

Universidad de las Ciencias Informáticas
Facultad 10



**Gestión de Configuración y Cambios para el grupo de Proyecto de
Gestión Documental y Archivística**



Trabajo de Diploma para optar por el Título de
Ingeniero Informático


Autor:

Jorge Amed Prieto Valdés

Tutor:

Ing. Maxora Rorayma Castro Pérez

Ciudad de La Habana, _____ del 2009.
“Año 50 de la Revolución”



“Revolución es unidad, es independencia, es luchar por nuestros sueños de justicia para Cuba y para el mundo, que es la base de nuestro patriotismo, nuestro socialismo y nuestro internacionalismo.”

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Yo Jorge Amed Prieto Valdés me declaro como autor único de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estimen pertinente con este trabajo. Para que así conste firmo la presente a los _ días del mes de junio del 2009.

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Gestión de Configuración y Cambios para el grupo de proyecto de Gestión Documental y Archivistica” fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a ____.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Gestión de Configuración y Cambios para el grupo de proyecto de gestión documental y archivística.

Autor: Jorge Amed Prieto Valdés.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

- **Independencia**
- **Originalidad**
- **Creatividad**
- **Laboriosidad**
- **Responsabilidad**

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Firma

Fecha

Dedicatoria

A todas aquellas personas que de una forma u otra contribuyeron a que este trabajo se hiciera posible después de tantas noches de desvelo. A esta Revolución hija de las ideas guiada por nuestro Comandante en Jefe. A mis padres Bárbaro Prieto Corrales (viejo), por ser el padre que todo hijo quisiera tener a su lado por siempre, por su carisma y ternura, a mi madre María de los Ángeles Valdés Pérez (vieja) por darme los ánimos necesarios para que este sueño se cumpliera y tener fe en mí incluso en los momentos más turbulentos de mi vida, a mi hermana por estar siempre ahí cuando más la necesito para sermonearme y darme buenos consejos, de las maneras más sutiles, a mis abuelos, a mi novia y a mis amigos, a todos ustedes está dedicado este trabajo.

Agradecimientos

Quisiera agradecer primeramente a esta Revolución y a esta Universidad por haberme dado la oportunidad de ver realizado el mayor de mis sueños. A mis queridos padres por haberme dado el sostén necesario para mantener viva la llama de la fe y la esperanza para terminar mi carrera y convertirme en un profesional, por haberme dado una correcta educación y ser los autores intelectuales de este sueño realizado, por ser cómplices de mis problemas y sufrir calladamente mis sufrimientos, a mi hermanita Katy por ser mi guía y mi faro, a mis queridos abuelos que me han enseñado mucho de la vida y por demostrarme que el tesoro más grande que una persona puede ostentar es una familia unida, a mi novia Ray por tener una paciencia y una ternura inagotable y someterse a tantas noches de desvelo por mi culpa (un beso para ti, te quiero mucho), ojala nunca me tuviese que ir, al marido de mi hermana (Nilo) por haberme demostrado siempre lo que es ser un amigo y un hermano (Barca Campeón), a mis suegros (los mejores del mundo). A mi tutora Maxora (Xory) como cariñosamente le dicen, por ser participe en este triunfo. A todas mis amistades y compañeros de aula, los estimo mucho a todos y les agradezco que hayan aparecido en mi vida, aquí tienen un amigo para contar siempre y un hombro donde descansar y llorar y una casa en Pinar de Río donde se les recibirá siempre con los brazos abiertos.

...A todos y a cada uno de ustedes Muchas Gracias.

Resumen

En la Universidad de las Ciencias Informáticas (UCI), se encuentra el proyecto de Gestión Documental y Archivística (GDA), que bajo los fundamentos de desarrollar un sistema para la gestión de documentos y archivos, tiene como objetivo principal elaborar productos con una calidad óptima.

Este trabajo de diploma tiene como centro de atención la proposición e implantación de una estrategia de Gestión de Configuración y Cambios (GCC) para el proyecto de Gestión Documental y Archivística, con el objetivo de dar una solución a uno de los principales problemas que ha venido afectado a este proyecto, la inexistencia de una robusta GCC, de esta manera se contribuye a mejorar la eficiencia e integridad de los productos finales. Esta estrategia tiene como característica fundamental la de ser concebida bajo los preceptos de una metodología completamente novedosa, debido a ser esta, la piedra angular de la gestión de documentos. Esta metodología se denomina: Designing and Implementing Recordkeeping System (DIRKS), lo que puede traducirse al español: Diseño e implementación de sistemas para almacenamiento y mantenimiento de registros.

El punto de partida lo constituye una investigación exhaustiva de las tendencias actuales de la GCC a nivel Internacional, Nacional y en los proyectos productivos de la Universidad.

Tabla de Contenidos

Capítulo 1	6
Estudio del Estado del Arte de la Gestión de Configuración y Cambios.....	6
1.1 Introducción	6
1.2 Gestión de Configuración del Software (GCS).....	6
1.2.1 Procesos de Gestión de Configuración del Software.....	8
1.3 Conceptos generales concebidos en la configuración del software	9
1.3.1 Elementos de configuración de software	9
1.3.2 Versión	10
1.3.3 Variantes	10
1.3.4 Línea base	11
1.3.5 Control de configuración	12
1.3.6 Control de cambios	12
1.3.7 Informes de Estado	13
1.3.8 Control de versiones	13
1.3.9 Repositorio.....	14
1.4 Herramientas para la Gestión de la Configuración	14
1.4.1 International Business Machines (IBM)	14
1.5 Herramientas para el control de versiones	15
1.5.1 Concurrent Versions System (CVS).....	16
1.5.2 Visual Source Safe (VSS).....	17
1.5.3 Subversion (SVN).....	18
1.6 Herramientas para la gestión de proyecto.....	21
1.6.1 Gforge	21
1.7 Tendencias actuales de la GCS	23
1.7.1 Procyon	23
1.7.2 Softel.....	24
1.8 Estrategias de Gestión de Configuración en la UCI.....	25
1.9 Conclusiones	29
Capítulo 2	30
Definición de la estrategia de GCC para el grupo de proyecto GDA.	30
2.1 Introducción	30
2.2 Situación de la GCS del proyecto GDA	30
2.2.1 Uso de la herramienta para el control de versiones (SVN).....	30
2.2.2 Satisfacción del cliente	30
2.2.3 Documentación y control de cambios de los ECS.....	30
2.2.4 Cantidad de elementos de configuración en desarrollo	31
2.2.5 Cantidad de elementos de configuración en revisión	31
2.2.6 Cantidad de elementos de configuración terminados	31
2.2.7 Control de no conformidades o solicitudes de cambios realizadas por los clientes	31

