva línea base. Las tareas de la migración son llevadas a cabo desarrollando un plan de migración, notificando a los usuarios la migración, proporcionando entrenamiento, avisando del término de la migración, calculando el impacto del nuevo entorno y archivando datos. Se efectúan actividades igualmente de control y soporte. Y como salidas de esta actividad plan de migración, herramientas de migración, notificación de intentos, producto software migrado, notificación de finalización y datos archivados.

Retiro del software: Una vez que el producto ha alcanzado el final de su vida útil debe retirarse. Se debe hacer un análisis para ayudar en la toma de la decisión de retiro de un producto software fundamentalmente basado en aspectos. Como entradas de esta actividad está el producto software a retirar, el nuevo producto software, el antiguo entorno. Como tareas para realizar el retiro el mantenedor desarrolla un plan de retiro, notificando a los usuarios dicho retiro, notificando la finalización de la actividad de retiro y archivando los datos. Se llevan a cabo actividades de control y soporte para ello. Y finalmente como salida de esta actividad plan de retiro, notificación de intento, resultados del retiro, personas entrenadas, producto software retirado, notificación de finalización, línea base del producto retirado archivada.

Estas actividades descritas son las que recogen la norma ISO 14764 en su procedimiento para realizar el Mantenimiento de Software y se publicó en 1998. Como todos los estándares, el acceso de su lectura está restringido a aquellos que son miembros de la organización, o pagan por hacerlo. (13)

A pesar de tener un enfoque dirigido a una tercerización del mantenimiento, esta norma identifica adecuadamente el que hacer en las actividades y tareas a desarrollar en el proceso de mantenimiento y serán tenidas en cuenta a la hora de realizar la propuesta de procedimiento.

2.1.2 Estándar IEEE 1219

A continuación se realizará un breve análisis del estándar IEEE 1219, hasta 1998, era el único estándar que íntegramente se ocupa del proceso de Mantenimiento del Software. En él se detalla un proceso iterativo para gestionar y realizar las actividades de mantenimiento.

Este estándar divide en varias fases al proceso de mantenimiento definiendo cambios en un producto software. Dicho proceso es iterativo y en cascada, con una gran semejanza al ciclo de vida del desarrollo clásico. Las fases mencionadas se detallan a continuación:

- Identificación y Clasificación del Problema o de la Modificación.
- Análisis.
- Diseño
- Implementación.

- Pruebas del Sistema.
- Pruebas de Aceptación.
- Liberación del Producto.

A continuación se hará referencia brevemente a cada una de estas fases, detallando en qué consiste cada una de ellas.

Inicialmente, antes de entrar en las fases correspondientes, el estándar plantea la realización de una solicitud formal de modificación, que tiene asociados formularios e informes asociados a los fallos. Esta solicitud constituye el paso inicial para la entrada en la primera fase planteada por dicho estándar.

Identificación y Clasificación del Problema o de la Modificación: Se refiere a que se identifiquen, se clasifiquen y se le asigne una prioridad a cada una de las posibles modificaciones a realizar en el software. Esta clasificación permitirá posteriormente hacer agrupamiento de dichas modificaciones. Con la Solicitud de Modificación, comienza el Mantenimiento del Software. En esta fase los procedimientos a seguir son:

- Asignación de un Número de Identificación.
- Clasificación del tipo de mantenimiento.
- Análisis de la modificación para determinar si se acepta, se deniega o se evalúa.
- Realizar una estimación preliminar de la magnitud de la modificación.
- Priorizar la modificación.
- Asignar Solicitudes de Modificación a bloques de tareas planificadas para su implementación.

Los pasos en procedimiento son discutidos, analizados y aprobado por el equipo de desarrollo de los en reuniones de revisión. Al finalizar esta fase se tendrá:

- Relación de los problemas o nuevos requerimientos.
- Evaluación de los problemas o nuevos requerimientos.
- Clasificación del tipo de mantenimiento solicitado.
- Prioridad inicial.
- Datos de verificación (para el mantenimiento correctivo)
- Estimación inicial de los recursos necesarios para modificar el sistema existente.

Se hace énfasis que los elementos asociados a la estimación de costo inicial pues están sujetos a cambios debido al propio carácter iterativo del proceso.

Análisis: Se corresponde con la viabilidad de generar en función de las solicitudes de modificación varios requerimientos funcionales, que a su vez pueden ser descompuestos

en requisitos de software. En esta fase también se realiza un estudio de la viabilidad y el alcance de las modificaciones, y se genera un plan preliminar de diseño, implementación, pruebas y liberación del software. Se hace referencia a aspectos a tener en cuenta por parte de los analistas en las etapas de esta fase entre los que se pueden mencionar análisis de viabilidad, análisis detallado que incluye examinar todos los elementos que están afectados, realizar una estrategia de pruebas, desarrollar un plan de implementación preliminar de manera que se prevea el menor impacto posible para los usuarios del sistema. Al finalizar esta fase se tendrá como repositorio:

- Informes de viabilidad para las Solicitudes de Modificación recibidas.
- Informe de análisis detallado
- Actualización de los requisitos
- Lista de modificaciones preliminares
- Estrategia de pruebas
- Plan de implementación.

Diseño: Para la realización del diseño se parte de la documentación existente en el proyecto y del sistema. Se realiza una identificación de los módulos software que van a ser objeto de modificación, con el fin de hacer constar la planificación de tareas y ver la previsión de las mejoras a introducir documentando los mismos. Se vela por que se cumplan estándar y se tienen presentes elementos de robustez, seguridad realizando una adecuada documentación de la modificación. Como cierre de la fase de implementación se tiene:

- Lista de modificaciones revisada.
- Guía básica del diseño actualizada.
- Planes de pruebas actualizados.
- Análisis detallado actualizado.
- Requisitos verificados.
- Plan de implementación revisado.
- Lista de restricciones y riesgos bien documentados.

En esta fase también se vela por el aseguramiento de la fiabilidad y mantenibilidad del software, previendo que las decisiones de diseño sobre las modificaciones de módulos software pueden afectar a la calidad del propio software.

Implementación: Se parte de la información generada en las fases anteriores. En esta fase, se van a seguir unos determinados procesos que serán iterativos, y gradualmente

incrementales, es decir, se irán repitiendo y desarrollando en mayor detalle, hasta obtener el resultado previsto por la organización. Estos procesos son:

- Codificación y pruebas de unidad.
- Integración.
- Análisis de riesgo y revisión.
- Revisión de disponibilidad de pruebas.

Destacar que en las revisiones y las pruebas del software, pueden aparecer riesgos debido a los cambios que se van a producir en el software existente. Estos riesgos deben ser identificados y documentados e incluso resueltos cuando proceda.

Por último, el resultado de la fase de implementación, será un nuevo software que deberá estar bajo el control de la Gestión de Configuración del Software (SCM, Software Configuration Management), al que se acompañará de toda la documentación actualizada de diseño, documentación de pruebas, documentación de usuario y material de aprendizaje, sin olvidar la declaración de riesgos e impacto para los usuarios y el informe de revisión de la disponibilidad de pruebas.

Pruebas del sistema: Refiere que las pruebas deben ser realizadas por entidades independientes ajenas a los desarrolladores y diseñadores del software y que deben estar presente los clientes y usuarios finales. Disponiendo de la documentación generada en las fases anteriores. Esta fase ha de llevarse un sistema completamente integrado. Los procesos que han de realizarse durante las pruebas de sistema son:

- Pruebas de funcionalidad del sistema
- Pruebas de interface
- Pruebas de regresión
- Revisión de la disponibilidad de pruebas del sistema para valorar el nivel de preparación para las pruebas de aceptación.

Asociado a la realización de las pruebas se adjudica un conjunto de documentos que indican el estado del criterio que se ha establecido en el plan de pruebas y sus resultados, permitiendo obtener así un sistema completamente integrado y probado.

Pruebas de aceptación: Se realizan por el cliente, por el usuario o por un tercero designado por el cliente. Se llevan a cabo para asegurar que el resultado de las modificaciones es satisfactorio para el cliente, tanto del software como de la documentación generada.

Liberación del producto: Aquí se indica los pasos necesarios para liberar el producto:

- Conducir una Auditoría de Configuración Física.
- Notificar a la comunidad de usuarios.

- Desarrollar una versión de archivo del sistema para salvaguarda del mismo.
- Realizar la instalación y formación para el cliente. Se ha de proveer del material de sistema necesario para facilitar la utilización a los usuarios.
- Se ha de completar la Documentación de Descripción de la Versión.
- Hay que actualizar el estado de la base de datos de cuentas y auditoría.
- Y finalmente, hay que poner todo bajo el control de la Gestión de Configuración del Software.

Se sugiere para la instalación del sistema planificar y documentar los procedimientos de instalación alternativos, que podrían asegurar el mínimo impacto sobre los usuarios del sistema debido a fallos del software imprevistos, no detectados durante las pruebas.

Esta norma plantea un proceso de mantenimiento con gran nivel de detalle y documentación a llevar para su desarrollo, haciéndolo muy útil y necesario sobretodo en las grandes industrias de software donde 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. La misma define y estructura de manera interesante las actividades a desarrollar en la actividad de mantenimiento que será objeto de base para la definición del procedimiento a proponer.

Una vez revisado estos dos estándares sobre el Mantenimiento de Software, se puede apreciar que el enfoque que plantean está dirigido a equipos de desarrollo vinculados a la industria de software donde cada actividad es debidamente documentada y aprobada por los roles correspondientes, llevando todo un control de cada una de ellas en el momento preciso. También en las actividades que plantean para la realización del mantenimiento se puede apreciar un orden lógico y secuencial de las mismas que va desde la identificación de los problemas hasta la retirada del software que será sustituida por su nueva versión, resultado del mantenimiento. Ambos estándares destacan el costo del mantenimiento como un elemento a tener en cuenta a la hora de su realización, fundamentalmente por la cantidad de recursos que hay que poner en función de ello.

2.1.3 Modelo de Madurez de Mantenimiento de Software (S3M)

También enmarcado en el contexto actual del Mantenimiento de Software se encuentra el Modelo de Madurez de Mantenimiento de Software (S3M) que se basa en el Modelo de Madurez de Capacidades Integradas (CMMI) del Instituto de Ingeniería de Software (SEI). El S3M debe ser visto como complemento de CMMi y el fin de evaluar y mejorar la función de una organización dedicada al proceso de Mantenimiento de Software en una organización.

En este modelo se presenta a un inventario exhaustivo de los procesos de mantenimiento y sus actividades. También se identifica que la representación actual del Mantenimiento de Software en las normas internacionales (ISO12207 y el ISO14764) cubre sólo parcialmente sus procesos y actividades. Además confirma que no hay ninguna propuesta de modelo de madurez que cubra el conjunto de procesos y actividades del Mantenimiento de Software y propone un modelo más completo para su función.

El S3M se basa en el concepto de una faceta al igual que CMMi. Una faceta es un conjunto de prácticas relacionadas que se centra en un área de organización o de necesidad, o un elemento específico en el proceso de Mantenimiento de Software. Cada faceta representa una capacidad importante para una organización de Mantenimiento de Software. Dentro de un aspecto determinado, el nivel de práctica se basa en su respectivo grado de madurez de capacidad. Las prácticas más fundamentales se encuentran en un nivel inferior, mientras que los más avanzados se encuentran en un nivel más alto.

El S3M detallado incluye 4 Proceso de Dominios, 18 Áreas de Procesos Claves (KPA), de 74 facetas y 443 prácticas. Mientras que algunos KPA son exclusivos de mantenimiento, algunos otros fueron derivados de CMMi y otros modelos y ligeramente modificados para asignar más de cerca a las características de mantenimiento diario

Cada uno de los 6 niveles de madurez de capacidad puede ser caracterizada, en el modelo S3M asociados al proceso de Mantenimiento de Software, los cuales van desde el nivel 0 catalogado como Incompleto donde no se evidencia que dicho proceso se ejecute en la organización, hasta el nivel 5 catalogado como Proceso Optimizado estando cuantificado los objetivos del proceso lográndose una mejora continua del mismo dentro de la organización. Por cada uno de los niveles se definen atributos genéricos que los caracterizan y que se pueden evidenciar dentro de la organización que desarrolla la actividad de Mantenimiento de Software.

2.1.4 Metodología MANTEMA

Es una metodología para realizar una gestión estructurada del Mantenimiento de Software basada en ISO/IEC 12207 y desarrollada en la universidad española Castilla la Mancha. Según el grupo de desarrollo de la misma, una metodología en un sentido habitual se refiere a una serie de métodos o técnicas relacionadas. Según OPEN, una metodología tiene un modelo de proceso, un conjunto de técnicas, un conjunto de entregables, guías para la gestión del proyecto, lo que incluye roles, definición de una estructura del equipo, etc. Un conjunto de métricas y herramientas. (14)

Esta metodología realiza un análisis de los tipos de mantenimientos clasificándolos fundamentalmente en 2 niveles de anidamientos referidos a:

No Panificable (NP):

- Correctivo Urgente (UC): localizar y eliminar los posibles defectos que bloquean el programa o los procesos de funcionamiento de la empresa.

Panificable (P):

- Correctivo No Urgente (NUC): localizar y eliminar los posibles defectos de los programas que no son bloqueantes.
- Perfectivo (PER): añadir al software nuevas funcionalidades solicitadas por los usuarios.
- Adaptativo (A): modificar el software para adaptarlo a cambios en el entorno de trabajo (hardware o software).
- Preventivo (PRE): modificar el software para mejorar sus propiedades (calidad, mantenibilidad, etc.).

La macroestructura de la metodología consta de 3 fases fundamentales: Definición del Plan de Mantenimiento, Ejecución de la Intervención y la Retirada y Fin. Dentro de las especificaciones encontradas en la primera fase se considera la variante del outsourcing, la que consiste en ceder a un tercero la ejecución de determinados procesos software que, en general, no forman parte del “core-business”.

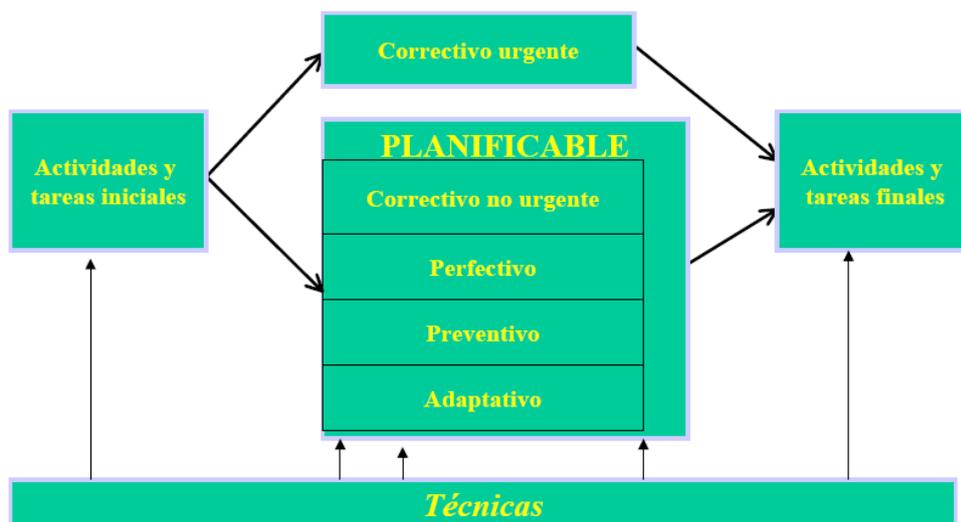


Fig. 11 Estructura general del modelo de procesos de la metodología MANTEMA

En el estudio inicial plantea como actividades la recogida de información mediante la solicitud de prestación del servicio de mantenimiento, se realiza un cuestionario Inicial. Posteriormente se prepara la propuesta y se establece el contrato que va a regir la actividad del mantenimiento. Una vez establecida la planificación se da lugar a la realización del Mantenimiento de Software en cuestión y como actividades finales se realiza un registro de la intervención, se procede a la recolección de métricas y se da fin de la internalización referido a la entidad outsourcing vinculada a la tarea de mantenimiento. Dicha metodología esta soportada por el software MANTIS que la automatiza. De manera general esta metodología se presenta proponiendo un proceso de mantenimiento mucho

más controlado y disciplinado, con numerosas políticas y normas, con cierta resistencia a los cambios (15). Esto último hace que sea vista como una metodología pesada para aplicarla en pequeñas organizaciones para la realización del Mantenimiento de Software en las mismas pero sin embargo su estructura de agrupación de actividades y la propuesta de tipos de mantenimiento sin incluir el correctivo urgente serán tenidas en cuenta para la propuesta de procedimiento a realizar.

2.1.5 Ágil_MANTEMA

Ágil_MANTEMA está creado a partir de la agilización de la metodología MANTEMA que muestra la visión del proceso de mantenimiento desde el mayor nivel de abstracción. Ágil_MANTEMA está enfocada a pequeñas organizaciones y pretende definir un proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, buscando guiar paso a paso el proceso de Mantenimiento de Software para este tipo de organizaciones. La misma está basada en la filosofía de las metodologías ágiles, en particular en SCRUM.

Se establecen tres niveles de capacidad para el proceso de mantenimiento, según interfaces de soporte al cliente, gestión de resolución de problemas, gestión de la configuración, aseguramiento de la calidad, gestión de cambio de requisitos y gestión de proyectos. A pesar de ser una metodología mucho más ligera dedicada a simplificar los elementos burocráticos y generales de MANTEMA presenta, al igual que esta, un enfoque de externalización del Mantenimiento de Software. Va dirigida a organizaciones que brindan servicios de mantenimiento a otras organizaciones por lo que contemplan elementos de financiación del proyecto, retorno de la inversión, lanzamiento del proyecto. Se ven de manera separadas la figura del equipo mantenedor y la del cliente que contrata el servicio de mantenimiento evidenciando que representan organizaciones diferentes. Dentro de las actividades a desarrollar en la fase de mantenimiento se propone la realización de sucesivos controles del trabajo mediante reuniones habituales y seguimiento de cambios. Si se tiene en cuenta que estos elementos se desarrollan no entre 2 organizaciones, una como consumidora y otra proveedora de un servicio, sino a lo interno de una misma organización, no sería necesario definir las dentro de la metodología, estas estarían implícitas en propio proceder del procedimiento.

2.1.6 Otras metodologías y procedimientos

Como documentación asociada al Mantenimiento de Software también se encontró el Procedimiento *Mantenimiento de Sistemas de Información* de la delegación Gustavo Medero de la Ciudad de la Esperanza de México. El mismo estima un proceso donde se describen las tareas a acometer especificando la unidad administrativa encargada, la actividad propia a desarrollar y el tiempo estimado para ello. Se grafican las actividades propias a realizar por cada una de las áreas correspondientes. Este procedimiento se

enfoca en su descripción como la institución que lo adopta es una entidad que brinda servicio de mantenimiento a sistemas de información y por cada solicitud que se genere se procede a crear un proyecto, por lo que no tiene en cuenta que las fuerzas de realización del mantenimiento se encuentran dentro de la empresa que lo requiere y además no especifica cuáles son los resultados concretos de cada uno de las actividades y no cuenta con artefactos que permitan documentar a la par que se le realiza la labor del mantenimiento al sistema en cuestión.