2.2.8 Cantidad de solicitudes de cambios en espera de ser revisadas	31
2.2.9 Cantidad de solicitudes de cambios en revisión.....	31
2.2.10 Cantidad de solicitudes de cambios cerradas	32
2.2.11 Evaluación de los Procedimientos llevados a cabo en el proyecto	32
2.3 Organización del equipo de trabajo.....	32
2.3.1 Actividades que se realizan en cada rol	33
2.4 Procesos de Gestión de Configuración en el proyecto GDA.....	34
2.4.1 Definición y establecimiento de la configuración del software	35
2.4.2 Control de Versiones	41
2.4.3 Control de Cambios	43
2.4.4 Auditoría de la configuración.....	47
2.4.5 Generación de Informes de estado.....	48
2.5 Conclusiones	50
Capítulo 3	51
Implantación de la estrategia de GCC para el grupo de proyecto GDA	51
3.1 Introducción	51
3.2 Metodología a usar	51
3.3 ECS seleccionados	52
3.4 Proceso de Gestión de Configuración del Software en el proyecto GDA	59
3.5 Proceso de control de cambios en el proyecto	60
3.5.1 Estados definidos por los que pasa un cambio en el proyecto GDA.....	61
3.5.2 El comité de control de cambios. Su funcionamiento	62
3.6 Control de Versiones	63
3.7 Auditorías a la Configuración.....	74
3.8 Generación de Informes de Estado de la Configuración	74
3.9 Conclusiones	75
<i>Conclusiones Generales</i>	76
<i>Recomendaciones</i>	77
<i>Bibliografía Citada</i>	78
<i>Bibliografía Consultada</i>	80
<i>Anexos</i>	81

Introducción

A lo largo de la historia de la humanidad siempre que el hombre ha pretendido llevar a buen término cualquier proyecto iniciado, una de las primeras cuestiones que ha tenido que plantearse es cómo realizar un correcto control de su trabajo, de manera que se le de seguimiento a su evolución y resultados, y que además sea posible asumir y tramitar los cambios que puedan surgir, en vistas a que se afecte en la menor cuantía posible el curso normal de la planificación.

El proceso de construcción de software debido a la complejidad que en la mayoría de los casos conlleva, es consecuentemente uno de los procesos que mayor control reclama, y sin duda alguna uno de los que mayor cantidad de cambios debe enfrentar desde su misma concepción. Cuando estamos desarrollando o dando mantenimiento a un producto de software, los cambios son inevitables, de hecho, es lo único que se mantiene constante dentro de un proyecto de desarrollo; ya sea porque el cliente necesita que se efectúen cambios, porque se cometieron errores, o simplemente porque el entorno en el que se desenvuelve el producto evoluciona. La incorporación de esos cambios al software de forma descontrolada, frecuentemente provoca caos en los proyectos, retrasos en la entrega de los productos y problemas en su calidad.

Cualquier organización que desee administrar adecuadamente los cambios al software, se debe asegurar de que se propongan y se evalúen cuidadosamente, que las personas indicadas tomen decisiones sobre esos cambios, que los cambios se comuniquen oportunamente a todos los afectados, y que el proyecto incorpore los cambios de una forma disciplinada. Para conseguir esto se hace entonces imprescindible contar con un buen proceso de Gestión de Configuración de Software.

Los orígenes y el auge alcanzado por la disciplina de Gestión de Configuración en los últimos años, están dados debido al desarrollo de la sociedad actual, producto de las nuevas condiciones comerciales y sus requisitos que se modifican constantemente, así como las nuevas necesidades de los clientes que piden cambios en las funcionalidades de sus trabajos, además de las transformaciones que pueden ocurrir en las prioridades del proyecto o en su estructura organizativa, surgiendo así la necesidad de gestionar la configuración para cualquier fase en la que se encuentre el proyecto.

La Gestión de Configuración se realiza durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que este deja de utilizarse, esto facilita el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo. Permite además obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto y de la satisfacción del cliente.

Atendiendo entonces a la importancia de esta disciplina, este trabajo de diploma pretende concebir una estrategia que en lo adelante garantice una correcta Gestión de Configuración dentro del grupo de proyecto Gestión Documental y Archivística.

El grupo de proyecto Gestión Documental y Archivística (GDA) pertenece al polo productivo de la facultad 10: Gestión de de la Información y el Conocimiento. El objetivo central de este proyecto es el desarrollo de un sistema para la gestión de documentos y archivos.

Un Sistema de Gestión Documental Archivístico (SGDA) es un conjunto de operaciones y técnicas integradas en la gestión administrativa general, basadas en el análisis de la producción, tramitación y los valores de los documentos que se destinan a la planificación, el control, la utilización, la conservación y la eliminación o la transferencia de los documentos a un archivo, con el objetivo de racionalizar y unificar el tratamiento y alcanzar una gestión eficaz y rentable.

El modelo tradicional de gestión basado en papel presenta, al menos en el mundo empresarial, una serie de desventajas que entorpecen la exigencia dinámica de los negocios de la era globalizada y que son resueltas por los documentos electrónicos y los sistemas de gestión documental electrónica:

- ✓ Dificultad para la localización de los documentos.
- ✓ Lentitud y elevado costo de recuperación.
- ✓ Copias innecesarias.
- ✓ Demoras en el tratamiento y necesidad de rearchivo.
- ✓ Pérdida de documentos.

- ✓ Falta de lineamientos de normalización.
- ✓ Pérdidas de tiempo en la distribución.
- ✓ Costo del lugar de almacenamiento.
- ✓ Bajo control de los aspectos de seguridad y confidencialidad
- ✓ Baja calidad del servicio clientes/usuarios.
- ✓ Progresivo deterioro del proceso por desvalorización de la actividad.

Situación Polémica:

En el grupo de proyecto GDA no se tienen bien identificados y documentados las características funcionales y físicas de los elementos de configuración, por otra parte tampoco se cuenta con un plan bien estructurado que posibilite, no solo, controlar los cambios surgidos a lo largo del ciclo de vida de un producto sino también reportar el proceso e implantación de los mismos.

Todas estas problemáticas dificultan el seguimiento del estado de los productos durante su evolución, provocando que no siempre sean satisfechas las necesidades funcionales de los usuarios y que en ocasiones se incumplan con las expectativas de costo y fecha de entrega de los productos.

Problema científico: La carencia de una política que permita gestionar correctamente la evolución de los productos desarrollados por el grupo de proyecto: GDA.

Objeto de estudio: Los procesos de gestión de configuración de software.

Campo de acción: Los procesos de gestión de configuración de software para la gestión documental.

Objetivo General: Desarrollar una estrategia que guíe los procesos de gestión de configuración en el grupo de proyecto GDA.

Objetivos específicos:

- ✓ Establecer los procesos necesarios para controlar los cambios al software en el grupo de proyecto: GDA.

- ✓ Organizar la gestión de configuración del Grupo de Proyecto.

Hipótesis: Si se desarrolla una correcta gestión de la configuración en el proyecto GDA entonces se le dará cumplimiento a las tareas productivas de forma eficiente en el tiempo establecido.

Tareas a desarrollar:

- ✓ Realizar un estudio de la teoría y las tendencias actuales de la GCS.
- ✓ Realizar una investigación exhaustiva sobre el estado de la gestión de configuración en el proyecto GDA.
- ✓ Establecer los procesos que contendrá la estrategia a seguir para la GCS.
- ✓ Definir las tareas de la Gestión de Configuración que debe desempeñar cada rol dentro del equipo de trabajo.
- ✓ Detallar las actividades que se desarrollarán dentro de cada proceso de la estrategia.
- ✓ Identificar los elementos de configuración de software existentes en el proyecto GDA.
- ✓ Establecer los procesos de gestión de configuración y cambio dentro del proyecto GDA.

Los **métodos teóricos** utilizados en la investigación son:

Histórico-lógico: Se utilizó este método para hacer un estudio detallado de trabajos anteriores referentes a la GCS, como fueron: consultas a tesis, consulta a literatura y publicaciones especializada en el tema.

Análisis-síntesis: Este método permitió el procesamiento de información detallada y precisa de las tendencias actuales referentes a este tema, los cuales se ajustan al desarrollo de la investigación.

Los **métodos empíricos** utilizados son:

Entrevista: Se realizaron entrevistas para capturar las experiencias en el tema de líderes de proyecto, profesores y estudiantes.

Este trabajo de diploma contiene una introducción, tres capítulos, conclusiones y recomendaciones. En su capítulo inicial se realiza un estudio sobre el estado del arte de la disciplina GCS. En el capítulo 2 se reflejan la estrategia a seguir para controlar los cambios y gestionar la configuración y finalmente en el capítulo 3 se establece dicha estrategia dentro del proyecto GDA.

Capítulo 1

Estudio del Estado del Arte de la Gestión de Configuración y Cambios

1.1 Introducción

En este capítulo se realiza un estudio de las tendencias actuales de la GCS tanto a nivel internacional, nacional, como específicamente dentro de la universidad. A lo largo de todo el capítulo se verán reflejadas una serie de definiciones detalladas extraídas de las bibliografías que aportan mayor conocimiento del tema.

“El arte de coordinar el desarrollo de software para minimizar la confusión se llama gestión de configuración. La gestión de configuración es el arte de identificar, organizar y controlar modificaciones al software que se construye por medio de un equipo de programación. La meta es maximizar la productividad al minimizar las equivocaciones”

Babich.

1.2 Gestión de Configuración del Software (GCS)

En la investigación realizada para dar una definición de esta disciplina se realiza un acercamiento al criterio de diferentes autores y se pudo comprobar que aunque algunos dan una definición más sencilla y otros más completa ninguna discrepa de la otra y de manera general todas se centran en la organización y control del proceso de construcción del software.

Uno de estos criterios definen que : “La Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.”(1)

Una definición en términos sencillos de Gestión de Configuración es brindada por Angélica de Antonio: “Dentro de las disciplinas de control se encuentra la Gestión de la Configuración del Software, cuyo objetivo es mantener la integridad de los componentes del producto software, evaluar y controlar los cambios sobre ellos, y facilitar la visibilidad del producto.”(2) En realidad esta definición de gestión de configuración es muy simple aunque bastante abarcadora.

“La Gestión de Configuración del Software es uno de los procesos clave para toda organización dedicada a la Ingeniería del Software, ya que posibilita una mejor organización del desarrollo y mantenimiento, consiguiendo la visibilidad del producto y facilitando el resto de procesos de producción”.(3)

“La gestión de configuración del software es una actividad de protección que se aplica a lo largo de todo el proceso del software. La GCS identifica, controla, audita e informa de las modificaciones que invariablemente se dan al desarrollar el software...” (4)

En el proceso de desarrollo del software se obtienen como resultados un conjunto de elementos que según Pressman “...se pueden dividir en tres amplias categorías: 1. Programas de computadora (tanto en forma de código fuente como ejecutable), 2. Documentos que describen los programas de computadora (tanto técnicos como de usuario) y 3. Datos (contenidos en el programa o externos a él).”(4)

“Configuration management is unique identification, controlled storage, change control, and status reporting of selected intermediate works products during the life of software” (5)

En términos generales se podría deducir que la gestión configuración del software es una actividad de autoprotección que se aplica durante el proceso de desarrollo del software, ya que el cambio es algo inevitable y que puede surgir en cualquier momento, las actividades de la gestión de configuración del software pueden identificar y controlar el cambio, así como garantizar que este se ha implementado correctamente e informar a todos aquellos que puedan estar interesados.