Métrica versión 3 es una metodología de Planificación, Desarrollo y mantenimiento de sistemas de información. Fue desarrollada por el ministerio de Administraciones Públicas en España y es de uso obligado a nivel interno y para todos los contratos con el gobierno. En el caso del proceso de mantenimiento incluye actividades como Registro de la petición, Análisis de la petición, preparación de la implementación de la modificación, Seguimiento y evaluación de los cambios hasta la aceptación. Estas actividades a su vez incluyen otros elementos propios de las mismas. Métrica estudia tan sólo el Mantenimiento Correctivo y Evolutivo de un sistema de información, pero no el Mantenimiento Adaptativo y Perfectivo, siendo estos últimos los tipos de mantenimiento más comunes a realizarse normalmente en las actividades de Mantenimiento de Software habituales.

En la revisión realizada en este epígrafe se puede constatar que la referencia de estándares, modelos y metodologías para realizar la actividad de Mantenimiento de Software por la magnitud y envergadura de las actividades que proponen, está dirigida fundamentalmente a las grandes industrias de software donde se brinda el Mantenimiento de Software como servicio a diferentes organizaciones. En su mayoría cuando detallan las actividades se centran fundamentalmente con cierta similitud entre ellas en Implementación del proceso, Análisis de problemas y modificaciones, Realización de las Modificaciones, Revisión y aceptación del mantenimiento así como Migración y Retirada proponiendo realizar las tres intermedias de manera cíclica tal como se muestra en el anexo 8.

2.2 Características de las pequeñas organizaciones

La mayor parte de las empresas y organizaciones cubanas se pueden catalogar como pequeñas organizaciones, siguiendo algunas clasificaciones dadas de acuerdo al número de trabajadores, y que establecen que las empresas u organizaciones pequeñas son aquellas que tiene de 11 a 50 trabajadores como se plantea en (16).

Estudios realizados por el Centro de Estudios de Ingeniería de Sistemas (CEIS) en empresas nacionales, arrojaron que incluso en la industria cubana de software está compuesta fundamentalmente por pequeñas y medianas empresas, como se ilustra a continuación según encuesta realizada en (16).

Total de miembros en los equipos de proyectos

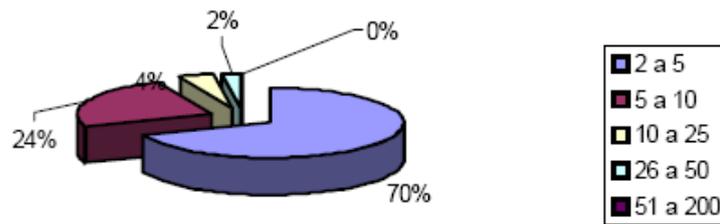


Fig. 12 Total de miembros de los equipos de proyectos de las empresas de ICSW

Para lograr un mayor desempeño de su ámbito de negocio, estas organizaciones se apoyan en sistemas de software que le permiten llevar a cabo una mejor gestión dentro de su área de negocio.

Dentro de estas organizaciones existe un área de tecnología de información que cuenta con el personal encargado de la atención de las necesidades de informatización de la organización. Como indicó la encuesta realizada, por lo general se destina dentro de la empresa a la actividad de mantenimiento de estos sistemas a una persona, como se ilustra en la figura 3.

En la Industria de software, como se revisó en los epígrafes anteriores, existen procedimientos, metodologías y estándares documentados que guían el proceso de mantenimiento dirigido a equipos de desarrollo y mantenimiento conformados por varios integrantes donde cada uno juega un rol muy bien identificado. Esto deja un tanto al margen a pequeñas organizaciones cuyo propio personal informático asume la actividad de mantenimiento a sistemas de gestión que cubren procesos propios de negocio de dichas instituciones.

2.3 Características generales de los Software de Gestión

Para el control de la información dentro de una entidad pueden existir tantas herramientas de gestión como las áreas de conocimiento para las que hayan sido pensadas y desarrolladas. Sin embargo tienen un fin en común: pretenden dar solución mediante la informática a un problema en particular. El objetivo de los sistemas de gestión es integrar y organizar la estructura de una empresa, mediante el registro de las transacciones diarias y la generación de reportes que presentan información. En el trabajo cotidiano es común contar con herramientas para gestionar la administración de la oficina, como lo son las planillas de cálculo, los procesadores de texto, software para presentaciones. Sin embargo, en ocasiones, estas herramientas no bastan porque la complejidad intrínseca de la organización requiere algo más. Lo que se necesita es una gestión o administración de sistema independiente del usuario que permita mantener la integridad de los datos y la información y el control de los accesos de los usuarios a ella. Estas funciones sólo pueden ser llevadas a cabo si el sistema está respaldado por un sistema de Base de Datos que

generalmente utiliza la tecnología Cliente – Servidor y está estructurado mediante distintos módulos que permite que el usuario interactúe con dicha aplicación, es esto lo que se conoce como sistemas o software de gestión.

De manera general se puede mencionar entre las características generales del software de gestión:

- Están estructurados por módulos que abarcan las distintas funcionalidades específicas.
- Abarcan un área funcional dentro de una organización.
- Registran información propia de su negocio y la recuperan mediante informes.
- Realizan operaciones CRUD sobre la información que almacenan.

Las empresas y organizaciones comúnmente poseen su propio departamento de sistemas. En este caso la compañía desarrolla sus propios sistemas con su propio personal, lo que generalmente se conocen como sistemas hechos a medida (18), es decir, se trata de sistemas desarrollados en forma interna o externa a la organización, que han sido realizados en base a los requerimientos específicos del negocio.

2.4 Herramienta de Soporte

A la hora de desarrollar y definir procedimientos, modelo de procesos o metodologías es importante la estructura y formatos en la cual las mismas estarán soportadas. Por lo general tradicionalmente las mismas eran descritas sobre la base de archivos de texto. Este formato permite únicamente que sea destinado para su lectura (personas, no sistemas). Con las necesidades de integración e interoperabilidad surge una nueva área que es la enfocada en la ingeniería de procesos de software donde se han desarrollado herramientas para soportar procesos siendo estos escritos en XML siguiendo el estándar SPEM con el cual es posible realizar cualquier tipo de procesamiento automático que interese; por ejemplo, para generar una web con información de todos los procesos, tareas, roles, etc. Esta última posibilidad permite proveer a los desarrolladores una base de conocimiento de capital intelectual que puede ser usada como referencia y material educativo y que les permita buscar, administrar y desplegar contenido.

Eclipse Process Framework Composer (EPF Composer), una herramienta libre soportada por la comunidad Eclipse. Esta herramienta se ha diseñada para definir, documentar y publicar procesos (19). La misma facilita el acceso y economiza los tiempos necesarios para comprender la información haciendo que los equipos de trabajo agilicen más en la capacitación de los nuevos miembros. La necesidad de uso y aplicación de una herramienta que cubra este entorno se debe a:

- Los equipos de desarrollo necesitan un acceso fácil y centralizado a la información.

- Los contenidos de los procesos de desarrollo deben estar en formatos estándar que permitan una fácil integración.
- Los equipos de desarrollo requieren una base de conocimiento actualizada para que ellos mismos aprendan sobre métodos y mejores prácticas.
- Los equipos de desarrollo necesitan soporte para dimensionar correctamente sus procesos.
- Los equipos de desarrollo requieren la habilidad de estandarizar prácticas y procesos dentro de las organizaciones.
- Los equipos de desarrollo necesitan cerrar la brecha entre la ingeniería de procesos y el establecimiento de los mismos en las organizaciones por medio del uso de representaciones y terminologías similares.

2.5 Calidad en el Mantenimiento de Software

El control de calidad juega un papel muy importante dentro de las organizaciones ya que permite controlar los parámetros críticos de los procesos que tienen defecto en la calidad y desempeño del producto.

Un hito en la definición de estándares de calidad de producto software, destinado a evaluación, fue cuando en 1992 se aprobó el estándar de calidad y el proceso de evaluación ISO/IEC llamado "ISO 9126: Software Product Evaluation: Quality Characteristics and Guidelines for their Use".

Este nuevo estándar ISO 9126 define un modelo de calidad del software en el que *la calidad se define como la totalidad de características relacionadas con su habilidad para satisfacer necesidades establecidas o implicadas.*

El estándar ISO/IEC 9126 prescribe seis características que describen a la calidad de software. También informa acerca de un conjunto de subcaracterísticas de calidad para cada característica en particular como se muestra en la figura 13.

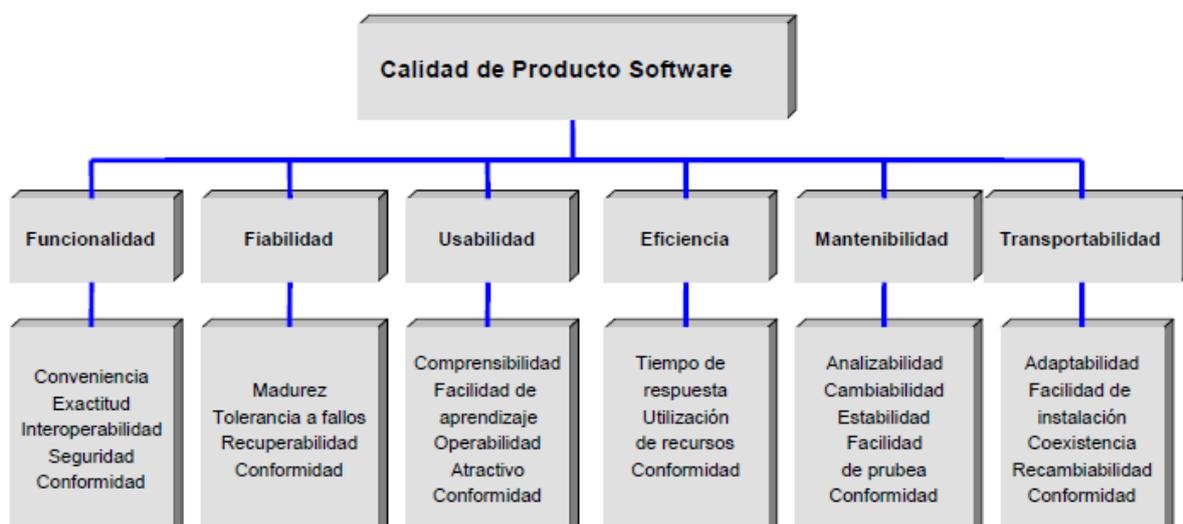


Fig. 13 Calidad del Producto de Software (Modelo 9126)

Una de las características que denota la calidad del producto es la mantenibilidad. El IEEE (1990) define mantenibilidad como: *“La facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarse a cambios en el entorno”*. (20)

Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requerimientos o en las especificaciones funcionales. Es el atributo de calidad del software que más directamente influye en los costes y necesidades del mantenimiento, debido a que a mayor mantenibilidad (facilidad de mantenimiento), menores costes de mantenimiento, y viceversa.

Existen unos pocos factores que afectan directamente a la mantenibilidad, de forma que si alguno de ellos no se satisface adecuadamente, ésta se resiente. Los tres más significativos son:

- Proceso de desarrollo: la mantenibilidad debe formar parte integral del proceso de desarrollo del software. Las técnicas utilizadas deben ser lo menos intrusivas posible con el software existente. Los problemas que surgen en muchas organizaciones de mantenimiento son de doble naturaleza: mejorar la mantenibilidad y convencer a los responsables de que la mayor ganancia se obtendrá únicamente cuando la mantenibilidad esté incorporada intrínsecamente en los productos software.
- Documentación: En múltiples ocasiones, ni la documentación ni las especificaciones de diseño están disponibles, y por tanto, los costes del mantenimiento se incrementan debido al tiempo requerido para que un mantenedor entienda el diseño del software antes de poder ponerse a modificarlo. Las decisiones sobre la documentación que debe desarrollarse son muy importantes cuando la responsabilidad del mantenimiento de un sistema se va a transferir a una organización nueva.
- Comprensión de Programas: La causa básica de la mayor parte de los altos costes del Mantenimiento de Software es la presencia de obstáculos a la comprensión humana de los programas y sistemas existentes. Estos obstáculos surgen de tres fuentes principales:
 - La información disponible es incomprensible, incorrecta o insuficiente.
 - La complejidad del software, de la naturaleza de la aplicación o de ambos.
 - La confusión, mala interpretación u olvidos sobre el programa o sistema.

Dependiendo de cómo se haya construido el software se puede aumentar la mantenibilidad. Más concretamente, se han identificado los factores específicos que influyen en la mantenibilidad:

- Falta de cuidado en las fases de diseño, codificación o prueba.
- Pobre configuración del producto software.
- Adecuada cualificación del equipo de desarrolladores del software.
- Estructura del software fácil de comprender.
- Facilidad de uso del sistema.
- Empleo de lenguajes de programación y sistemas operativos estandarizados.
- Estructura estandarizada de la documentación.
- Documentación disponible de los casos de prueba.
- Incorporación en el sistema de facilidades de depuración.
- Disponibilidad del equipo (computador y periféricos) adecuado para realizar el mantenimiento.
- Disponibilidad de la persona o grupo que desarrolló originalmente el software.
- Planificación del mantenimiento.

Existen maneras de medir la mantenibilidad para todos los elementos software que están o estarán sometidos a mantenimiento: código, documentos de usuario, documentos de análisis o diseño, etc.

En los últimos años se ha investigado sobre modelos de regresión polinomial que predicen la mantenibilidad del software. Para ello, se utiliza una combinación de variables de predicción en una ecuación polinomial para definir un Índice de Mantenibilidad (IM) mide la facilidad de mantenimiento del producto considerado.

El Índice de Mantenibilidad (IM) mide la facilidad de mantenimiento del producto considerado.

Welker, en 1995, definió el concepto de índice de mantenibilidad IM para intentar cuantificar la mantenibilidad de un sistema. Este concepto ha sido perfeccionado por múltiples autores a lo largo de estos años. (20)

El IM expresa una fuerte correlación entre la métrica Halstead, la complejidad ciclomática, las líneas de código, y el número de observaciones al sistema que es objeto del Mantenimiento de Software.

El IM de un conjunto de programas más básico es un polinomio de la siguiente forma:

$$\mathbf{IM = 171 - 5.2 * \ln(aveV) - 0.23 * aveV(g') - 16.2 * \ln(aveLOC) + 50 * \sin(\sqrt{2.4 * perCM})}$$

Donde:

aveV: es la media del volumen por módulo según Halstead..

aveV(g'): es la media de la complejidad ciclomática por módulo, que mide el número de caminos linealmente independientes a través de un módulo de programa.($M = \text{Número de condiciones} + 1$)

aveLOC: es la media del número de líneas de código por módulo.

perCM: es la media porcentual de líneas de código comentadas.

Para realizar un análisis de si se ha elevado o no la mantenibilidad de un sistema tributando a su calidad se debe comparar el indicador IM, cuanto mayor sea el IM, mayor mantenibilidad tendrá el sistema. Un experimento aplicado a Hewlett-Packard estableció dos puntos de corte para el análisis de la calidad de los sistemas con las métricas del polinomio. Los valores superiores a 85 indican que el software es muy mantenible, valores entre 85 y 65 sugieren moderado de mantenimiento, y los valores por debajo de 65 indican que el sistema es difícil de mantener.

Conclusiones Parciales

La industria, incluso la de software a nivel nacional está formada mayoritariamente por pequeñas organizaciones y empresas sin embargo los estándares con mayor aceptación internacional están dirigidos a las grandes empresas productoras de software.

En la revisión realizada por los diferentes estándares, modelos y metodologías para realizar el mantenimiento se comprobó que aunque abarcan todas las actividades definidas en el proceso de mantenimiento estos están dirigidos a la industria desarrolladora de software y a empresas que prestan servicio de contratación de mantenimiento refiriéndose en mayor medida a la externalización del mantenimiento. Estos estándares, procedimientos y metodologías se presentan con un enfoque general y abarcador expresados en su mayoría hacia la tercerización del mantenimiento y a entidades que cuenten con un equipo de desarrollo multidisciplinario. Este análisis evidencia que las organizaciones nacionales están desprovistas de un Procedimiento de referencia para la definición, adiestramiento y ejecución de las actividades asociados al proceso de Mantenimiento de Software.

Se revisó las características de la herramienta EFP Composer que permite documentar las metodologías y procedimientos siendo de fácil acceso y alta navegabilidad, permitiendo una rápida exploración para encontrar el contenido que se desea en poco tiempo.

Se propone para medir la mantenibilidad del sistema como característica de calidad de software el uso del Índice de Mantenibilidad.

Capítulo 3 Propuesta de procedimiento para el Mantenimiento de Software

Introducción

En este capítulo se describe la propuesta de un procedimiento de Mantenimiento de Software para equipos que estén integrados por pocos miembros, de hasta 2 ó 3 personas encargados de realizar dicha actividad dentro del ciclo de vida del software. Este procedimiento está basado en prácticas establecidas en la actualidad y sobre la referencia de procedimientos, estándares y normas revisados en el capítulo anterior para la realización de esta actividad. Se sustenta teniendo en cuenta el modelo general del proceso de mantenimiento ya revisado en la norma ISO/IEC 12207, combinando elementos de algunos de los procedimientos existentes, y ajustándolo a las características de las pequeñas organizaciones y empresas nacionales como escenario particular.

3.1 Consideraciones generales

Sobre la base de los estándares revisados en el capítulo 2 y otros procedimientos comprendidos en el marco del Mantenimiento de Software y teniendo en cuenta las características de los entornos de las pequeñas organizaciones y sus aplicaciones de gestión, se realizará la propuesta para ejecutar un proceso de mantenimiento a sistemas que cumplan con estas características.

Desde este punto de vista, el proceso de mantenimiento puede verse como el conjunto de todas las operaciones que es necesario realizar sobre el software para implementar las modificaciones solicitadas o identificadas en el sistema que se somete a mantenimiento en cuestión. En ambos casos la propuesta de procedimiento se refiere a dichas modificaciones como *elemento de modificación*.

Para dotar a este conjunto de operaciones de una base metodológica, es preciso definir con anterioridad el propio proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se lleve a cabo, sea conforme a un proceso de mantenimiento predefinido (15).

Es por ello que este procedimiento tiene como objetivo conducir el Mantenimiento de Software de gestión en pequeñas organizaciones, de manera organizada y disciplinada, logrando una buena gestión y optimización de las modificaciones según su criterio de clasificación y módulos priorizados, lo que permitirá un mejor manejo la complejidad inherente al proceso de Mantenimiento de Software.

3.2 Alcance

La propuesta va dirigida a pequeñas organizaciones que tengan procesos automatizados y que necesiten que las mejoras propias de la evolución de dichos procesos en la

organización se reflejen los sistemas de gestión que los apalancan. Es aplicable a los proyectos de Mantenimiento de Software, en estas instituciones, a realizarse en aplicaciones de gestión, basándose en las características propias de este perfil que se referenciaron el epígrafe 2.4 para el caso de los sistemas de gestión.

3.3 Roles

La definición de una estructura para la organización del Mantenimiento de Software debe ser flexible y adecuarse a las características de la organización. Cuando para enfrentar este trabajo se cuenta con pocos miembros existe la necesidad de una mayor organización para minimizar los esfuerzos y brindar un mayor resultado. A continuación se detallan los roles participantes como miembros de la actividad de mantenimiento:

Directivo: Representa a persona con autoridad en la organización y de inmediato interés en que se le realice el servicio de mantenimiento al sistema de gestión en cuestión para mejorar un proceso de negocio que tiene lugar en su entorno.

Usuario: Representa a la persona conocedor del sistema, que lo opera regularmente en el control de la información del proceso que automatiza la aplicación de gestión. Aporta gran valor a la hora de identificar las modificaciones necesarias a realizar según la importancia de las mismas y constituye un ente fundamental en el proceso de prueba a realizar sobre cada una de las modificaciones implementadas. Este rol es clave y constante dentro del equipo de mantenimiento de manera que se encuentre totalmente involucrado en esta actividad ya que se necesita aprovechar la experiencia acumulada por él, siendo quien domina el sistema permitiéndole al resto de los miembros del grupo de mantenimiento, aprovechar sus conocimientos en áreas determinadas del software, incluso a nivel de funcionamiento interno, de manera que se contribuya a minimizar los esfuerzos extra que constituye en sí la realización de esta actividad por parte del personal. Incluir al usuario constituye una fortaleza para poder obtener un feedback rápido del desempeño de la actividad en la entrega rápida y periódica de atención a las peticiones de mantenimiento.

Mantenedor: Realiza la modificación del software por lo que identifica los elementos de modificación con sus correspondientes atributos de conjunto con el usuario. Planifica y diseña cuidadosamente las modificaciones que posteriormente va a implementar. Realiza determinadas pruebas a las modificaciones y está a cargo de la migración paulatina y retirada del sistema.

El *equipo de mantenimiento* en general denota de al personal que realiza dicha actividad y que entra dentro de la clasificación de simple según (6) o sea integrado por no más de 2 ó 3 miembros que conforman los roles de usuario y mantenedor.

3.4 Tipos de mantenimiento

La categoría de mantenimiento que comprende el procedimiento que se propone es del tipo Planificable asociado a los tipos de mantenimientos revisados en el capítulo 1 y de acuerdo a los que establece la norma 12207 revisada en el capítulo 2 por la cual se rige la propuesta. Corresponden estos a: Correctivo No Urgente (CNU), Perfectivo (PER), Adaptativo (A), Preventivo (PRE). A continuación se detalla la situación que comprende cada uno de ellos:

CNU: Se produce cuando existe un error en el producto software que no es crítico, pero que tal vez impida el funcionamiento de la aplicación o el normal funcionamiento de la empresa en un período de tiempo relativamente corto.

PER: Se ocupa de añadir al software en explotación nuevas características o funcionalidades, habitualmente solicitadas por el usuario.

A: Se aplica cuando el software en explotación va a cambiarse para que continúe funcionando correctamente en un entorno cambiante.

PRE: Es aplicado cuando se desea mejorar las características internas de un producto software buscando que en un futuro el esfuerzo de mantenimiento sea menor.

Estas clasificaciones comprenden los posibles tipos de mantenimientos a realizar en el procedimiento que se propone y serán utilizados a la hora de identificar el tipo de mantenimiento a realizar según la modificación necesaria a realizar.

3.5 Estructura general del procedimiento para el Mantenimiento de Software

El procedimiento que se propone consta de 5 etapas claves que cumplen con las indicaciones del estándar ISO/IEC 12207 y además le permiten al equipo de Mantenimiento de Software, utilizar un procedimiento abreviado para realizar esta actividad. Cada una de estas actividades encierra a su vez varias tareas que definen y comprenden la secuencia que describe la actividad. Las actividades pueden ser agrupadas en iniciales, intermedias y finales. Las tareas comprendidas entre las actividades intermedias y parte de las actividades finales se irán realizando a ciclo completo por los módulos según los elementos de modificaciones definidos. Se propone además una estructura de documentación complementaria como herramientas a utilizar que permitirá recoger una traza del proceso realizado generando así la documentación esencial como constancia del mantenimiento realizado. Esta documentación tributa al seguimiento de una adecuada Configuración del Software (documentación e información sobre los requerimientos, especificación, diseño y pruebas) lo que reduce la cantidad de esfuerzo requerido en el mantenimiento y mejora la calidad general de los cambios.

A continuación las actividades del procedimiento para realizar el proceso de mantenimiento en sistemas de gestión:

1. Solicitud del mantenimiento
 - a. Manifestar necesidad de mantenimiento de un software determinado
2. Análisis del sistema
 - a. Preparar entorno de prueba operativo
 - b. Comprensión y Revisión detallada del software
 - c. Identificación de los elementos de modificación
 - i. Tipo: Recogidas por el usuario, Detectadas en la aplicación, Mejoras propuesta
 - d. Clasificar y priorizar los elementos de modificación
 - i. Tipo: correctivo no urgente, preventivo, perfectivo o adaptativo a un nuevo entorno
 - ii. Alcance: Referente al tamaño de la modificación
 - iii. Criticidad: Referente a si es un elemento clave dentro de la modificación del que dependen otros elementos del software
 - e. Realizar análisis de impacto, planificación y estimación de la duración de la actividad de mantenimiento
 - i. Calcular valor de mantenibilidad
3. Diseño de las modificaciones
 - a. Modificación a realizar en cada uno de los elementos de cambio
4. Ejecución de las modificaciones
 - a. Implementación de la modificación
 - b. Realización de pruebas a la modificación
5. Migración y retirada
 - a. Migración paulatina o Actualización de la modificación realizada
 - b. Retirada total

Como se había comentado anteriormente las etapas claves antes mencionadas pueden ser agrupadas en actividades Iniciales que comprenden las etapas de *Comprensión del software* y *Análisis del sistema*, en actividades Intermedias que comprenden las etapas de *Diseño y Ejecución de las modificaciones* y las actividades Finales asociadas a la etapa de *Migración y Retirada* dentro del proceso de mantenimiento.

Las actividades iniciales son realizadas como parte de una valoración y estimación inicial por parte del equipo de mantenimiento. Posteriormente las actividades Intermedias se realizaran según las modificaciones identificadas por cada uno de los módulos que componen el sistema de gestión. De haber algún problema en algunas de las tareas definidas en la ejecución de la modificación pues se corregirá nuevamente el diseño de la

modificación. De esta manera se realizarán las actividades agrupadas como intermedias y la actividad de migración paulatina de las actividades finales, volviendo nuevamente a las actividades intermedias pero del próximo módulo.

A continuación se muestra una la estructura del proceso de mantenimiento propuesto según las actividades generales a desarrollar.

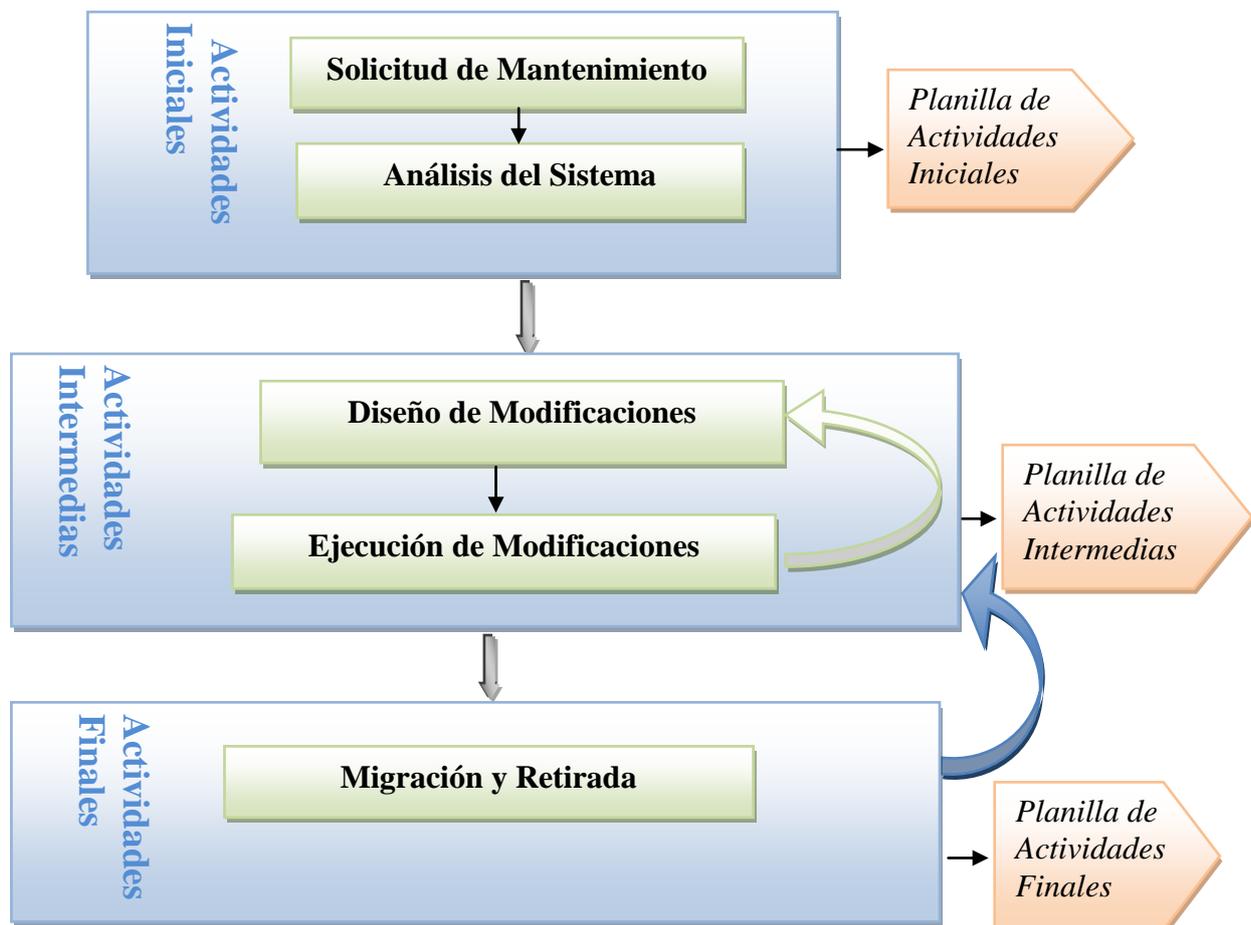


Fig. 14 Propuesta gráfica general del procedimiento para la realización de la actividad de Mantenimiento de Software en aplicaciones de Gestión.

3.6 Estructura detallada del Procedimiento de Mantenimiento

A continuación se presenta especificado el procedimiento de mantenimiento propuesto en el epígrafe anterior dando su estructura detallada de en qué consiste cada una de las etapas definidas y asociándolo con la herramienta propuesta según la actividad a realizar para documentar el proceso de mantenimiento en cuestión. Siguiendo la idea de ISO/IEC 12207, cada bloque mostrado en la Figura 14 está formado por un conjunto de actividades, estando cada actividad compuesta de un conjunto de tareas.

Los artefactos que las actividades producen como salida son documentos en forma de planillas, que en el texto se hará referencia a estos documentos que se encuentran recogidos en los Anexos.

Actividades Iniciales:

✓ *Solicitud del mantenimiento*

Rol: Usuario, Directivo

Entrada: Inconformidad en la operatividad de un sistema.

Salida: Identificación del Sistema a Mantener, Equipo de Mantenimiento

Soporte: Planilla de Actividades Iniciales

Descripción:

La organización que sustenta procesos internos sobre la base de sistemas automatizados paulatinamente requiere lograr que el mantenimiento de sus productos software se realice de manera disciplinada mediante el cumplimiento y elaboración sistemática de los productos de trabajo propuestos. Para ello se necesita mantener una corta distancia entre el funcionamiento del proceso real y el proceso automatizado, subsanar errores que afectan el adecuado funcionamiento de la aplicación, así como introducir mejoras necesarias para mantener la competitividad y la utilidad de la herramienta que viabiliza el mejor desempeño de una determinada actividad. Cuando se expresa alguno de estos elementos determinantes, y si se cuenta con los recursos necesarios para emplearlos en esta actividad de Mantenimiento de Software, pues se desencadena un proyecto a lo interno de la organización enmarcado en esta fase.

Para realizar esta actividad se debe realizar la siguiente tarea:

○ Petición de mantenimiento de un software determinado

Descripción:

El proceso de mantenimiento comienza cuando se expresa la voluntad y necesidad por parte del directivo de la organización de actualizar sus sistemas en función de los cambios propios asociados a los procesos en que estos se enmarcan, para corregir fallas de funcionamiento del software o para mejorarles sus características desde el punto de vista de elementos técnicos del sistema.

En tal caso se debe expresarse la solicitud correspondiente utilizando la sección de solicitud de la planilla de Actividades Iniciales del anexo 6 que se propone donde se especificarán datos generales referente al sistema que se le aplicará el mantenimiento, la organización a la que pertenece dicho sistema, la fecha de la solicitud y las razones de la misma. En esta actividad quedan designados las personas involucradas en la actividad de mantenimiento siendo generalmente el usuario inmediato de la aplicación que se somete a mantenimiento en el rol de

Usuario y uno o dos especialistas de TI de la organización en el rol de Mantenedor quedando conformado de esa manera el equipo de mantenimiento.

✓ *Análisis del sistema*

Rol: Mantenedor, Usuario

Entradas: Componentes del Software Original, Documentación existente del sistema.

Salidas: Resumen Técnico del sistema, Lista de Elementos de Modificación, Planificación de la actividad de mantenimiento.

Soporte: Planilla de Actividades Iniciales

Descripción:

En esta actividad se debe lograr mediante un estudio y repaso detallado de los módulos que componen la identificación y distinción de los elementos de modificación necesarios según su tipo de mantenimiento para una buena gestión y optimización de la atención a los elementos de modificación, ofreciendo un criterio para clasificarlas y priorizarlas, así como para realizar una estimación del tiempo necesario para llevarlos a cabo y del resultado de su impacto en la nueva versión de la aplicación.

Para realizar esta actividad se deben realizar las siguientes tareas:

○ Preparar entorno de prueba operativo

Descripción:

Para comprobar el sistema y comenzar a realizar los trabajos propios de mantenimiento que a continuación se detallarán se propone que el mantenedor prepare el entorno de prueba operativo que consiste en a partir de los componentes originales del sistema, realizar copias del software, preparar las base de datos y archivos (el entorno), que sean semejantes a la realidad y que cubran la totalidad de las funcionalidades del sistema. El objetivo es que el software pueda funcionar en un ambiente aislado, para que no afecte la operación normal del sistema original en uso.

○ Comprensión y revisión detallada del software

Descripción:

Esta actividad presupone la comprensión del software y de los cambios a realizar por parte del equipo de mantenimiento por lo que desde el enfoque funcional de la aplicación en esta actividad el usuario final del sistema como miembro activo del equipo juega un papel fundamental en esta etapa. Para poder modificar un programa, los programadores necesitan conocer su funcionalidad y objetivos, su estructura interna y los requisitos de operación. Se tomará en cuenta, de existir, la documentación, manuales de usuario o

cualquier otra especificación existente referente al sistema que se le someterá a mantenimiento. En esta actividad de revisión y asimilación del código se debe realizar una evaluación general a manera de Resumen Técnico, centrándose y profundizando en tecnología de desarrollo, arquitectura, cada uno de los módulos que componen la aplicación, desde el correcto funcionamiento de las operaciones que lo componen hasta el código que lo soporta el cual en ocasiones resulta es poco legible, es por ello que se plantea dicha tarea como necesaria. Como parte de la dinámica de la actividad se puede utilizar técnicas de apoyo a la actividad como pueden ser la realización de entrevista con los usuarios, observar cómo trabaja éste, con el fin de obtener el conocimiento necesario y suficiente del producto software a mantener. Se propone para documentar esta actividad utilizar la sección de Análisis General del Sistema de la planilla de Actividades Iniciales del anexo 6.

- Identificación de los elementos de modificación

Descripción:

De la revisión detallada realizada anteriormente en el sistema recorriendo cada uno de sus módulos se construye una lista de los Elementos Software a Corregir (módulos, rutinas, documentos, entre otros) que se convierten a los efectos del equipo de mantenimiento, en elementos de modificación los cuales según su origen pueden ser solicitudes explícitas realizadas por el usuario, fallas detectadas en la revisión de la aplicación o mejoras que se proponen desde el punto de vista de mejorar el funcionamiento u otro atributo del software. Estos elementos de modificación son agrupados según el módulo al que se identificó y por ende pertenece. También se procede a analizar y reproducir, de ser posible, las causas que verifiquen el origen de la petición de mantenimiento por parte del Mantenedor y con la colaboración del Usuario. Además estudia diferentes alternativas para implementar la modificación en caso de la corrección de un error. Esta actividad se enmarca en la gestión de problemas y petición de modificaciones e implementar el proceso de gestión de configuración para gestionar las modificaciones al sistema existente.

Para documentar esta actividad se propone la sección de módulos en la planilla de Actividades Iniciales del anexo 6, donde se especificarán por módulos los elementos de modificación.

- Clasificar y priorizar los elementos de modificación

Descripción:

Una vez identificados los elementos de modificación, estos pueden ser clasificados acorde a las diferentes tipos de mantenimiento expuestos en el

capítulo 1, epígrafe 1.5, en correctivo no urgente, preventivo, perfectivo o adaptativo a un nuevo entorno. Se definirá el alcance de la modificación referido a hasta donde comprenderá la modificación y el tamaño de la misma, así como la criticidad de las modificaciones en función de la prioridad e importancia para el sistema que permitirá establecer un orden prioritario en la realización de las mismas en alta, media y baja. Para documentar esta actividad se propone la sección de módulos en la planilla de Actividades Iniciales del anexo 6, como atributos de los elementos de modificación.

- Realizar análisis de impacto, planificación y estimación de la duración de la actividad de mantenimiento

Descripción:

Esta actividad también incluye un análisis de la petición de modificación con relación a su impacto en la organización, el sistema existente y los sistemas con los que interacciona y se procederá a la Planificación de la actividad de mantenimiento. Se puede realizar el cálculo del índice de mantenibilidad de ser necesario, para considerarlo como elemento de mejora durante la actividad de mantenimiento, así como para la planificación de la actividad de mantenimiento. Se propone para realizar dicho cálculo la métrica de Índice de Mantenibilidad revisada en el capítulo anterior. En esta actividad tiene un papel activo el mantenedor en cuestión como protagonista en cada una de las tareas a realizar siempre apoyándose en el conocimiento y la experiencia del usuario. Para la planificación y estimación del tiempo de duración de la actividad de mantenimiento se puede auxiliar de métodos definidos para ello como el diagrama de Gantt. Para documentar esta actividad se propone la sección análisis de impacto y planificación en la planilla de Actividades Iniciales del anexo 6.