1.2.1 Procesos de Gestión de Configuración del Software

“El estándar IEEE Std. 1074-1995 ([IEEE 1074]) para el desarrollo de procesos del ciclo de vida del software, establece el proceso de gestión de configuración del software como uno de los procesos integrales. Estos son los procesos necesarios para completar exitosamente las actividades del proyecto, y son utilizados para asegurar la finalización y calidad de las funciones del proyecto. Este proceso está conformado por las siguientes actividades:

1. Planificar la Gestión de Configuración.
2. Desarrollar la Identificación de la Configuración.
3. Realizar el Control de la Configuración.
4. Realizar la Contabilidad de Estado.”(1)

Sin embargo una versión más actualizada de la IEEE, la IEEE 1987 y 1990 establecen un grupo de actividades similar aunque más eficaces, estas son:

1. Identificación de la Configuración.
2. Control de Cambios en la Configuración.
3. Generación de Informes de Estado.
4. Auditoría de la Configuración. (6; 7)

La Dr. Aylin Febles en el modelo Mconfig.PM propone un conjunto de procesos que resulta nuevo en su composición, donde los elementos bases son los Elementos de Configuración de Software (ECS):

1. Identificación de los Elementos de Configuración.
2. Gestión de Cambios en la Configuración.
3. Gestión de Versiones.
4. Planificación de la Gestión de Configuración.
5. Generación de Informes y Comunicación del Estado de la Configuración. (8)

Antonio Navarro plantea en su libro *Gestión de la Configuración Software* una serie de actividades para la Gestión de Configuración, estas son:

1. Identificación de ECSs.
2. Control de versiones.
3. Control de cambios.
4. Auditoria de la configuración.
5. Informes de estado. (9)

Se puede concluir diciendo que existe una similitud entre los conceptos de varios autores acerca de los procesos a incluir en la Gestión de Configuración.

1.3 Conceptos generales concebidos en la configuración del software

1.3.1 Elementos de configuración de software

Angélica de Antonio plantea que: “A cada uno de los componentes de la configuración del software se le va a llamar ELEMENTO DE CONFIGURACIÓN DEL SOFTWARE (ECS). El ECS es la unidad de trabajo para la GCS. El término CONFIGURACIÓN DEL SOFTWARE designa, por tanto, el conjunto de todos los elementos de configuración del software de un proyecto”. (2)

Se define un elemento de configuración a una unidad física y/o lógica parte de un conjunto mayor de elementos, producida o adquirida, que por sus características es distinguible de las demás y cuya evolución interesa administrar.

Un elemento de configuración del software (ECS) es información que se crea como parte del proceso de ingeniería de software. En el extremo, se puede considerar que un ECS es una sola sección de una gran especificación o un caso de prueba de un gran conjunto de pruebas.

De manera más realista, un ECS es un documento, un conjunto completo de casos de prueba o un componente de un programa dado (por ejemplo, una función C++ o un applet de Java).

Los ECS están organizados para formar objetos de configuración susceptibles de catalogar en la base de datos del proyecto con un solo nombre.

1.3.2 Versión

Se puede definir una versión como una instancia de un elemento de configuración, en un momento dado del proceso de desarrollo, que es almacenada en un repositorio, y que puede ser recuperada en cualquier momento para su uso o modificación. El término se usa para señalar a un elemento de configuración del software que tiene un conjunto definido de características funcionales.

Al grupo de versiones distintas que aparecen según transcurre el ciclo de vida del proyecto y de un producto en sí se denomina revisiones.

En el grafo de evolución se describe la historia del objeto y sus cambios, las grandes modificaciones hacen que un objeto cambie, por lo que cambia el número de versión principal.

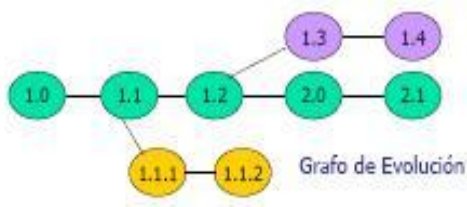


Figura 1 Grafo de evolución(10)

1.3.3 Variantes

Una variante es una versión de un componente (o de la configuración global) que evoluciona por separado. Las variantes representan una variación espacial, mientras que las revisiones representan una variación temporal.

Se define variante como una versión que es una alternativa a otra versión. Las variantes pueden permitir a

un elemento de configuración satisfacer requerimientos en conflicto. Una variante es una nueva versión de un elemento que será añadida a la configuración sin reemplazar a la versión anterior.

1.3.4 Línea base

Inmerso en el libro “La Gestión de la Configuración del Software”, del autor Antonio Navarro se encuentra una definición bastante exacta de la línea base, el cual plantea que la misma no es mas que:“(…) un punto de referencia en el proceso de desarrollo del software que queda marcado por la aprobación de uno o varios Elementos de Configuración del Software, mediante una revisión técnica formal.”(9)

Otra buena definición es la que ofrece la IEEE, que define una línea base como:

“Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.”(11)

Una línea base está constituida por los ECS que han sido aprobados por los miembros de la dirección del proyecto. Ya establecida la línea base solo se podrá transformar o cambiar mediante un proceso formal de cambios.

La Figura 2 nos indica que cada documento que resulta del proceso de software, se convierte en Línea Base, una vez que dicho documento se haya revisado, corregido y aprobado

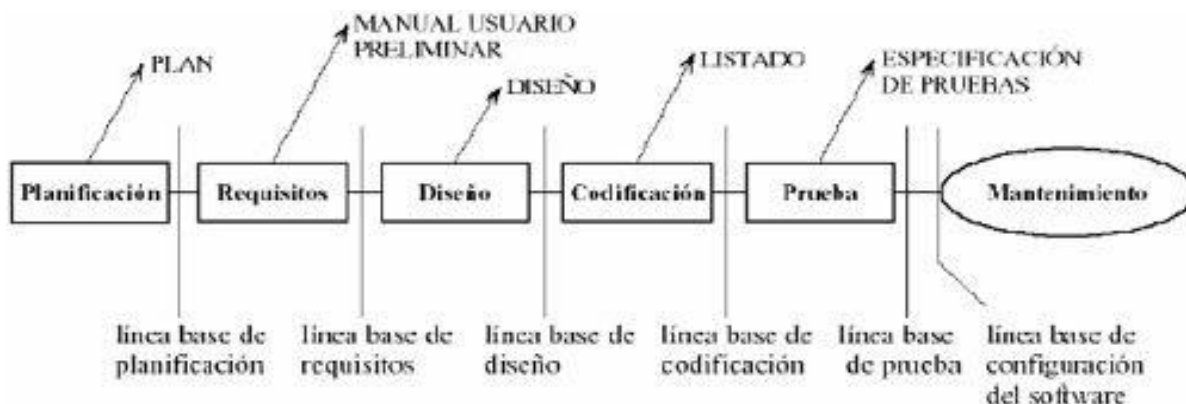


Fig. 2 Línea Base(12)

1.3.5 Control de configuración

Una “configuración” es una combinación de versiones particulares de los componentes que forman un sistema consistente. Desde el punto de vista de evolución, es el conjunto de las versiones de los objetos componentes en un instante dado.