Actividades Intermedias:

- ✓ *Diseño de las modificaciones*

Rol: Mantenedor

Entradas: Lista de Elementos de Modificación

Salidas: Diseño de solución propuesta para implementar en los Elementos de Modificación

Soporte: Planilla de Actividades Intermedias

Descripción:

En esta actividad documenta y diseña las modificaciones a realizar en los elementos de modificación identificados en las actividades iniciales.

Para realizar esta actividad se debe realiza la siguiente tarea:

- Modificación a realizar en cada uno de los elementos de cambio

Descripción:

De los elementos de modificación identificados en la actividad de análisis, se refiere a especificar las acciones de modificación por módulos a realizar por cada uno de los elementos de modificación encontrados previamente en la actividad de análisis. Se debe identificar todos los componentes del producto software (rutinas, bases de datos, etc.) afectadas por la intervención. En caso de ser necesario se puede referenciar y combinar artefactos, procedimientos y metodologías necesarias para implementar dichas modificaciones. La planilla de actividades intermedias recoge los datos propios de esta actividad como se muestra en el anexo 7.

El responsable de llevar a cabo dicha actividad es el mantenedor siempre corroborando en caso de que se requiera con el usuario que valida fundamentalmente las propuestas de mejoras a realizar desde la óptica funcional del sistema.

✓ *Ejecución*

Rol: Mantenedor

Entradas: Diseño de solución propuesta para implementar en los Elementos de Modificación,

Salidas: Elementos de Modificación corregidos, Documentación de las pruebas realizadas

Descripción:

Para incorporar los cambios necesarios se deben crear y modificar las estructuras de datos, la lógica de los procesos, las interfaces y la documentación pertinente. El equipo de mantenimiento debe conocer lo mejor posible las repercusiones que tienen en el sistema los cambios que están realizando, con el fin de evitar al máximo posible los efectos secundarios.

- Implementación de la modificación

Descripción:

En esta actividad se procederá a realizar la implementación de las modificaciones identificadas en la etapa de análisis y posteriormente concebidas en la actividad de diseño, haciendo efectivo las acciones de mantenimiento. Se debe tener en cuenta la clasificación de criticidad realizada en la etapa de diseño identificando primeramente los módulos significativos que sirven de base al resto para dar comienzo a la implementación del mantenimiento. La misma debe realizarse teniendo en cuenta todos los elementos de modificación perteneciente a un módulo, de manera que no se proceda con un próximo módulo sin antes haber cerrado con el módulo anterior, siempre y cuando sea posible. Una vez realizada dicha implementación, se llevará a cabo la realización de las pruebas para garantizar el

correcto funcionamiento de los mismos, siempre y cuando lo requiera la modificación.

- Realización de pruebas a la modificación

Descripción:

El Equipo de mantenimiento realiza las pruebas unitarias y de integración sobre el producto software intervenido. Se debe comprobar que el elemento de modificación fue atendido de acuerdo al diseño realizado y de que los diferentes elementos software funcionan correctamente en forma conjunta. Las pruebas pueden ser implementadas usando métodos y criterios de prueba y evaluación conocidos para probar y evaluar las partes modificadas y no modificadas del sistema. El propósito es verificar el correcto funcionamiento del software tanto en módulos independientes como en todo el sistema. El mantenedor deberá llevar a cabo revisiones con el usuario del sistema que integra el equipo de mantenimiento. Una vez concluida las pruebas, se debe obtener la aprobación de terminación satisfactoria de la modificación. En caso de algún fallo de prueba, inconformidad por parte del usuario final de la aplicación o alguna implicación en la modificación realizada se debe nuevamente revisar la propuesta de solución a la modificación y además la implementación de la misma, y el resto de las tareas secuenciales consiguiente.

En esta actividad participan activamente tanto el mantenedor como el usuario y de ser necesario el directivo de la organización para verificar alguna modificación de especial interés, como pudieran ser los informes resultantes los cuales son objeto de análisis en la organización.

Los artefactos adicionales utilizados se especifican y anexan a las planillas del procedimiento para documentar todo el proceso de mantenimiento en la sección de comentarios de la planilla de actividades finales del anexo 8.

Actividades Finales:

- ✓ *Migración y retirada*

Rol: Mantenedor

Entradas: Aprobación de fin de mantenimiento de los módulos.

Salidas: Planificación de la migración, Planificación de la retirada.

Soporte: Planilla de Actividades Finales

Descripción:

En esta actividad se hacen efectivos en el sistema original los cambios que se han estado realizando durante el proceso de mantenimiento. En la etapa de migración debe mantener una alternativa del sistema o producto software (incluyendo los datos) de un entorno de

operación de la versión inicial intacto, al menos hasta que no concluya satisfactoriamente la migración y retirada total del sistema en cuestión que se somete a mantenimiento lográndose la migración y retirada total exitosa del software dando paso a una herramienta renovada. Para recoger brevemente una síntesis asociada a la retirada y migración del software se propone el uso de la planilla de actividades finales donde se recogen datos propios de esta actividad como se muestra en el anexo 8.

- Actualización de la modificación realizada

Descripción:

Esta tarea se ejecuta una vez obtenido un resultado satisfactorio de las pruebas realizadas a los elementos de modificación del módulo en mantenimiento. Los ficheros modificados de dicho módulo en el entorno operativo de prueba son reemplazados por los ficheros equivalentes en el entorno real de trabajo con los cambios realizados en el proceso de modificación. Se produce así una migración paulatina del sistema según se vaya cerrando el ciclo de mantenimiento por módulos de la aplicación, de forma tal que el usuario podrá contar con su herramienta de trabajo en transición y de manera activa para su explotación reflejándose paso a paso las correcciones de las modificaciones necesarias solicitadas e identificadas. Se debe verificar la interacción de dicha actualización de la capa de aplicación con la capa de los datos con la cual interactúa. La ejecución de la migración de cada uno de los módulos que componen la aplicación queda recogida en la sección de Migración de la planilla de actividades finales del anexo 8.

- Retirada total

Descripción:

Esta actividad es la declaración oficial de la finalización de la fase de mantenimiento de la que ha sido objeto el sistema, obteniéndose así una nueva versión del mismo que queda en funcionamiento para explotación. Dentro de las actividades asociadas a la migración se debe incluir, de ser necesario, una breve capacitación donde se adiestre al usuario del sistema en las nuevas opciones incorporadas a la aplicación. Se debe mantener operativa la versión anterior durante un tiempo establecido por el equipo de mantenimiento. En esta actividad tiene un fuerte componente el mantenedor enfocado en aspectos técnicos y el usuario del sistema como parte de la asimilación de la migración paulatina y final del sistema. La ejecución de esta tarea queda recogida en la sección de Retiro de la planilla de actividades finales del anexo 8.

Con esta última actividad se da por finalizado el ejercicio de mantenimiento al sistema en cuestión de manera formal. Se adiciona a la documentación generada de las planillas de

mantenimiento a la existente del sistema que se le culminó el ciclo de mantenimiento dejando así constancia de la evolución obtenida en su nueva versión.

El procedimiento puede aplicarse iterativamente durante cada fase de mantenimiento necesaria según la figura 1 de acuerdo al proceso evolutivo del software.

Especificando en forma de procesos las entradas y salidas correspondientes por cada una de las actividades identificadas el procedimiento se tiene:

Rol	Entrada	Control	Actividad	Salida
<ul style="list-style-type: none"> ▪ Directivo ▪ Usuario 	<ul style="list-style-type: none"> ▪ Inconformidad en la operatividad de un sistema. 	Planilla de actividades iniciales	Manifiestar necesidad de mantenimiento	<ul style="list-style-type: none"> ▪ Identificación del Sistema a someter a mantenimiento ▪ Equipo de mantenimiento conformado
<ul style="list-style-type: none"> ▪ Mantenedor ▪ Usuario 	<ul style="list-style-type: none"> ▪ Componentes del Software Original ▪ Documentación existente del sistema 	Planilla de actividades iniciales	Análisis del sistema Preparar entorno de prueba operativo	<ul style="list-style-type: none"> ▪ Resumen Técnico del sistema. ▪ Lista de Elementos de Modificación. ▪ Planificación de la actividad de mantenimiento
<ul style="list-style-type: none"> ▪ Mantenedor 	<ul style="list-style-type: none"> ▪ Lista de Elementos de Modificación 	Planilla de actividades Intermedia	Diseño de las modificaciones	<ul style="list-style-type: none"> ▪ Diseño de solución propuesta para implementar en los Elementos de Modificación
<ul style="list-style-type: none"> ▪ Mantenedor ▪ Usuario 	<ul style="list-style-type: none"> ▪ Diseño de solución propuesta para implementar en los Elementos de Modificación. ▪ Ficheros de la aplicación con los nuevos cambios realizados 	Planilla de actividades Intermedia	Ejecución	<ul style="list-style-type: none"> ▪ Elementos de Modificación corregidos. ▪ Documentación de las pruebas realizadas.
<ul style="list-style-type: none"> ▪ Mantenedor 	<ul style="list-style-type: none"> ▪ Aprobación de fin de mantenimiento de los módulos 	Planilla de actividades Finales	Migración y retirada	<ul style="list-style-type: none"> ▪ Planificación de la migración de módulos concluidos. ▪ Planificación de la migración, Planificación de la retirada.

A continuación la se modela la propuesta detallada descrita anteriormente en función de las actividades generales y de sus actividades internas.

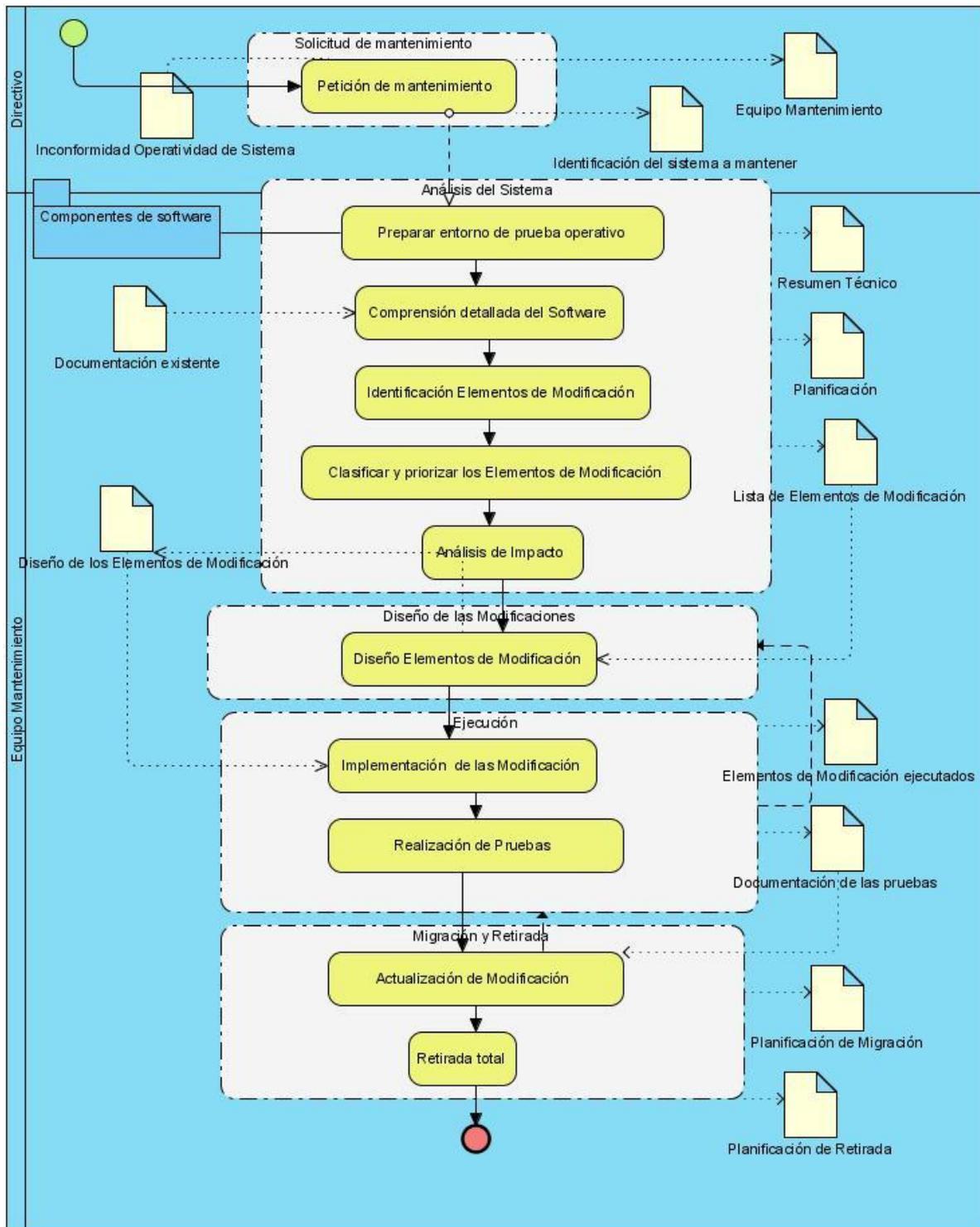


Fig. 15 Propuesta gráfica del procedimiento para la realización de la actividad de Mantenimiento de Software en aplicaciones Gestión.

Conclusiones Parciales

En este capítulo se realizó la propuesta del procedimiento que permite guiar el proceso de mantenimiento basado en un conjunto de 3 bloques de actividades agrupadas en iniciales, intermedias y finales. Estas son sustentadas por herramientas en forma de planillas que recogen la traza de información en cada una de las modificaciones necesarias a realizar y que permiten llevar un seguimiento de la evolución de un sistema de software. El mismo se encuentra enmarcado en el proceso general que define para ello, las indicaciones del estándar ISO/IEC 12207.

Se especifica en detalle las actividades del mismo, así como sus entradas y salidas correspondientes mediante un diagrama para reflejarlo gráficamente, lo que facilita su implantación, comprensión y ejecución de manera gradual e integrada.

Los aportes fundamentales del procedimiento radican en la incorporación del usuario de la aplicación como miembro directo del equipo de mantenimiento, la inclusión de la actividad de migración dentro del ciclo cerrado e iterativo a realizarse dentro del proceso de mantenimiento y la simplificación en cuanto a normativas y documentación a realizarse para llevar esta actividad en pequeñas organizaciones resumidas y expresadas en el soporte de planillas propuestas por el procedimiento.

Capítulo 4 Aplicación del Procedimiento Propuesto

Introducción

En este capítulo final de la investigación se realizará la aplicación del procedimiento propuesto en el capítulo anterior. Dicha ejecución se basará en el software GSP, un sistema para la Gestión de la Superación Pedagógica del CREA del cual inicialmente se expondrán las características generales del mismo. Se evaluará la calidad del sistema GSP mediante el cálculo del Índice de Mantenibilidad descrito en el capítulo anterior. Dicha medición se realizará antes y después de aplicado el procedimiento propuesto lo cual permitirá evaluar los resultados del mantenimiento realizado. Además se expondrán los resultados obtenidos luego de haber realizado la ejecución de la propuesta del procedimiento de mantenimiento en el software GSP del CREA y del propio procedimiento en general.

4.1 Descripción del Sistema GSP: Gestión de la Superación Pedagógica

El software GSP: Gestión para la superación pedagógica, es la aplicación que se utiliza en el área de superación pedagógica y eventos del CREA para llevar un control de las actividades de postgrado que en el centro tienen lugar. Para interactuar con la misma se cuenta con varios roles. El rol de jefe de profesor, que le permite a cada profesor registrar en la aplicación información pertinente de cada una de las actividades de postgrado que realizó, como cursos, y las llamadas otras actividades, que incluyen talleres, mesas redondas, conferencias, entre otras actividades. También se encuentra el rol de administrador que gestiona los distintos usuarios que pueden registrarse en la aplicación así como el rol que los mismos desempeñan. El jefe de área es otro rol con que cuenta la aplicación que es mayormente quien lleva toda la carga operativa y funcional del sistema. Este además de ver las opciones propias del profesor también realiza la gestión de los entrenamientos que en el centro tienen lugar. De cada entrenamiento a una persona en particular se le lleva una constancia de las actividades programadas para el mismo así como de las evaluaciones en cada una de ellas hasta culminar con criterio de evaluación del entrenamiento. Este rol es el encargado de gestionar la información relacionada con las maestrías que imparte el centro. El software le permite llevar registro de las distintas ediciones de las maestrías, de las asignaturas asociadas a la misma, los maestrantes y sus notas, así las evaluaciones en los actos de predefensa y defensa de la maestría. El sistema también cubre lo referente a la gestión de doctorado en Ciencias Pedagógicas que en el centro tiene lugar. De dicha actividad se registra lo concerniente a los aspirantes al grado de doctor en ciencias pedagógicas, sus datos personales, sus planes de trabajo, participaciones en eventos, publicaciones, las evaluaciones en sus exámenes de mínimo, así como las evaluaciones de cierre de predefensa y defensa de su tesis de doctorado. Además de la gestión propia de cada una de estas actividades el rol jefe de área también

obtiene informes dinámicos pertinentes a cada una de estas actividades de postgrado así como la gestión primaria de los profesores y los países en los que se desarrollan las actividades.

Otro rol asociado a la aplicación es el de la secretaria, la cual puede registrar las evaluaciones de los maestrantes en las asignaturas de la maestría.

A continuación se muestra la relación entre los módulos que componen el sistema y los usuarios que interactúan con estos.

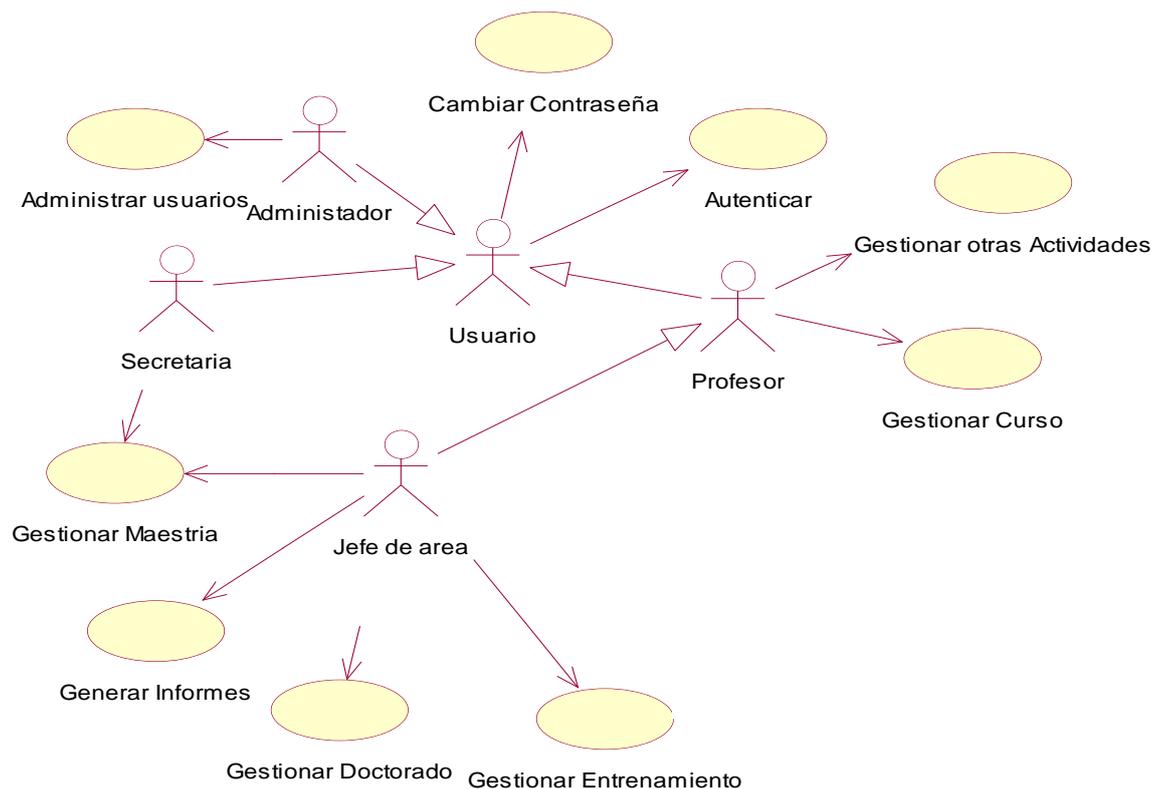


Fig. 16 Diagrama de Casos de Uso del Sistema GSP

La aplicación posee mecanismos de seguridad que garantizan la privacidad asociada a cada uno de los roles anteriormente descritos. Inicialmente mediante la autenticación requerida para acceder a la aplicación en sí. La aplicación permite cambiar la contraseña en cualquier momento, así como cerrar sesión garantizando un adecuado trabajo con la sesión propia de cada uno de los usuarios.

Dicha aplicación fue desarrollada usando la tecnología Web, utilizándose como lenguaje de programación del lado del servidor PHP, como gestor de bases de datos MySQL y servidor web Apache. Como tecnologías complementarias se usó javascript, CSS, DOM formando en su conjunto el DHTML. El ciclo de vida del software fue desarrollado por una sola persona, encargada de realizar cada una de las fases, levantamiento de requisitos,

análisis, diseño, implementación y prueba, obteniendo finalmente un resultado satisfactorio y desarrollando dicho sistema a la medida propio para centros que gestionen actividades de superación posgraduada.

4.2 Evaluación de la calidad de software antes de la ejecución de la propuesta

Usando la técnica del índice de mantenibilidad propuesta en el capítulo 2 epígrafe 2.5 para evaluar la calidad del software se realizará aplicación de la misma en el módulo gestión de profesores de la aplicación GSP seleccionado anteriormente.

Sobre la base del polinomio del IM:

$$IM = 171 - 5.2 * \ln(\text{aveV}) - 0.23 * \text{aveV} (g') - 16.2 * \ln(\text{aveLOC}) + 50 * \sin(\sqrt{2.4 * \text{perCM}})$$

aveV: es la media del volumen por módulo según Halstead.

A continuación se realiza el cálculo del volumen del módulo de profesores según la propuesta de Halstead y la complejidad ciclométrica por cada uno de los ficheros que lo componen el módulo de profesores:

Fichero	n1	N1	n2	N2	g'
Agregar_profesor.php	11	23	8	19	2
Eliminar_profesor.php	14	41	5	18	1
Listado_profesores.php	38	88	13	27	16
Modificación_profesor.php	37	124	11	24	18
Modificar_profesor.php	11	34	10	19	1
Nuevo_profesor.php	35	118	12	26	4
Ver_profesor.php	31	115	14	27	9
Validacion_profesor.php	27	69	12	28	7
Imprimir_profesor.php	17	93	9	21	9

$$n = n1+n2$$

$$N = N1+N2$$

$$n = 221+94$$

$$N = 705+209$$

$$n = 315$$

$$N = 918$$

$$V = N * \log_2 n$$

V=2295

aveV=2295

aveV(g'): es la media de la complejidad ciclomática por módulo, que mide el número de caminos linealmente independientes a través de un módulo de programa. ($m = \text{Número de condiciones} + 1$)

aveV(g')= 68

aveLOC: es la media del número de líneas de código por módulo.

aveLOC=841

perCM: es la media porcentual de líneas de código comentadas.

perCM=58

$IM = 171 - 5.2 * \ln(\text{aveV}) - 0.23 * \text{aveV}(g') - 16.2 * \ln(\text{aveLOC}) - 50 * \sin(\sqrt{2.4 * \text{perCM}})$

$IM_A = 171 - 5.2 * \ln(2295) - 0.23 * 68 - 16.2 * \ln(841) - 50 * \sin(\sqrt{2.4 * 58})$

IM_A=46.6

El cálculo del IM realizado arroja que un valor considerablemente pequeño lo que indica que el sistema actual resulta hostil para la realización de su mantenimiento como fase evolutiva del mismo.

Estos resultados se deben mayormente porque al realizar un análisis del código del sistema actual, posee un código altamente estructurado, entremezclando diferentes tipos de lenguajes que combina la web como es el HTML, javascript, php el uso de librerías para el acceso a dato poniendo su nomenclatura propia, css para los estilos y todo eso sin un orden formal ni una estructura de reutilización donde no se hace uso de programación orientada a Objetos. Lo anterior es lo que se conoce en la literatura como código espaguetis y que influye en el alto grado de hostilidad que presenta el código para su entendimiento. La combinación de estos factores, hacen que el sistema GSP sea altamente complejo y prácticamente inentendible para especialistas mantenedores que no hayan participado en su desarrollo y tengan que enfrentar su análisis para poder realizar la actividad de mantenimiento.

Una de las recomendaciones que generalmente se realizan como elemento de éxito en el Mantenimiento de Software es que de ser posible los mismos desarrolladores del sistema pues estén a cargo de su mantenimiento ya que poseen el conocimiento y la experiencia necesaria para llevar a cabo dicha actividad. En el caso del mantenimiento del sistema GSP a pesar del bajo IM que posee, cuenta dentro del equipo de mantenimiento del sistema a la desarrolladora del mismo, por lo que esta baja codificación de la mantenibilidad del sistema se verá relativamente compensada con la participación dentro

del equipo de mantenimiento de la desarrolladora del mismo, lo propicia que a pesar de esto dicha actividad de mantenimiento pueda ser realizada satisfactoriamente.

4.3 Ejecución de la propuesta de mantenimiento

GSP, es un sistema de gestión de la información que se desarrolló en el área de Superación Pedagógica del CREA. El mismo fue desarrollado en el año 2004, con el levantamiento de requisitos y necesidades existentes en ese momento, según las distintas actividades de superación que en el área tenían lugar. El sistema se ha mantenido en explotación con aceptación desde ese entonces, brindando un conjunto de facilidades al encargado del área. Dentro de las mismas se puede mencionar el permitir un control más detallado y organizado sobre las actividades de postgrado que se realizan en el área de superación pedagógica por parte de los profesores que lo integran. La puesta en práctica del mismo permitió un cuantioso ahorro del tiempo para el profesor a cargo de dicha tarea, permitiéndole dedicar el mayor tiempo posible y encaminar los principales esfuerzos al cumplimiento de las metas trazadas en el área.

En el de cursar de la actividad diaria han surgido cambios propios de la evolución del proceso de control y registro de las actividades de superación pedagógica y postgrados que en el centro tienen lugar. Las necesidades de información tanto por parte de la dirección del centro como de la dirección de postgrado de la CUJAE, varían en función de los nuevos requerimientos que surgen para realizar la toma de decisiones. Lo anteriormente expuesto indica que actualmente el sistema se encuentra con un pequeño desfase con respecto a la forma precedente con que se desempeñaba el control y la gestión de la superación pedagógica en dicha área con la que tiene lugar actualmente.

Estas nuevas necesidades de registro y almacenamiento de información que han ido surgiendo con su uso y explotación continua de la aplicación, hacen que el sistema GSP necesite que se le implemente una mejora progresiva desde el punto de vista del desarrollo e implementación como producto de software. Los esfuerzos que realizan en el centro en la solución de los problemas de mantenimiento de dicho sistema son grandes y poco formalizados.

Para la aplicación del procedimiento que se propone se tomará de base el sistema GSP. El mismo cumple con las condiciones generales a tener en cuenta para el adecuado uso de este procedimiento. La aplicación y documentación para la puesta en práctica y ejecución del procedimiento propuesto se consolidará básicamente en el mantenimiento en función del módulo de Gestión de los Profesores como muestra de módulo básico para el adecuado funcionamiento posterior del resto de las actividades de postgrado. Por sus características generales de módulos de gestión en el mismo tienen lugar las operaciones CRUD de cada uno de las actividades de superación que se realizan es por ello que la complejidad operativa entre los mismos es de gran similitud.

La tabla que se muestra a continuación corresponde a parte de las actividades iniciales de la propuesta en correspondencia con el sistema GSP al cual se le aplicará el mantenimiento correspondiente.

Se hacen referencia los motivos de la solicitud, la fecha y los interesados en la realización de la actividad. También indican el sistema sobre el cual se le aplicara el mantenimiento.

Con la confección de esta parte inicial de la planilla queda realizada, recogida y documentada la primera actividad descrita en la propuesta de mantenimiento.

PLANILLA DE ACTIVIDADES INICIALES			
Solicitud de mantenimiento			
Entidad:	Centro de referencia para la Educación Avanzada (CREA)	Fecha Solicitud:	12 – 12-2008
Sistema:	GSP. Sistema para la gestión de la superación pedagógica en el CREA.		
Solicitante:	Dr. Elsa Herrero Dr. Tomas Cañas Louzau		
Petición:	El centro CREA se encuentra en un proceso de acreditación como parte del proceso que se viene desarrollando en la CUJAE. Para ello se necesita de tener un control detallado y organizado de cada uno de las áreas que conforman el centro. Particularmente en el Área de superación pedagógica, desde el 2004 se cuenta con el sistema GSP, desarrollado para el control de la actividad de postgrado en el centro constituyendo una gran utilidad en la gestión de la información relacionada con la superación posgraduada que en el centro tiene lugar. Actualmente el sistema se encuentra desactivado debido a problemas de configuración del mismo en los servidores del centro y además en el área se registra información que no contempla el sistema por lo que se hace necesario la reincorporación del mismo como herramienta de trabajo en el área para obtener resultados satisfactorios y garantizar el adecuado funcionamiento del centro en ese indicador. Constando así la solicitud de la necesidad de revisión y mantenimiento de dicho sistema por parte de la dirección del centro.		

Tabla #1 Sección inicial de la Planilla de actividades iniciales.

La sección que se muestra a continuación de la planilla de actividades iniciales en la tabla # 2 corresponde a la actividad de la propuesta referida al análisis general del sistema.

Análisis general del sistema
El sistema GSP es una aplicación de gestión sobre Web desarrollada utilizando el lenguaje de programación PHP, como servidor Web Apache 2.0 y como gestor de bases de datos MySQL. El mismo se encuentra estructurado en un conjunto de carpetas que se corresponden con cada uno de los módulos de gestión que integran el sistema los cuales son como básicos y necesarios para el funcionamiento del resto Módulo de Países y Módulo de Profesores. También se cuenta con Módulos para la gestión de la información propia de las actividades que se desarrollan tales como Cursos, Otras Actividades (Talleres, Mesas redondas, Seminarios), Entrenamientos, Maestrías, Doctorados y el Módulo de Informes que se refiere a la recuperación provechosa en función de cada una de estas actividades de posgrado en función de las necesidades de la dirección del centro y la universidad. Los Módulos de informes están integrados por 9 módulos correspondientes a la recuperación provechosa de la información según las diferentes actividades de superación.

Tabla #2 Sección de análisis general de la Planilla de actividades iniciales.

Esta descripción sobre las características generales del sistema GSP permite identificar la tecnología en la que fue desarrollado el sistema en su versión inicial, así como corroborar que es una aplicación soportada sobre la plataforma Web. A pesar que el código de la aplicación presenta un alto grado de estructuración, está muy bien delimitado cada uno de los módulos del software presentando clara identificación de los mismos, por los cuales se transitarán en ciclo del mantenimiento cerrando cada uno de ellos de acuerdo con lo que plantea la propuesta de mantenimiento.

En la tabla #3 se muestra la identificación de defectos o modificaciones asociado a uno de los módulos que conforman el sistema. En la tabla se abunda y se refiere solamente al módulo de profesores de la aplicación GSP, debida a la extensión de la documentación del análisis por todos los módulos.

Profesores		Módulo de gestión de los profesores vinculados al centro. Comprende el listado de los mismos, el registro de nuevos profesores, la modificación de la información, eliminar profesores que dejan de cumplir su función si no tienen ninguna actividad asociada y el mostrar los datos asociados a un profesor en particular así como un informe imprimible del mismo.				
# ITEM	Elemento de Modificación	Origen	Priorización			Descripción
			Tipo	Alcance	Criticidad	
GSP5	Programación del Código del módulo de profesor a OO	Mejora Propuesta	Preventivo	Definir en una clase todas las responsabilidades asociadas a la gestión del profesor.	Alta	El código del módulo se encuentra de manera estructurado con todo el código entremezclado con el html, el php el código asociado de la librería de BD ADOdb y dificulta considerablemente la legibilidad y la reutilización del modulo en cuestión.
GSP6	Implementación de la opción buscar Profesor	Recogidas por el usuario	Perfectivo	Permitir la búsqueda en la página de	Media	El número de profesores crece a medida que se van desarrollando las

				listado de profesores que cumplan con un determinado criterio de entrada.		actividades de superación y se necesita viabilizar la búsqueda de los mismos en la lista tan exhaustiva de estos.
GSP7	Implementación del paginado de profesores	Recogidas por el usuario	Perfectivo	Permitir mostrar una cuota menor en el listado de profesores ofreciendo páginas según la cantidad total.	Baja	El número de profesores crece a medida que se van desarrollando las actividades de superación y se necesita mostrar un número menor de estos en el listado de los mismos en vez de la lista tan exhaustiva en una sola página.
GSP8	Recoger la especialidad del profesor	Recogidas por el usuario	Perfectivo	Incorporar en los datos del profesor la especialidad a la que este se dedica.	Media	En el propio funcionamiento del negocio se ha hecho necesario recoger la especialidad del profesor para vincularlo con tareas y actividades de superación a fines con su especialidad.
GSP9	Mostrar en PDF información del profesor	Mejora Propuesta	Perfectivo	Mejorar el formato de los datos del profesor en un formato estándar y reconocido de reporte.	Baja	Cuando se necesita ver los datos generales de un profesor se muestra una página en un nuevo navegador y sobre formato html. Este formato no se corresponde con los tipos clásicos de documentos a utilizar o a imprimir como pudiera ser un fichero de extensión .doc o un pdf.
GSP10	Opción de Modificar Profesor	Detectada en la aplicación	Correctivo	Chequear y corregir la falla que imposibilita una adecuada modificación en el sistema de los datos del profesor.	Alta	Cuando se modifica la información correspondiente a un profesor que se acciona el botón de modificar, se pasa a una página que se muestra totalmente en blanco, y al retornar a ver la información del mismo se aprecia que no se hizo efectiva la modificación.

Tabla #3 Sección de identificación de modificaciones al sistema de la planilla de actividades iniciales correspondiente al módulo de Profesores.

Para la realización y la estimación de la duración de las actividades de mantenimiento y la identificación del impacto de la actividad se documentó la sección de la planilla inicial de manera que quede recogido y estimado el tiempo que se dedicará a dicha actividad. En la planificación del tiempo asociado a las actividades de mantenimiento se usó la técnica que implementa diagrama de Gantt. Este diagrama muestra en el eje de las ordenadas las tareas correspondientes a desarrollarse en la propuesta de mantenimiento y en el eje de las abscisas el tiempo. Este diagrama permite observar con detalle la evolución del proceso. Aunque la propuesta no se atañe a ningún método específico dejando al equipo de realización el uso de la técnica que se ajuste a sus necesidades y se considere más adecuado para ello.

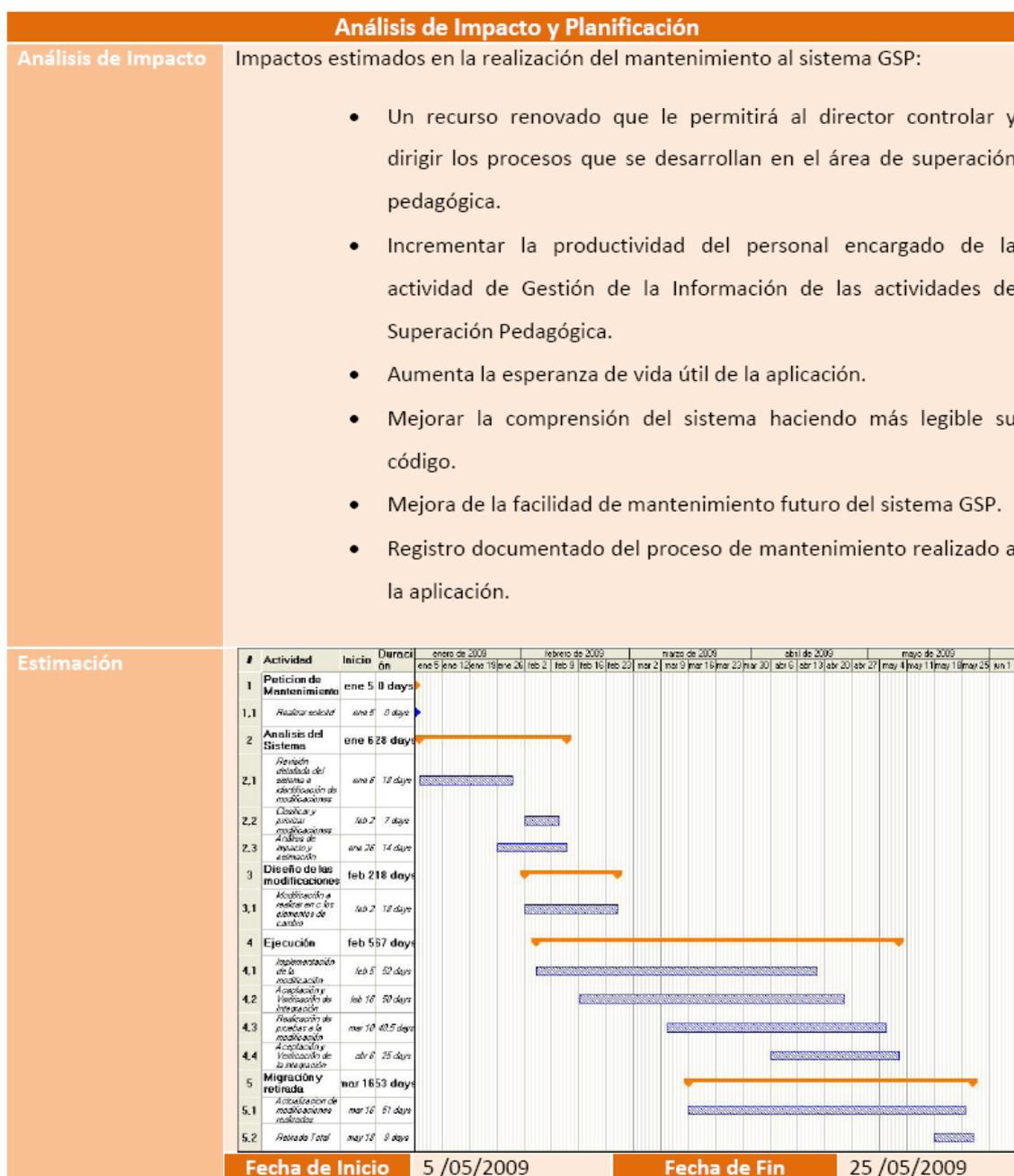


Tabla #4 Sección de Análisis de Impacto y Planificación de la planilla de actividades iniciales.