1.3.6 Control de cambios

La primera Ley de la ingeniería de sistemas establece: “Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida”.

¿Cuál es el origen de estos cambios? La respuesta a esta interrogante es muy variada y puede ser respondida de distintas formas y maneras. Sin embargo todas parten de las cuatro fuentes del cambio, estas son:

- ✓ Nuevos negocios o condiciones comerciales que dictan los cambios en los requisitos del producto o en las normas comerciales.
- ✓ Nuevas necesidades del cliente que demandan la modificación de los datos producidos por sistemas de información, funcionalidades entregadas por productos o servicios entregados por un sistema basado en computadora.
- ✓ Reorganización, crecimiento o reducción del negocio que provoca cambios en las prioridades del proyecto o en la estructura del equipo de ingeniería del software.
- ✓ Restricciones presupuestarias o de planificación que provocan una redefinición del sistema o producto.

El control de cambio es un concepto relacionado con la metodología de desarrollo de software. Se trata de hacer el desarrollo de forma evolutiva, mediante cambios sucesivos realizados de una manera disciplinada. El control de cambios es el paso de una versión de la línea base a la siguiente, además

puede incluir modificaciones del contenido de algún componente y de la estructura del sistema, adicionando o eliminando componentes.

En un gran proyecto de desarrollo de software, el cambio incontrolado lleva rápidamente al caos. El control de cambios combina los procedimientos humanos y las herramientas automáticas para proporcionar un mecanismo para el control de cambio.

En un contexto de la ingeniería de software, James Bach lo resumió: “El control del cambio es vital. Pero las fuerzas que lo hacen también lo tornan irritante. Nos preocupamos por los cambios porque una pequeña perturbación en el código puede crear una gran falla en el producto. Pero también puede resolver una gran falla en el producto.”(12)

1.3.7 Informes de Estado

La generación de informes de estado de la configuración (IEC) consiste en elaborar un informe, que contenga:

1. ¿Qué pasó?
2. ¿Quién lo hizo?
3. ¿Cuándo pasó?
4. ¿Qué más se vio afectado?

El mero hecho de generar informes de estado es de gran importancia para el éxito del proyecto, estos informes posibilitan la comunicación entre los desarrolladores del proyecto.

1.3.8 Control de versiones

Consiste en mantener un registro histórico de las diferentes versiones por las que pasan los componentes de un producto, que permita la recuperación de cualquiera de ellas.

El control de versiones combina procedimientos y herramientas para gestionar las versiones de los objetos de configuración creadas durante el proceso de ingeniería del software. Varios sistemas de control de la

versión establecen un conjunto de cambio, una colección de todos los cambios que requiere la creación de una versión específica del software.

El usuario podrá especificar configuraciones alternativas del sistema mediante la selección de las versiones adecuadas de los objetos de configuración. Para ello, a cada versión del software se le asocia un conjunto de atributos y luego el usuario especifica la configuración mediante la descripción del conjunto de atributos deseados.

1.3.9 Repositorio

El lugar donde comúnmente se almacenan los componentes de un sistema, inclusive las distintas versiones de estos se denomina repositorio. El repositorio permite ahorrar espacio de almacenamiento, evitando guardar por duplicado elementos comunes a varias versiones o configuraciones, además facilita el almacenamiento de la información de la evolución del sistema (historial).

1.4 Herramientas para la Gestión de la Configuración

En este epígrafe se describen las principales características y funcionalidades de algunas de las herramientas para la gestión de configuración más usadas actualmente.

1.4.1 International Business Machines (IBM)

1.4.1.1 Rational ClearCase

“Ofrece una gestión fiable, ampliable y flexible de los activos de software para equipos de desarrollo de gran tamaño y tamaño medio.

IBM Rational ClearCase proporciona una gestión del ciclo de vida y control de los activos de desarrollo de software. Con un control integrado de versiones, una gestión del espacio de trabajo automatizado, un soporte de desarrollo en paralelo, una gestión de línea base y gestión de builds y releases.

Rational ClearCase proporciona las funciones necesarias para crear, actualizar, crear, ofrecer, reutilizar y mantener los activos más importantes del negocio.

- ✓ Rational ClearCase puede ayudar a aumentar la productividad a través del desarrollo en paralelo, obtener unos tiempos más reducidos del ciclo de build/release y una mayor reutilización del software. La integración con los IDE líder, incluido Rational Application Developer, WebSphere Studio, Microsoft Visual Studio .NET y la infraestructura Eclipse de código abierto agiliza aún más el desarrollo.
- ✓ Unas interfaces locales, remotas y web permiten un acceso prácticamente a todos sitios y en cualquier momento.
- ✓ Un soporte de desarrollo de Linux, Windows, Unix y mainframe (z/OS) permite un desarrollo de builds y de aplicaciones a nivel empresarial.
- ✓ Rational ClearCase se integra sin fisuras con Rational “(13)

1.4.1.2 Rational ClearQuest

“Rational ClearQuest proporciona un seguimiento flexible de defectos y cambios en toda la empresa.

- ✓ Seguimiento basado en actividad de cambios y defectos.
- ✓ Soporte robusto y flexible para flujos de trabajo, que incluye notificaciones por correo electrónico y opciones de envío.
- ✓ Fácil personalización mediante funciones de "apuntar y pulsar".
- ✓ Soporte completo para consultas con generación de multitud de informes y gráficos.
- ✓ Interfaz web para acceder fácilmente desde cualquier navegador web estándar.
- ✓ Integración transparente con Rational ClearCase para conseguir una solución SCM completa.
- ✓ Integrado con los IDE líderes en el sector, como WebSphere Studio, Eclipse y Microsoft .NET “(14)

1.5 Herramientas para el control de versiones

Las herramientas para el control de versiones varían mucho según las tendencias y preferencias mundiales. En este epígrafe se realiza un estudio profundo de las herramientas más importantes para el control de versiones aunque la herramienta a usar en el proyecto GDA está predefinida.