Los participantes asociados en la realización de estas actividades se exponen en la sección final de la planilla donde se muestran los roles asociados a estas actividades iniciales, que forman parte del equipo de mantenimiento y que está catalogado como de pequeño. El mismo lo integran 2 participantes en los roles de mantenedores dedicados a esta actividad, lo que recoge las condiciones necesarias para la aplicación de esta propuesta.

Participantes	Responsabilidad
Ingeniero Informático	Realizar el estudio del sistema, análisis del mismo y levantamiento de los ítems de modificación a realizarse en el mantenimiento.
Usuario del sistema	Identificar y notificar las deficiencias encontradas y las mejoras necesarias a incorporar en los sistemas según las necesidades surgidas en la interacción y uso del mismo.
Elaborado por:	Ing. Dania Souchay Fábrega
Fecha:	31/03/2009

Tabla #5 Sección que recoge de los participantes en las actividades iniciales del equipo de mantenimiento al GSP.

Las realizaciones de las actividades intermedias reflejaran el diseño e implementación de los elementos a modificar identificados anteriormente. A continuación se refleja dicha planilla para el módulo de profesores.

PLANILLA DE ACTIVIDADES INTERMEDIAS



Modificación del software

Sistema GSP. Sistema para la gestión de la superación pedagógica en el CREA.

Ejecución de la Modificación

# ITEM	Cambio propuesto	Revisión y aceptación	
		Pruebas	Integración
GSP5	Definir e implementar una clase CGestion_PROFESOR la cual se encargue de recoger todo lo concerniente a la gestión de profesores como Insertar, Modificar, Eliminar, Devolver un listado, Devolver datos de un profesor, Cantidad de profesores registrados, Buscar un profesor.	Satisfactoria	Satisfactoria
GSP6	En la página de listado de profesores en la parte superior derecha se incorpora un el control input edtBuscar el cual permitirá introducir el criterio de búsqueda correspondiente para filtrar los profesores.	Satisfactoria	Satisfactoria
GSP7	En la página de listados de profesores en la parte inferior antes del pie de página se incluye enlaces correspondientes a páginas desde 1 hasta la cantidad necesaria de manera que no se muestren en cada paginación más de 10 profesores. Se usara para ello librerías o bibliotecas para la implementación del paginado. La biblioteca estará almacenada en la carpeta de Útiles dentro de la estructura del sitio.	Satisfactoria	Satisfactoria
GSP8	Incorporación dentro del formulario de Nuevo profesor, Modificar profesor el control input edtresenna. Adicionar en la tabla Profesores de BD ASPE el campo de tipo varchar(50) resenna que dará soporte al registro de esta nueva información. Modificar las sentencias SQL asociadas a la manipulación de datos del profesor e incorporar el nuevo elemento de registro.	Satisfactoria	Satisfactoria
GSP9	Utilizar la librería de class.ezpdf para la generación de ficheros .pdf desde php en la Web. La biblioteca estará almacenada en la carpeta de Utiles dentro de la estructura del sitio.	Satisfactoria	Satisfactoria
GSP10	Revisar la sentencia SQL de modificación del profesor, y el redireccionamiento de los datos modificados del formulario hasta la página controladora que registra en la BD y detectar el error correspondiente.	Satisfactoria	Satisfactoria

Participantes	Responsabilidad
Usuario del sistema	Revisar y aceptar las modificaciones realizadas al sistema en la actividad de mantenimiento.
Ingeniero Informático	Realizar las modificaciones necesarias al sistema, según las modificaciones propuestas, realizar las pruebas y ver si el usuario final acepta y está de acuerdo con dichas modificaciones a realizarse en el mantenimiento.
Elaborado por:	Ing. Dania Souchay Fábrega
Fecha:	13/04/2009

Tabla #6 Sección que recoge las modificaciones a los elementos identificados en el análisis realizado en las actividades iniciales del equipo de mantenimiento al GSP Módulo de profesores.

A continuación se muestra en detalles las pruebas realizadas a las modificaciones realizadas en el módulo de profesores.

Modulo: Gestionar Profesores.
Caso de prueba: Registrar información requerida de los profesores mediante el método insertar de la clase CGestion_PROFESOR.
Entrada: Se registran los datos del profesor en el formulario de la página nuevo_profesor.php y se direccionan a la página intermedia agregar_profesor.php donde se capturan los datos del nuevo profesor registrado y se le pasan a la función insertar de la clase definida anteriormente, redireccionándolo al listado de profesores posteriormente.
Resultado: <ul style="list-style-type: none"> • Se inserta en la base de datos una tupla correspondiente a los datos del nuevo profesor cuya información fue suministrada. • Aparece el nuevo profesor registrado en el listado de profesores existentes.

Modulo: Gestionar Profesores.
Caso de prueba: Buscar Profesores que cumplan un determinado criterio de selección.
Entrada: Se ingresa en el campo de búsqueda incorporado en la página listado_profesores.php el criterio de “crea” y se oprime el botón que hará efectiva la búsqueda en los distintos campos asociados al profesor que contengan la combinación “crea”.
Resultado: <ul style="list-style-type: none"> • Se muestra un listado de profesores cuya institución asociada es el CREA, profesores cuya especialidad temática investigativa es la <i>creatividad</i>.

Modulo: Gestionar Profesores.
Caso de prueba: Mostrar el listado de los profesores realizando un paginado de los mismos.
Entrada: Se realiza la solicitud del listado de los profesores desde el menú principal de la aplicación GSP.
Resultado:

- Se muestran los 10 primeros profesores ordenados alfabéticamente en la pantalla de listado de profesores.
- Aparece en la parte inferior del listado un pie con una numeración desde 1 hasta 9 constituyendo hipervínculos a nuevos listados de profesores que se pueden navegar individualmente y en el extremo aparece un hipervínculo *siguiente* para ir transitando por cada uno de ellos. Una vez en el listado del paginado no correspondiente al inicial aparece un hipervínculo *anterior* indicando el retorno a la lista precedente.
- Aparece en negritas y entre paréntesis el total de profesores a listar: (90 profesores) que se corresponde con el total de profesores registrados en la aplicación.

Modulo: Gestionar Profesores.

Caso de prueba: Especialidad a la cual pertenece el profesor.

Entrada:

Se modifica el campo correspondiente a la “reseña” en el formulario de la página nuevo_profesor.php y modificación_profesor.php por “especialidad” correspondiéndose con el campo de la actividad investigativa que realiza el profesor.

Resultado:

- Aparece la opción de registro de la especialidad investigativa a la cual se dedica el profesor.

Modulo: Gestionar Profesores.

Caso de prueba: Reporte de la información general del profesor.

Entrada:

Se solicita desde la página ver_profesor.php una versión imprimible de la información del mismo que recoja todos los datos personales de un profesor en particular: “Dra. Elsa Herrero Tunis”.

Resultado:

- Se genera en una nueva ventana un informe sobre AcrobaReader PDF que contiene los datos personales de la Dra. Elsa Herrero Tunis (nombre, categoría docente, teléfono, especialidad, grado científico, correo), que incluye el logotipo del centro CREA y las credenciales de quien genera el reporte además de la fecha y la hora de su conformación.
- Un informe profesional correspondiente a los estándares de presentación de información para su posterior utilización.

Modulo: Gestionar Profesores.
Caso de prueba: Modificar información del profesor.
Entrada: Se actualizan los datos correspondientes al profesor que se necesita actualizar la información en la página de modificación_profesor.php: Modificar el correo del profesor Dr. Tomás Cañas Louzau que actualmente es tcanas@tesla.cujae.edu.cu por tcanas@crea.cujae.edu.cu . Una vez realizado el cambio correspondiente accionar el botón de modificar para hacer efectiva la modificación.
Resultado: <ul style="list-style-type: none">• En el ver_profesor.php se muestran los datos generales del profesor Dr. Tomás Cañas Louzau y el correo que fue el campo actualizado pues se muestra con la modificación efectiva realizada.

Como culminación de la realización del mantenimiento que del que está siendo objeto el sistema GSP se tiene la planilla de actividades finales que recoge las actividades de migración y retiro del sistema anterior por la nueva versión desarrollada.

PLANILLA DE ACTIVIDADES FINALES



Migración del software		
Sistema:	GSP. Sistema para la gestión de la superación pedagógica en el CREA.	
Módulos	Migración y Retiro Parcial	
Generalidades	Elementos de cambio (GSP1, GSP2, GSP3, GSP4) Actualización para la migración parcial 5/marzo	
Profesor	Elementos de cambio (GSP5, GSP6, GSP7, GSP8, GSP9, GSP10) Actualización para la migración parcial 20/marzo Capacitar en la uso de la opción del Buscar dentro del la listado de profesores	
Curso	Elementos de cambio (GSP11, GSP12, GSP13, GSP14) Actualización para la migración parcial 10/abril Capacitar en la uso de la opción del Buscar dentro del la listado de Cursos	
Retiro		
<ul style="list-style-type: none"> El 18 de mayo terminó satisfactoriamente la migración y el retiro parcial de cada uno de los módulos del sistema. Se mantendrá en periodo de prueba durante 3 días con las actualizaciones realizadas. El 22 de mayo se procederá al retiro total del sistema versión 1.0 del sistema para oficializarlo en su versión 2.0 dando por concluido el proceso de mantenimiento. 		
Comentario		
<ul style="list-style-type: none"> La migración y el retiro se realizarán por módulos una vez concluidos satisfactoriamente las modificaciones realizadas a los elementos de cambios asociados a un módulo del sistema. Se realizará una salva de los ficheros y de la Base de Datos del sistema GSP como medida de protección antes de comenzar las actividades de migración y retiro. Se mantendrá en fase de observación y seguimiento en el uso de los módulos actualizados por un período de 3 días. El usuario del sistema ha estado incorporado dentro del equipo encargado del mantenimiento, por lo que domina cada una de las modificaciones realizadas al sistema. De ser necesario recibirá una pequeña capacitación en la operación de los nuevos elementos introducidos. 		
Participantes	Responsabilidad	
Ingeniero Informático	Realizar las modificaciones necesarias al sistema, según las modificaciones propuestas, realizar las pruebas y ver si el usuario final acepta y está de acuerdo con dichas modificaciones a realizarse en el mantenimiento.	
Elaborado por:	Ing. Dania Souchay Fábrega	Fecha: 20/05/2009

Tabla #7 Planilla que recoge la planificación para la migración y el retiro del sistema GSP.

4.4 Evaluación de la calidad de software después de la ejecución de la propuesta

En el mantenimiento del sistema GSP se propuso para la evaluación de la misma la técnica del Índice de Mantenibilidad como característica de calidad fundamental a tener en cuenta en esta investigación.

Anteriormente al mantenimiento se realizó una evaluación de la misma arrojando que la mantenibilidad asociada al sistema era considerada como baja. Una vez realizado el Mantenimiento del Software se volverá a hacer una evaluación de esta misma característica para ver si experimentó algún cambio significativo una vez que se concluya dicha actividad.

Nuevamente sobre la base del polinomio del IM:

$$IM = 171 - 5.2 * \ln(\text{aveV}) - 0.23 * \text{aveV} (g) - 16.2 * \ln(\text{aveLOC}) - 50 * \sin(\text{sqrt}(2.4 * \text{perCM}))$$

aveV: es la media del volumen por módulo según Halstead.

A continuación se realiza el cálculo del volumen del módulo de profesores según la propuesta de Halstead y la complejidad ciclométrica por cada uno de los ficheros que lo componen el módulo de profesores:

Ficheros	n1	N1	n2	N2	g'
Agregar_profesor.php	6	20	15	20	1
Eliminar_profesor.php	6	9	8	18	1
Listado_profesores.php	64	312	49	163	14
Cgestion_profesor.php	14	129	27	119	9
Modificación_profesor.php	128	154	41	174	13
Modificar_profesor.php	6	14	16	61	1
Nuevo_profesor.php	37	187	11	25	7
Ver_profesor.php	29	105	18	31	15
Validacion_profesor.php	31	88	22	68	14
Imprimir_profesor.php	19	213	17	39	9

$$n = n1+n2$$

$$n=615$$

$$N = N1+N2$$

$$N=1941$$

$$V = N * \log_2 n$$

$$V = 1941 * \log_2 615$$

$$V = 1941 * 2.8$$

$$V = 5435$$

$$\mathbf{aveV = 5434}$$

AveV (g'): es la media de la complejidad ciclomática por módulo, que mide el número de caminos linealmente independientes a través de un módulo de programa. (m = Número de condiciones + 1)

$$\mathbf{AveV (g') = 84}$$

aveLOC: es la media del número de líneas de código por módulo.

$$\mathbf{aveLOC = 627}$$

perCM: es la media porcentual de líneas de código comentadas.

$$\mathbf{perCM = 346}$$

$$IM = 171 - 5.2 * \ln(aveV) - 0.23 * aveV (g') - 16.2 * \ln(aveLOC) + 50 * \sin(\sqrt{2.4 * perCM})$$

$$IM_A = 171 - 5.2 * \ln(5434) - 0.23 * 84 - 16.2 * \ln(627) + 50 * \sin(\sqrt{2.4 * 346})$$

$$\mathbf{IM_A = 71.4}$$

Este resultado del mantenimiento realizado permite constatar los esfuerzos realizados, si se compara el resultado obtenido después de finalizado el mantenimiento que es de un 71.4 con los límites de valores establecidos por la catalogación del índice de Mantenibilidad puede clasificarse de discreto.

4.5 Resultados Obtenidos

Luego de realizada la actividad del mantenimiento al sistema GSP aplicando el procedimiento propuesto se obtuvo un resultado positivo en la calidad del software en función de la característica de mantenibilidad. Aunque el valor obtenido resulta discreto con relación a los extremos que fija el índice de Mantenibilidad utilizado si de manera considerable con valor existente inicialmente que fue de 46.6.

Seguidamente se presenta un resumen de los resultados obtenidos después de aplicada la propuesta donde se presentan los factores que reflejan la mejora de la mantenibilidad obtenida.

Modificación en la estructura y organización de la aplicación para una mejor reutilización del código común que se ve reflejada en el número de ficheros que componen el sistema. En el siguiente gráfico se ilustra como en algunos módulos el número de ficheros se incrementa, lo que se debe a la inclusión de ficheros con la clase controladora del módulo

en cuestión, y en los módulos que disminuye se debe fundamentalmente a la reestructuración de los ficheros que son reutilizables a varios módulos en común que se agruparon en un módulo de *útiles*, lo que indica un mejor diseño de las modificaciones a llevar a cabo.

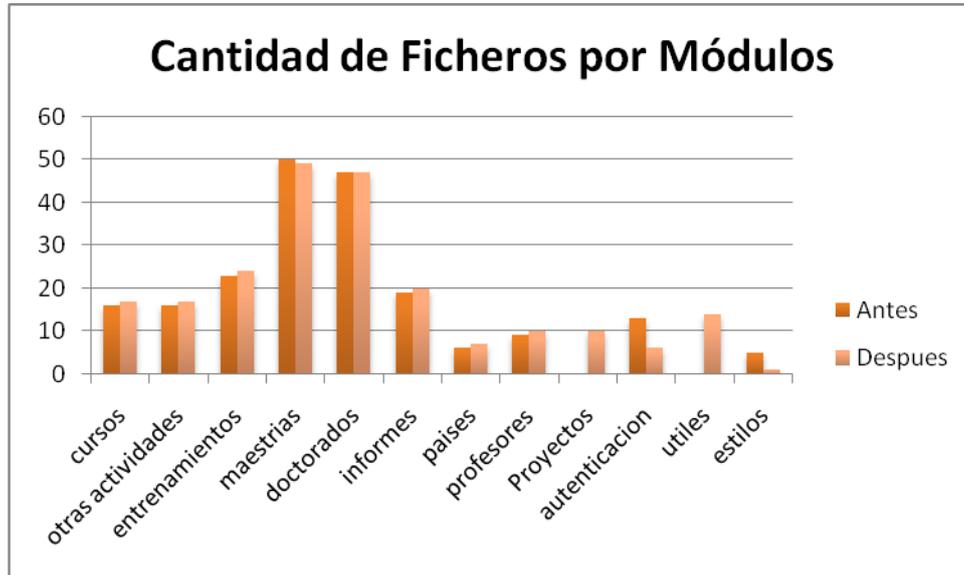


Fig. 16 Gráfica de cantidad de ficheros por módulos.

Por otra parte se obtuvo una disminución considerable en el número de líneas de código que componen los módulos. En el módulo de Profesores documentado se puede ver como se evidencia una disminución en cuanto a líneas de código por ficheros lo que redundo en una mejor comprensibilidad del código que lo componen mejorando las condiciones internas del mismo.

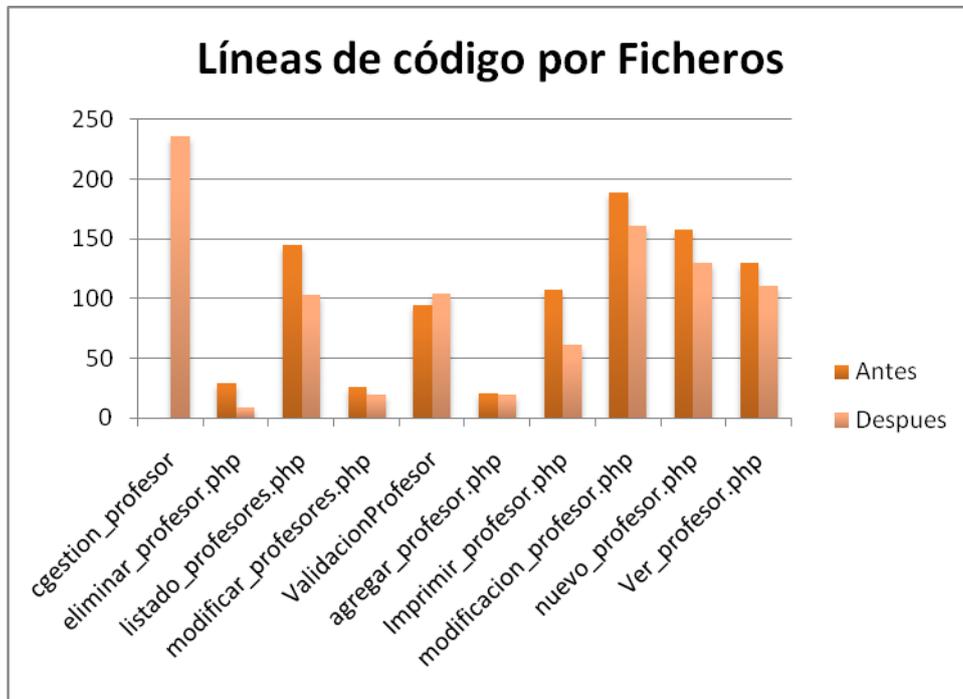


Fig. 17 Gráfica de cantidad de línea de códigos por ficheros.

De igual forma en el módulo de profesores haciendo un análisis de los comentarios por cada uno de los ficheros que componen el módulo, se aprecia un incremento del mismo con relación a la versión inicial que en su mayoría era prácticamente nulo por lo que denota un considerable aumento en la comprensibilidad de la aplicación.

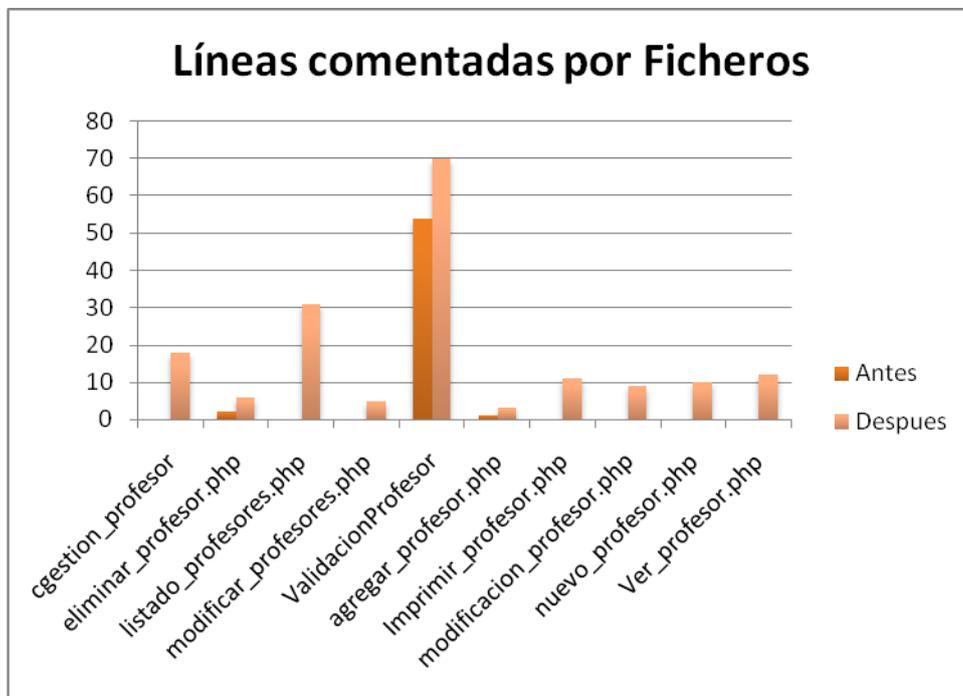


Fig. 18 Gráfica de cantidad de comentarios por ficheros.

Estos resultados evidencian el aumento de mantenibilidad que se obtuvo en el sistema GSP una vez realizada la actividad de mantenimiento utilizando el procedimiento propuesto. Se evidencia un código mejor organizado y con mayor legibilidad que puede ofrece un mayor nivel de comprensión y entendimiento que el anterior, lo que mejora sus condiciones internas del sistema considerablemente, se sigue una estructura común en cada uno de los módulos haciendo más entendible su composición y mejorando su mantenibilidad para futuras fases de mantenimiento incluso por personas que no hayan participado directamente en su desarrollo.

4.6 Validación de la propuesta contra el modelo S3M

Teniendo en cuenta la actualidad e importancia de S3M como modelo que mide la capacidad de madurez de la función del Mantenimiento de Software de referencia internacional se realizó un cotejo entre este modelo y la propuesta realizada donde se entrevistó a profesionales con experiencia en el campo de CMMi, y que actualmente están al frente del proceso de mejoras que lleva a cabo la Universidad de las Ciencias Informáticas actualmente.

A continuación se pretende identificar el grado de prácticas relacionadas de la propuesta de Mantenimiento de Software realizada para guiar esta función en pequeñas organizaciones nacionales, con el Modelo de Madurez para el Mantenimiento de Software.

El cotejamiento entre el procedimiento propuesto con respecto al Modelo S3M se realizará acorde tiene como objetivo ubicar la práctica guiada por el procedimiento en el correspondiente nivel de madurez propuesto por este Modelo.

El modelo comprende y evalúa los estadios e la madurez en la realización de la actividad del mantenimiento de software que como es de esperar dentro de los primeros niveles pues establece atributos que identifican a una organización totalmente inmadura en el tema, lo cual hace que no se corresponda con las prácticas que establece la propuesta. Según va transitando por los distintos niveles de madurez se irán identificando algunas prácticas incorporadas en el modelo.

CT: Contemplado Totalmente

CP: Contemplado Parcialmente

EM: En alguna Medida

NC: No Contemplado

Atributos S3M	CT	CP	EM	NC	Argumentación
Nivel 0 Proceso Incompleto				x	
P0.1 No hay evidencia de que el proceso existe.				x	Este es el estado actual de muchas de las organizaciones nacionales como lo evidencio la encuesta inicialmente aplicada. Este nivel se refiere a atributos que caracterizan a organizaciones que sólo realizan el
P0.2 La alta gerencia no es consciente de las repercusiones de no contar con esta actividad o proceso en la organización.				x	

P0.1 La actividad o proceso no cumple los objetivos establecidos por el modelo.				x	trabajo diario de mantenimiento de software, que no manejan peticiones del usuario o de eventos de software. No se esta formalmente responsables de las actividades y prioridades dentro de la actividad de mantenimiento. La propuesta de procedimiento parte a raíz de la necesidad de ello evidenciada por parte de estos atributos.
P0.1 No hay conocimiento o la comprensión de la actividad o proceso.				x	
P0.1 Existen debates sobre la actividad o proceso que de llevarse a cabo, pero no se encuentran pruebas de que la actividad .				x	
P0.1 Los registros históricos muestran que la actividad se ha realizado, pero no se está haciendo en este momento.				x	
Nivel 1 Proceso Improvisado					
P1.1 La organización es consciente de la necesidad de llevar a cabo esta actividad o proceso.			x		La propuesta de procedimiento identifica dentro de sus actividades la solicitud de mantenimiento como muestra de necesidad de que se realice la actividad de mantenimiento en la organización. En este nivel aun se muestra el carácter empírico evidenciado en las organizaciones según la encuesta realizada. Dentro de los atributos que se destacan en este nivel, el procedimiento recoge algunas prácticas como son el uso de métrica y el intercambio con el usuario final.
P1.2 Una persona realiza la actividad o proceso y los procedimientos no están documentados.				x	
P1.3 Insumos y resultados precisos de la actividad o proceso no puede ser reconocidos.				x	
P1.4 No hay ninguna medida de la actividad o proceso.				x	
P1.5 Los resultados (productos) no se utilizan, no es fácil de utilizar, y no mantenerse al día, y su impacto es mínimo.				x	
Nivel 2 Proceso Gestionado					
P2.1 El proceso está documentado y seguido a nivel local.	x				El procedimiento propuesto incluye este nivel de madurez en gran medida debido a que es una propuesta documentada del proceso a realizar con sus correspondientes objetivos que seria identificada y adoptada por la organización, que incluye el cálculo de la mantenibilidad como métrica a utilizar.
P2.2 Los objetivos del proceso y las actividades son conocidos.	x				
P2.3 Las medidas cualitativas de algunos atributos se llevan a cabo.	x				
Nivel 3 Proceso Establecido					
P3.1 La práctica o proceso propuesto por el modelo se ejecuta.	x				La propuesta esta propuesta para que se adopte y se aplique durante el proceso de mantenimiento en las organizaciones y recoge las funciones a desempeñar por cada una de las actividades con sus roles correspondientes. Establece planillas para documentar y apoyar el proceso de mantenimiento.
P3.2 La misma práctica se utiliza a través de los grupos de mantenimiento de software.					
P3.3 Las medidas básicas se han definido y se recogen, verificados y comunicados.			x		
P3.4 Los empleados tienen el conocimiento para ejecutar la práctica o proceso (es decir, lo que implica que se definen las funciones y responsabilidades de las personas).	x				
P3.5 Técnicas, plantillas, datos del repositorio, y las infraestructuras disponibles y se utilizan para apoyar el proceso.		x			

P3.6 La práctica o el proceso es siempre utilizado por los empleados .	x				
P3.7 Las principales actividades del proceso son medidos y controlados.				x	
Nivel 4 Proceso Predecible					
La práctica es formalización. Hay una diferencia importante, en términos de la previsibilidad de los resultados de una práctica. La expresión " cuantitativamente gestionado " se utiliza cuando un proceso se controla mediante un control estadístico o una técnica similar adecuado para el control de la ejecución del proceso y sus actividades más importantes. La organización predice el rendimiento y controla el proceso.				x	No aplica en la propuesta
Nivel 5: Proceso Optimizado					
Implica una mejora continua. Los objetivos de mejora cuantitativa se han establecido y revisado para adaptarse a los cambios en los objetivos de negocio. Estos objetivos se utilizan como criterios clave para las mejoras. Impactos de las mejoras son medidos y evaluados con los objetivos de mejora cuantificados.				x	No aplica en la propuesta

Según el análisis anterior la propuesta de procedimiento comprende el nivel 2 y algunos atributos del nivel 3 según el Modelo de Madurez del Mantenimiento de Software por lo que la puesta en práctica del mismo en las organizaciones nacionales para realizar el Mantenimiento de Software de gestión permitirá:

- Elevar el grado de madurez actual en relación a la actividad de Mantenimiento de Software.
- Mayor visibilidad sobre el proceso de Mantenimiento de Software.
- Implementar prácticas más robustas y maduras en las organizaciones nacionales.
- Enfatizar el desarrollo de procesos en las organizaciones que permiten mejorar el mantenimiento de los productos y que responda a las expectativas organizacionales.

La propuesta de procedimiento para el Mantenimiento de Software resulta sencilla de aplicar en una organización y de fácil entendimiento para el equipo de aplicación. Constituye una propuesta de gran ayuda para las organizaciones en la realización de las funciones de este proceso como se evidencio en el mantenimiento del sistema GSP del CREA.

Su puesta en práctica permitió una mayor fiabilidad de las planificaciones, estableciéndose acuerdos claros sobre el servicio y la funcionalidad del producto a mantener. Permitted operar sobre la base de estándares documentados mejorando la calidad de los productos mantenidos.

Conclusiones Parciales

La propuesta del procedimiento para el proceso de Mantenimiento de Software se aplicó en el sistema de gestión sobre plataforma Web GSP perteneciente al centro CREA de la CUJAE. El procedimiento utilizado permitió llevar un hilo conductor guiado en cada una de las tareas a realizar en la actividad del mantenimiento de la aplicación y el uso de las planillas propuestas como herramientas posibilitó la recogida de una traza en los cambios realizados dejando a la vez documentación que complementa la ya existente sobre la trayectoria de vida del software GSP. La mayoría de los elementos de modificación fueron mejoras introducidas para elevar y facilitar la futura actividad de mantenibilidad del sistema y para actualizar ingenierilmente la aplicación con elementos que han evolucionado desde el momento en que fue desarrollada creada la aplicación hasta el momento.

El análisis de la mantenibilidad arrojó resultados favorables fundamentalmente en el mejoramiento de las características internas de la aplicación. El cotejamiento realizado entre la propuesta y el modelo S3M, dio como resultado que la utilización en las organizaciones nacionales pues logrará alcanzar un nivel discreto de madurez en el proceso de Mantenimiento de Software.

Conclusiones

En el presente trabajo se dio respuesta el problema planteado referente a la insuficiente adaptabilidad del proceso de mantenimiento dentro del ciclo de vida del software y se cumplió el objetivo general de este trabajo:

- Se desarrolló un procedimiento de Mantenimiento de Software para sistemas de gestión ajustado a las necesidades y características de pequeñas organizaciones.

Asimismo se cumplieron los objetivos específicos trazados para el trabajo:

- Se realizó el estudio de los elementos teóricos relacionados con contexto de Mantenimiento de Software y de su estado del arte a nivel mundial, lo que permitió conocer sus características, los diferentes tipos de mantenimiento a realizarse así diferentes estándares y metodologías existentes. Como resultado de este estudio se identificaron una serie de características y elementos asociados al proceso de mantenimiento, que se tuvieron en cuenta para el desarrollo de la propuesta.
- Se elaboró un procedimiento que permite guiar el proceso Mantenimiento de Software resultando útil para pequeñas organizaciones contribuyendo a suplir la carencia de formalización del proceso en la industria nacional, lo que tributa a la mejor ejecución del proceso y por tanto mayor calidad en el producto mantenido.
- Se aplicó el procedimiento propuesto en el mantenimiento del sistema de gestión GSP del CREA que permitió una adecuada planificación y ejecución del proceso, así como la documentación del mismo de forma clara y organizada. Para evaluar los resultados obtenidos se aplicó la métrica del índice de mantenibilidad, que reflejó un discreto aumento de esta característica, basado fundamentalmente en el mejoramiento de las condiciones internas de este sistema.

Recomendaciones

- Estudiar herramientas que puedan complementarse con el proceso de Mantenimiento de Software y que permitan la automatización del procedimiento propuesto.
- Profundizar en las métricas a utilizar en cada una de las actividades de la propuesta de mantenimiento de software que permita medir aspectos de interés relacionado con la calidad del software durante la fase de mantenimiento a fin de introducir futuras mejoras a la propuesta.
- Llevar a la herramienta EFP composer descrita en el capítulo 2 epígrafe 2.4 el procedimiento propuesto para una mejor documentación y estandarización del mismo.

Referencias Bibliográficas

1. **Rodríguez, Alfredo, Márquez, Antonio y Toro, Miguel.** Gestión de la evolución del software. El eterno problema de los Legacy Systems. [En línea] [Citado el: 27 de Enero de 2009.] <http://lsi.ugr.es/~gedes/actividades/tallerEvolucion2001/docs/1DEF.pdf>.
2. **Sicilia, Miguel-Angel.** Definiciones de Mantenimiento del Software. [En línea] [Citado el: 5 de mayo de 2009.] <http://cnx.org/content/m17404/latest/>.
3. **Ruiz, Francisco y Polo, Macario.** Mantenimiento del Software. *Curso de Mantenimiento del Software*. [En línea] [Citado el: 27 de Enero de 2009.] <http://alarcos.inf-cr.uclm.es/doc/mso/>.
4. **Scalone, Fernanda y García Martínez, Ramón.** *ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE*. Junio de 2006.
5. **Lientz B. P., Swanson E. B.** *Software Maintenance Management*. : Addison Wesley, 1980.
6. **López Trujillo, Yucely y André Ampuero, Margarita.** *Disciplina en el desempeño de roles para la Industria de Software Cubana*. Ciudad de la Habana : Instituto Superior Politécnico “José Antonio Echeverría” Centro de Estudios de Ingeniería de Sistemas, 2006.
7. Evolución del Software. [En línea] [Citado el: 28 de Enero de 2009.] <http:// triana.escet.urjc.es/aspf/ASPF-Tema1.pdf>.
8. Evolución del Software y Mantenimiento. [En línea] [Citado el: 28 de Enero de 2009.] <http:// triana.escet.urjc.es/aspf/ASPF-Tema1.pdf>.
9. **Sicilia, Miguel-Angel.** Leyes de la Evolución del Software. *Conexion*. [En línea] [Citado el: 28 de Enero de 2009.] <http://cnx.org/content/m17406/latest/>.
10. **Pedraza García, Gilberto.** *Evolución e Integración de Aplicaciones Legadas: Comenzar de Nuevo o Actualizar?* 2008.
11. *Tecnología de la información Procesos del ciclo de vida del software (ISO/IEC 12207:1995)*. [Norma Internacional ISO/IEC 12207:1995.] s.l. : AENOR, 1999. UNE 71044:1999.
12. **Polo, Macario, y otros.** *MANTEMA: A Software Maintenance Methodology Based on the ISO/IEC 12207 Standard*. Washington, DC, USA : IEEE Computer Society, 1999. ISBN:0-7695-0068-4.
13. **Lamayzi, Samira.** *Norma ISO 14764 sobre mantenimiento de software*. 1999.

14. **Dávila, Abraham.** *Aplicacion de los proyectos informaticos utilizando la NTP ISO/IEC 12207.* [Documento] s.l. : Comite tecnico de normalizacion, Universidad Catolica del Perú.
15. **Pino, Francisco, Ruiz, Francisco y Salas, Sebastián.** *Agil Mantema. COMPETISOFT (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica).* [En línea] 28 de Febrero de 2008. [Citado el: 7 de Mayo de 2009.] <http://personales.unican.es/ruizfr/is2/doc/teo/1/is2-t01-agilmantema.pdf>. IT 23.
16. **Estrada, MsC. Lic. Ailyn Febles y Febles Estrada, Ailyn.** *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software.* La Habana : s.n., 2003.
17. **Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo".** *masternewmedia.* [En línea] 2 de Noviembre de 2005. [Citado el: 13 de Marzo de 2009.]
18. **Prof. Laura Noussan- Lettry.** *Herramientas y Software de Gestión . Taller de Software de Gestión .* [En línea] [Citado el: 21 de Mayo de 2009.] <http://tsg.lnoussanl.com.ar/pagina41.htm>.
19. **Francisco Ruiz.** *Ingeniería de Procesos Software. Universidad de Castilla-La Mancha.* [En línea] http://alarcos.inf-cr.uclm.es/per/fruiz/cur/ips/trans/ips_spem.pdf.
20. **Sicilia, Miguel Angel y De la Morena, Verónica.** *Métricas del Mantenimiento de Software. C O N N E X I O N S.* [En línea] 2008. <http://cnx.org/content/col10583/1.9/> .
21. **Vanmeert, Rob.** *Conviviendo con sistemas legados. Paradigma.* [En línea] [Citado el: 28 de Enero de 2009.] <http://paradigma.uniandes.edu.co/index.php>.
22. **Minguet Melian, Jesus M. y Hernandes Ballesteros, Juan francisco.** *Calidad del Software y su medida.* s.l. : Centro de estudios Ramon Arces, S. A., 2003.
23. **Aja Quiroga., Lourdes.** *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones.* [En línea] [Citado el: 2 de Febrero de 2009.] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.
24. **Bartle, Phil.** *Información para la gestión y gestión de la información.* [En línea] [Citado el: 2 de Febrero de 2009.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
25. **I Sánchez Negrette, Marcia.** *La Importancia del Posgrado. El Universitario.* [En línea] [Citado el: 2 de Febrero de 2009.] <http://eluniversitario.unne.edu.ar/2005/67/pagina/enfoques.htm>.
26. **CREA . . .** [En línea] [Citado el: 29 de Enero de 2009.] <http://teleformacion.cujae.edu.cu/crea/>.
27. **Zamuriano Sotés, Ing. Roberto Félix y Álvarez Cárdenas, Dra. Sofía.** *Las Inspecciones de Software y las Listas de Comprobación.* Abril, 2004.