1.5.1 Concurrent Versions System (CVS)

CVS es una herramienta para el control de versiones que hoy en día es muy popular en el mundo del software libre debido a que es distribuido bajo la licencia GPL. Este es muy utilizado en los proyectos comerciales sobre todo en lo de código abierto. Es una herramienta que permite gestionar los cambios realizados sobre códigos fuentes de cualquier archivo en varias plataformas.

1.5.1.1 Principales Características:

CVS utiliza una arquitectura cliente-servidor, donde el servidor guarda la(s) versión(es) actual(es) del proyecto y su historial. Los clientes se conectan al servidor para sacar una copia completa del proyecto. Esto se hace para que eventualmente puedan trabajar con esa copia y más tarde ingresar sus cambios con comandos GNU.

- ✓ Una de sus características esenciales es que aunque el servidor utilice un sistema operativo libre, los clientes pueden trabajar en cualquier sistema operativo.
- ✓ Tiene por característica que varios clientes pueden sacar copias del proyecto al mismo tiempo, lo que en ocasiones puede provocar problemas en el momento de actualizar dichas modificaciones sobre un mismo fichero, en este caso el servidor deniega la última modificación e informa el conflicto al cliente, que debe resolver manualmente. Si la operación de ingreso tiene éxito, entonces los números de versión de todos los archivos implicados se incrementan automáticamente, y el servidor CVS almacena información sobre la actualización
- ✓ Los clientes pueden también comparar diferentes versiones de archivos, solicitar una historia completa de los cambios además de consultar un número de revisión determinado.
- ✓ No requiere que cada versión de los archivos sea guardada sino que se mantiene en un repositorio desde donde son administradas las versiones.

1.5.1.2 Desventajas:

- ✓ Los archivos en el repositorio sobre la plataforma CVS no pueden ser renombrados, estos deben ser agregados con otro nombre y luego eliminados, esta situación que provoca más trabajo a los clientes, además de que con este procedimiento de eliminar y añadir cada vez que se desea cambiar nombres, se pierde el historial de las revisiones.
- ✓ Para crear copias de seguridad es necesario interrumpir el acceso al repositorio.

- ✓ Las transacciones de datos no son atómicas, o sea, cuando se suben un conjunto de datos se van subiendo uno a uno, lo que provoca que no se pueda asegurar la integridad de la información y además provoca que todos los ficheros no se etiqueten con el mismo número de revisión en el repositorio.

1.5.2 Visual Source Safe (VSS)

Microsoft Visual SourceSafe (también conocido por sus siglas VSS) es una herramienta de Control de versiones que forma parte de Microsoft Visual Studio aunque está siendo sustituida por el Visual Studio Team Foundation Server.

1.5.2.1 Principales características:

- ✓ Está estructurada por una arquitectura cliente-servidor y además está compuesta por dos módulos, uno de cliente y otro de administración.
- ✓ Esta herramienta permite administrar cualquier tipo de archivos (texto, gráficos, sonido o video) y la almacena en un base de datos, la cual permite que los cambios que se le realicen a los archivos se guarden de manera que puedan ser recuperadas en cualquier momento las versiones anteriores.
- ✓ Lo commits no son atómicos.
- ✓ Para las personas que desarrollan programas bajo el sistema operativo Windows, resulta una herramienta útil ya que se integra fuertemente con el entorno de desarrollo integrado o IDE de Visual Studio permitiendo un manejo relativamente simple de versiones sobre una computadora individual y en equipos de trabajo relativamente pequeños.

1.5.2.2 Desventajas:

- ✓ La principal desventaja de Visual SourceSafe reside en el método de acceso a los archivos compartidos que constituyen su repositorio mediante el protocolo SMB que no impide que estos sean manipulados de manera externa al producto por cualquier persona que tenga acceso al mismo, provocando corrupción de datos. Este mismo tipo de acceso a archivos compartidos provoca que en equipos de trabajo grandes, el acceso concurrente pueda ser particularmente lento.

- ✓ El SourceSafe es inestable cuando se suben ficheros binarios de gran tamaño, ya que espera solo ficheros de texto. Así que no vale para almacenar documentación, sólo código fuente.

1.5.3 Subversion (SVN)

Subversion es un software de sistema de control de versiones diseñado específicamente para solucionar las limitaciones y problemas presentados por CVS. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos.

1.5.3.1 Principales características:

- ✓ Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo
- ✓ Es muy fácil de usar pues cada usuario puede acceder a él para realizar modificaciones desde un ordenador que forme parte de una red local o tenga conexión a Internet, además una instalación básica solo requiere conocimientos a nivel de usuario del sistema operativo.
- ✓ Es bastante seguro. El acceso al servidor se realiza mediante contraseñas. Permite además realizar copias de seguridad de toda la información.
- ✓ Consume pocos recursos.
- ✓ Los commits son atómicos.
- ✓ Permite trabajar directamente sobre el repositorio, sin necesidad de hacer copia local.
- ✓ La numeración de versiones es global, o sea, con SVN los números de versión son para todo el repositorio.
- ✓ Facilidad a la hora de eliminar y cambiar nombres a los elementos de configuración de software.
- ✓ La creación de ramas y etiquetas es una operación más eficiente.
- ✓ Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- ✓ Se envían solo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
- ✓ Puede ser servido mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion en forma transparente.

- ✓ Copias fáciles sobre ramificaciones: Independientemente del número de ramificaciones creadas mantiene un árbol diferencial de cambios.
- ✓ Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- ✓ Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, PAM).

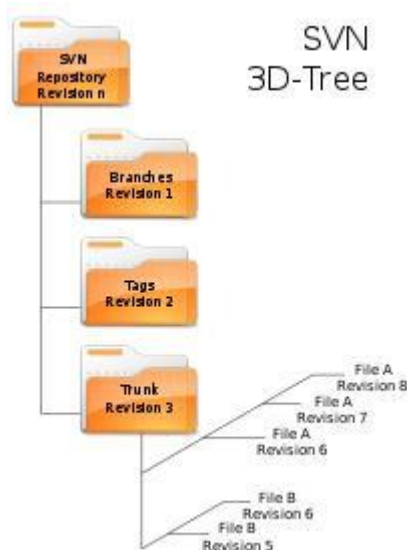


Figura 3 Árbol de proyecto subversion.