28. *Aplicando Gestion del Conocimiento al proceso de Mantenimiento de Software*. **Vizcaino, Aurora, y otros**. 031, Valencia, España : Revista Iberoamericana de Inteligencia Artificial, 2006, Vol. 10. 1988-3064.
29. Mantenimiento de Sistemas de Información. *MÉTRICA. VERSIÓN 3 Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información* . [En línea] [Citado el: 29 de Junio de 2009.] <http://www.csi.map.es/csi/metrica3/msiproc.pdf>.
30. **Welker, Kurt D. y Oman, Dr. Paul W.** Software Maintainability Metrics Models in Practice. [En línea] [Citado el: 15 de Julio de 2009.] <http://www.stsc.hill.af.mil/crosstalk/1995/11/Maintain.asp>.
31. **Aldekoa, Gentzane, y otros**. EXPERIENCE MEASURING MAINTAINABILITY IN SOFTWARE PRODUCT LINES. [En línea] CIMNE, Barcelona, 2006. <http://www.onekin.org/slide/AssociatedFiles/conf2006jisbdgen.zip/06jisbdgen.pdf>.
32. **Jacobson, Ivar. Booch, Grady. Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. s.l. : S.I Pearson Education, S. A. , 2000.
33. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico. Sexta Edición.* . s.l. : Mexico: Mcgraw-Hill., 2006.
34. **Larman, Craig.** *UML y patrones. Introducción al análisis y diseño orientado a objetos.* . s.l. : Prentice Hall Hispanoamericana, S.A., 1999.
35. *Visual Paradigm for UML Model-Code-Deploy platform. Visual Paradigm.* . s.l. : <http://www.visual-paradigm.com/product/vpuml/>., 2007.
36. **ArgoUML.** *ArgoUML Project home. Tigris.org Open Source Software Engineering Tools.* [En línea] : <http://argouml.tigris.org/>., 2006.
37. **9001:2000., ISO.** *Organización Internacional para la Estandarización. ISO/FDIS 9001:2000 (E). Sistemas de Gestión de la Calidad-Requisitos.* 2000.
38. **Souchay Fabrega, Dania.** *Trabajo de Diploma GSP: Sistema de Gestión para la Superación Pedagógica.* La Habana : s.n., 2004.
39. **Society, IEEE Computer.** *SWEBOOK Guide to the Software Engineering Body of Knowledge.* 2004.
40. **RUP.** *Rational Unified Process.* . 2003.
41. Sánchez Téllez, Amaury. *Propuesta de Modelo de Soporte de Software para el GREHU.* Ciudad de la Habana: Instituto Superior Politécnico “José Antonio Echeverría” Centro de Estudios de Ingeniería de Sistemas, 2006.

Anexos

Anexo 1: ENCUESTA PARA EL DIAGNÓSTICO DEL MANENIMIENTO DE SOFTWARE

Encuesta de apoyo a la investigación para la obtención de una referencia en la realización del mantenimiento de software.

En su organización si se ha realizado mantenimiento de software a sistemas que apoyan el funcionamiento interno de la misma diga:

1. ¿Qué nivel de prioridad se le asigna a la actividad de mantenimiento de software dentro de su organización?
 Alta
 Media
 Baja

2. Cantidad de personas que se dedican a realizar la actividad de mantenimiento por software de tamaño normal:
 No se realiza
 Individual
 De 2 a 3
 Hasta 5
 5 o más

3. ¿Qué factores han influido negativamente en la calidad del proceso de mantenimiento de software?
 La falta de tiempo para diseñar las modificaciones cuidadosamente.
 La falta de personal para dedicar a esta actividad.
 Características internas del sistema considerablemente hostiles como diseño pobre de las estructuras de datos, mala codificación, lógica defectuosa.
 La falta de un procedimiento o metodología que guíe el proceso de mantenimiento.
 La ejecución del mismo mediante un estilo libre establecido por el propio programador.
 Los altos costes que presupone su ejecución.
 Lo considera como una actividad "poco creativa".
 Errores ocultos introducidos durante el mantenimiento.
 Poca participación de los usuarios directos del sistema en el proceso de mantenimiento.
 Falta de disponibilidad de documentación y especificaciones de diseño o la existente es incomprensible, incorrecta o insuficiente.
 Otros. ¿Cuáles? _____

4. ¿Existe en la organización un procedimiento o metodología que guíe la actividad de Mantenimiento de Software?
 Sí
 Estándar

- Metodología
- Modelo
- Procedimiento
- Otro.

Diga Cual _____

No

No sé

5. ¿Qué actividades relacionadas con el mantenimiento de software se acometen durante dicho proceso?

Comprensión del software y de los cambios a realizar

Modificación del software

Realización de pruebas

Migración y Retiro

Otras _____

6. ¿Cuenta con plantillas, artefacto, programas u otra herramienta para documentar la actividad de mantenimiento?

Si _____ No _____ No sé _____

Cual _____

7. ¿Cómo califica la calidad del mantenimiento de software realizado en su organización?

Alta

Media

Baja

8. ¿Una vez realizada la actividad de mantenimiento mejora la mantenibilidad del sistema o sea se hace más fácil la actividad de mantenimiento?

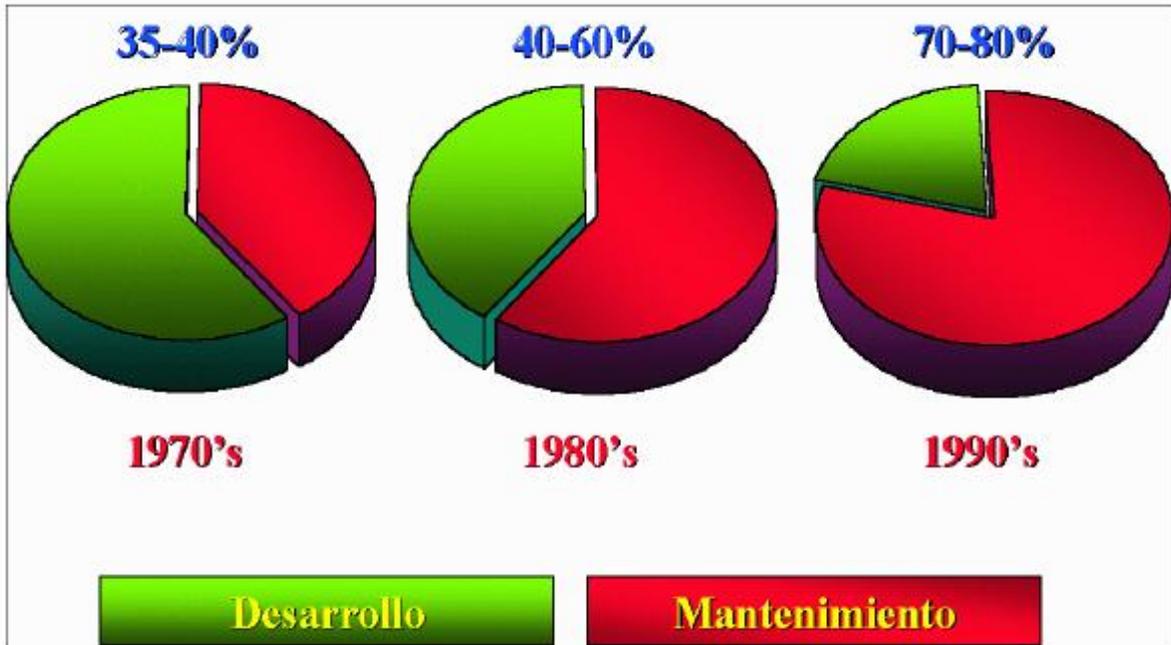
Mucho

Poco

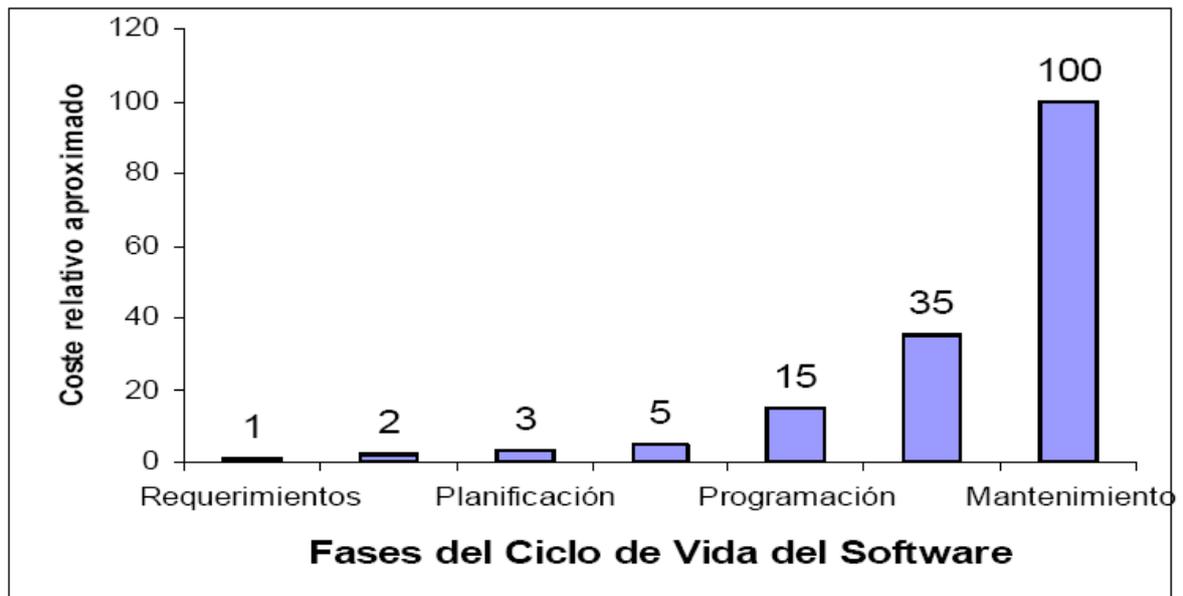
Nada

9. Observación :

Anexo 2: ENTREVISTA REALIZADA A USUARIO DEL GSP Y ESPECIALISTA PRINCIPAL DE TI DEL CREA

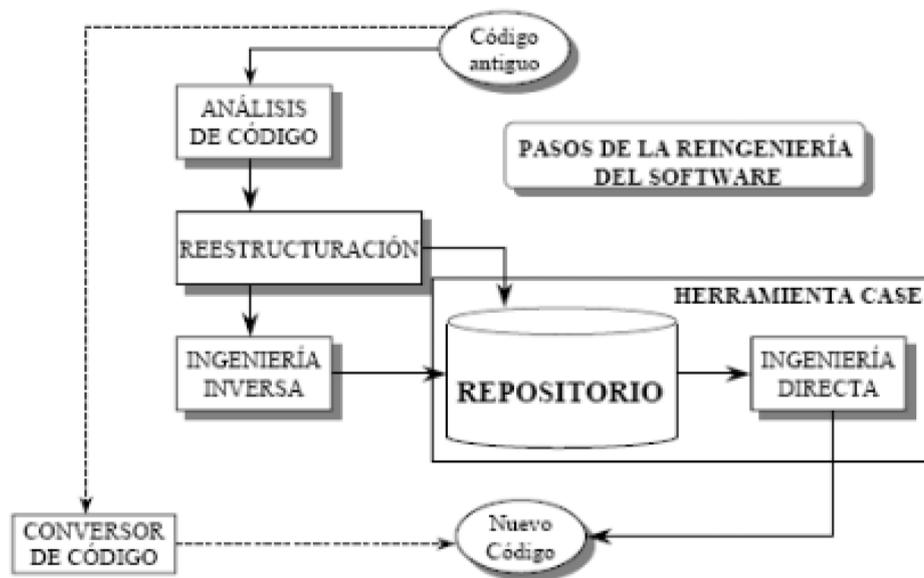


Anexo. 3 Tendencia por décadas de la elevación del costo del mantenimiento del Software



Coste relativo aproximado de detectar y corregir defectos.

Anexo. 4 Gráfica relacionada con los costos asociados a detectar y corregir fallos según las fases de del ciclo de vida del software.



Anexo. 5 Pasos de la Reingeniería del Software.

PLANILLA DE ACTIVIDADES INICIALES						
Solicitud de mantenimiento						
Entidad:		Fecha Solicitud:				
Sistema:						
Solicitante:						
Petición:						
Análisis general del sistema						
Identificación de modificaciones al sistema						
Modulo						
# ITEM	Elemento de Modificación	Origen	Priorización			Descripción
			Tipo	Alcance	Criticidad	
Análisis de Impacto y Planificación						
Análisis de Impacto						
Estimación		Fecha de Inicio	Fecha de Fin			
Comentario						
Participantes		Responsabilidad				
Elaborado por:		Fecha:				

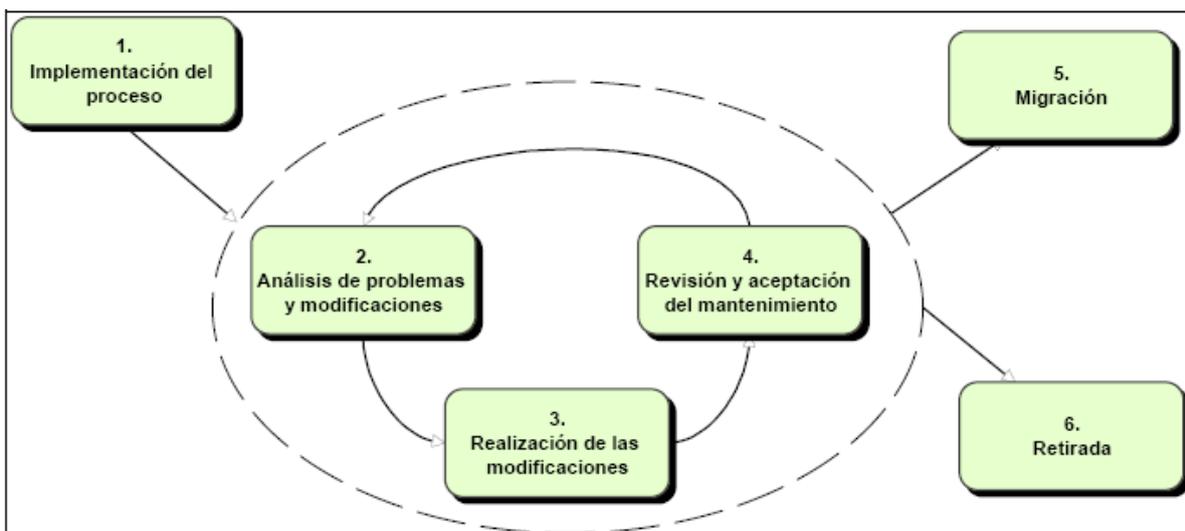
Anexo 6: Propuesta de plantilla para la realización de las actividades iniciales del mantenimiento de software.

PLANILLA DE ACTIVIDADES INTERMEDIAS			
Modificación del software			
Sistema			
Ejecución de la Modificación			
# ITEM	Cambio propuesto	Revisión y aceptación	
		Pruebas	Integración
Comentario			
Participantes		Responsabilidad	
Elaborado por:		Fecha:	

Anexo 7: Propuesta de plantilla para la realización de las actividades intermedias del mantenimiento de software.

PLANILLA DE ACTIVIDADES FINALES	
Migración del software	
Sistema:	
Módulos	Migración y Retiro Parcial
Retiro	
Comentario	
Participantes	Responsabilidad
Elaborado por:	Fecha:

Anexo 8: Propuesta de plantilla para la realización de las actividades finales del mantenimiento de software.



Anexo 9: Proceso de Mantenimiento según ISO.

Address: http://crea.cujae.edu.cu/creagegestion/aspectos/cursos/listado_cursos.php

GSP Gestión de la Superación Pedagógica

superación registrar informes

Superación / Cursos / Listado de cursos

Fecha Fin	Nombre	Modalidad	País	Participantes	Revisado	Puesto por
31-07-2003	Programación avanzada en JAVA	Presencial	Guatemala	32	OK	Tomaz Caña Louzau
31-07-2003	Programación Perl y JAVA	Presencial	Guatemala	36	OK	Tomaz Caña Louzau
15-08-2003	Formación de valores en el contexto escolar.	Presencial	Cuba	17	OK	Tomaz Caña Louzau

Anexo 10: GSP, Listado de Cursos antes de realizarse la actividad de mantenimiento.

Buscar

Superación / Cursos / Listado de Cursos

Fecha Fin	Nombre	País
31-05-2007	Pedagogía y Didáctica Universitaria para adiestrados	Cuba
30-05-2007	Fundamentos didácticos de la nueva Universidad Cubana	Cuba
09-05-2007	El uso de las TIC desde una Didáctica desarrolladora	Cuba
05-05-2007	Propuesta Didáctica para el desarrollo de valores desde las clases de Educación Física	Cuba
19-04-2007	Comunicación educativa	Cuba
17-04-2007	Presentaciones colectivas	Cuba
13-04-2007	Seminario de Tesis II	Cuba
11-04-2007	La aplicación de las TIC en la Educación II	Cuba
05-04-2007	Fundamentos didácticos de la nueva universidad cubana	Cuba
04-03-2007	Didáctica	Cuba

1 2 3 4 5 6 7 8 9 Siguiente >

Anexo 11: GSP, Listado de Cursos después de realizarse la actividad de mantenimiento.

Ingresos del Area - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://crea.cujae.edu.cu/creagegestion/aspectos/Informes/Reporte_Economico.php Ir

Reporte Económico

Cursos				
Fecha	nombre	Lugar	MN	USD
21-12-2004	Programación Perl y JAVA	Univ. Mariano Gálvez.	2	2
12-12-2002	La concepción del proceso de enseñanza-aprendizaje en el enfoque histórico-cultural.	III Simposio Iberoamericano de Investigación Educativa.	43	44

Actividades					
Fecha	nombre	Lugar	Tipo	MN	USD
12-12-2002	Impacto de las TIC en la Educación	III Simposio Iberoamericano de Investigación Educa	Mesa Redonda	22	22
12-11-2004	Impacto de las TIC en la formación de valores...	2o Foro Regional de Educación. Izúcar de Matamoros	Mesa Redonda	22	22

Listo Intranet local

Anexo 12: GSP, Formato de reportes antes de realizarse la actividad de mantenimiento.



GSP:Gestion de la Superación Pedagógica.

Información General de Ingresos Económicos

Ingresos de Cursos

Fecha	Nombre	Lugar	MN	CUC
10-12-2008	Pedagogía y Didáctica Universitaria para adiestrados	CREA-CUJAE	44	55

Ingresos de Otras Actividades

Fecha	Nombre	Lugar	Tipo	MN	CUC
12-12-2008	Solo el amor engendra la maravilla	silvio rodriguez	Seminario	34	34

Ingresos de Maestrías

Fecha	Nombre	Tutor(es)	País	MN	CUC
12-12-2008	Aplicación de las Tecnologías de la Información y las Comunicaciones (TIC) en la Educación Superior	1ª edición CREA			
		Solo el amor engendra la maravilla			
12-12-2008	Irene Bermúdez Lamadrid	Dra. Elsa Herrero Tunis	Cuba	600	8935

Maestranter

Anexo 13: GSP, Formato de reportes después de realizarse la actividad de mantenimiento.