1.5.3.2 Desventajas:

- ✓ El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.
- ✓ No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita el llevar la cuenta de qué cambios se han trasladado. Esto se resuelve siendo cuidadoso con los mensajes de commit. Esta carencia será corregida en la próxima versión.

1.5.3.3 Clientes

Existen varias interfaces a Subversion, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo.

- ✓ TortoiseSVN. Provee integración con el explorador de Windows. Es la interfaz más popular en este sistema operativo.
- ✓ Subclipse. "Plugin" que integra Subversion al entorno Eclipse.
- ✓ Subversive. "Plugin" alternativo para Eclipse
- ✓ Cervisia. Programa para interacción para Linux, combinada con Quanta Plus puede llegar a ser muy eficaz.
- ✓ ViewVC. Interfaz web, que también trabaja delante de CVS.
- ✓ Para Mac pueden emplearse los interfaces SvnX, RapidSVN y Zigversion
- ✓ RapidSVN. También corre en Linux.
- ✓ KDESvn. Provee integración en el escritorio KDE, muy parecido en apariencia, funcionamiento y características a TortoiseSVN.
- ✓ EasyEclipse es un paquete basado en eclipse es una plataforma de desarrollo, con algunos plugins de código abierto.
- ✓ Sventon. Interfaz web
- ✓ Versions. Interfaz de escritorio para Mac OS X

1.5.3.4 Uso y Reconocimiento

Subversion es muy conocido en la comunidad de software libre y se utiliza en muchos proyectos, incluyendo la fundación del software de Apache, KDE, GNOME, Free Pascal, FreeBSD, GCC, Paython, Django, Ruby, Mono, Source Forge.net, ExtJS y Tigris.org. El servicio Google Code también proporciona almacenamiento Subversión para sus proyectos de software libre. Los sistemas de BounteSorce lo utilizan exclusivamente. Codeplex ofrece a acceso tanto para Subversión como para otros tipos de clientes.

Subversión también está siendo adoptando en el mundo corporativo. En un informe 2007 de Forrester Research, reconocía a Subversion como el único líder en la categoría de sistema de control de versiones.

1.5.3.5 Funcionamiento

SVN está compuesto por una arquitectura cliente-servidor en la que los servidores contienen una copia de la información a compartir mientras que los usuarios usan el cliente para conectarse al servidor y descargar la información. Cuando un usuario realiza cambios en ficheros, lo sube al servidor para que otros puedan verlo también. El trabajo con el servidor puede hacerse mediante comandos. Además de los básicos (check out, add, commit, update) cuenta con algunos que brindan nuevas funcionalidades con respecto al CVS (copy, move, merge, reolve, medir, propget, proplit, propedit, revert, switch).

1.6 Herramientas para la gestión de proyecto

Como resultado de una investigación previa a este trabajo de diploma, donde se realizaron consultas a varios líderes de proyecto de la facultad 10, quedó establecida **Gforge** como única herramienta para la gestión de proyectos dentro de la facultad. A continuación se ofrecen algunas de las características que se tuvieron en cuenta para esta selección.

1.6.1 Gforge

“GForge fue desarrollado por la comunidad de software libre como un ambiente en el cual se almacenan proyectos en una manera en la que el código, la documentación, los archivos, son accesibles públicamente a todo el que desee verlos, y los miembros del público puedan contribuir con opiniones, detección de errores, ideas y sugerencias, además de ayudar a desarrollar el código, módulos, documentación y recursos para el software. Este provee de un completo sistema de desarrollo de software, incluido un generador de versiones, un sitio Web por proyecto y herramientas para la comunicación entre los miembros de un equipo de desarrollo. Sus herramientas permiten además a los miembros de un equipo de desarrollo una mejor organización del trabajo, y crear un conocimiento base para futuros proyectos. Es precisamente una herramienta muy poderosa para el desarrollo colaborativo de la comunidad del software.

“GForge es una bifurcación del código de 2.61 SourceForge, que estaba solamente disponible mediante CVS anónimo del VA Software Corporation. El proyecto de GForge fue formado y es mantenido por Tim Perdue, el autor de mucho del código original Web de SourceForge.”(15)

Una de las más importantes ventajas que ofrece Gforge es la posibilidad del desarrollo colaborativo del software, y aunque en estos momentos en la facultad se posee una discreta documentación, esta puede ser una de las potencialidades más explotadas del Gforge.

1.6.1.1 Las diferentes herramientas que Gforge brinda.

Gforge puede proveer de manera centralizada muchas utilidades y herramientas para la administración de proyectos. Estas herramientas son:

- ✓ Hospedaje Virtual para Proyectos.
- ✓ Foros de discusión.
- ✓ Seguimiento de errores.
- ✓ Solicitudes de soporte.
- ✓ Solicitudes de Funcionalidades nuevas.
- ✓ Seguimientos de registros.
- ✓ Listas de correo.
- ✓ Administración de Tareas.
- ✓ Administración de documentos.
- ✓ Encuestas para usuarios y administradores.
- ✓ Anuncios y noticias.
- ✓ Administración de Versiones de Ficheros.
- ✓ Repositorio de ficheros.
- ✓ Repositorio de código fuente.
- ✓ Mensajería Instantánea.
- ✓ Interfaz Web para subversion.
- ✓ Estadísticas de uso.

1.7 Tendencias actuales de la GCS

1.7.1 Procyon

Procyon es una empresa que se dedica a la elaboración de software para teléfonos celulares, centrándose en tres áreas fundamentales:

1. Software que se ejecuta dentro de dispositivos móviles, como juegos.
2. Producción de otros tipos de contenidos como imágenes, animaciones, sonidos, temas (skins) para teléfonos. Estos contenidos también necesitan adaptaciones para abarcar la mayor cantidad de modelos, contando también con software con ese propósito.
3. Software usado por operadores de telefonía celular o "agregadores" de contenidos.

En dicha empresa se utiliza la gestión de configuración del software como vía de solución para el control de versiones usando para esto la potente herramienta SVN y el trac. Utilizan además el lenguaje Java con el framework Enterprise Java Beans para el desarrollo del software, aunque cuentan también con algunos proyectos que utilizan el Visual Studio.

Desde sus mismos inicios esta institución decidió usar el SVN debido a las facilidades que esta herramienta posee y además por reconocer y entender las limitaciones de CVS.

Posee un sistema de build. "Los build son una versión funcional del sistema o una parte del sistema que presenta un subconjunto de las funcionalidades del producto final. Comprende uno o más elementos de implementación (muchas veces ejecutable), que a su vez son construidos por otros elementos, usualmente por un proceso de compilación conectado al código fuente.

El propósito del build, construido por otros elementos en la implementación, es entregar un subconjunto probable de funciones del tiempo de ejecución y funcionalidades del sistema. RUP sugiere que durante una iteración se vaya haciendo una secuencia de builds adicionando funcionalidades en cada uno, así

como los elementos de los subsistemas de implementación son adicionados o perfeccionados. Un build tiene descripción, subsistemas de implementación, elementos y referencias del plan de integración.

Los builds son obviamente obligatorios, sin embargo los tipos de builds que produce un proyecto pueden cambiar durante el ciclo de vida. En la fase de inicio, se pueden producir prototipos como una forma para entender mejor el problema y comunicarse con el cliente, en la elaboración, para producir una arquitectura estable y en construcción para adicionar funcionalidades. En la fase de transición el enfoque cambia a asegurarse de que el SW cumpla con la calidad de entrega. “(16)

“Un build no es mas que una construcción de entregables que se realiza desde las fuentes. Es un proceso automatizado y que ocurre en un servidor, el servidor de builds. En ese servidor el ambiente es controlado. Esto permite garantizar que los entregables oficiales producidos en ese servidor se archiven y sean identificados correctamente mediante firmas digitales. Asegura que todas las aseveraciones sobre la calidad de un entregable se corresponden con ese y no con otro entregable.”(17)

1.7.2 Softel

En la empresa de softel es indispensable el uso de la gestión de configuración por el abanico de posibilidades que esta brinda. Esta empresa utiliza como herramienta para el control de versiones el VSS aunque desean emigrar hacia la herramienta Clear Case por la necesidad que tienen de:

1. Integrar el ClearQuest (herramienta de gestión de cambio que se utiliza en la empresa) con una herramienta de control de versiones, que pudiera ser el VSS; pero consideran que les va a ser más útil migrar al ClearCase teniendo en cuenta sus beneficios y la política o estrategia empresarial de utilizar herramientas de forma integrada.
2. Dar continuidad a la vida del producto una vez instalado al cliente: En la empresa tienen la concepción de que cuando un software es bueno no debe morir.
3. No hacer al software dependiente a factores humanos: Actualmente existe mucha inestabilidad laboral y en ocasiones no se cuenta con las personas encargadas de algunas tareas y al prescindir de dichas personas el software puede quedar “en el aire”, sin embargo si se cuenta con una documentación concreta, actualizada y disponible del producto, los problemas se minimizan.

El trabajo con la herramienta la consideran indispensable, con el VSS gestionan todo lo que tiene que ver con código, documentación e incluso informaciones adicionales que son de su interés almacenar en el repositorio.

Se tiene entendido después de realizar varias investigaciones que las personas que trabajan con el VSS tienen pleno control sobre la herramienta y aunque existen algunas quejas de esta, los que ya poseen una gran experiencia tienen pleno conocimiento y los transmiten con el objetivo de lograr productividad.

A modo de conclusión se puede decir que:

Las empresas mencionadas anteriormente tienen diferentes formas de llevar a cabo la gestión de configuración del software, por ejemplo:

Procyon aplica la gestión de configuración de software con conocimientos empíricos, técnicos, o sea, no tienen definidos procedimientos, no cuentan con un Comité de Control de Cambios para gestionar los cambios (este comité se encarga de analizar los cambios y si estos se aceptan o no), no tienen establecidos ECS, sin embargo aplican cosas novedosas que hacen más sencillo el trabajo, tal es el caso del sistema Build, que como se explicó anteriormente, es un software creado por ellos que automatiza muchas de las actividades de la GCS.

Softel por su parte aplica la gestión de configuración de software de una forma teórica, o sea, cuenta con procedimientos bien definidos, poseen un equipo estructurado donde cada persona es responsable de las tareas que se le asignan, controlan los cambios mediante la herramienta VSS.

1.8 Estrategias de Gestión de Configuración en la UCI

1.8.1 Estrategia para la Gestión de Configuración de Software en el proyecto Registros y Notarías

“El presente proceso fue desarrollado por la UCI, plantea una Estrategia de Referencia de Gestión de Configuración (ErGCS), para ser aplicada en el Proyecto de Modernización de los Registros y Notarías de la República Bolivariana de Venezuela (R&N). En la estrategia se siguen algunos de los principios de RUP, por ser la metodología que actualmente guía el proceso de desarrollo del proyecto, además se tiene

en cuenta lo que sobre el tema plantea el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). En la ErGCS se proponen actividades que deben estar presentes en el proceso de GCS del proyecto. Además se muestran cada uno de los procedimientos a seguir en cada una de dichas actividades, así como la asignación de un rol responsable de ejecutarla. También se lleva a cabo la selección de herramientas que soporten un conjunto de actividades y procedimientos presentes en dicha propuesta.

La herramienta que ellos proponen para el apoyo y la automatización de las actividades de la GCS, específicamente para la actividad de Control de Versiones, la instalación de Subversion sobre un servidor Apache, para de esta forma poder disfrutar de la robustez, autenticación HTTP, compresión de datos, cifrado, etc. que ofrece Apache. Proponen además el uso combinado del Subversión con la herramienta TRAC 8, esta última permite mediante su sistema de tickets 9 la gestión de las tareas en el proyecto, ya sean errores a arreglar, mejoras pendientes, entre otras. Además sus capacidades de Wiki permiten crear dichos reportes de tickets totalmente personalizados, cambiar la apariencia del sitio o agregar páginas informativas del proyecto. Proponiendo además, la gestión y uso de una Red Privada Virtual (VPN) para la utilización de un repositorio único mediante una conexión segura entre los dos países. Una vez que ellos instalan el servidor de Subversion crean un repositorio sobre el cual estructuran el expediente del proyecto. Formando parte de esta estrategia se realiza una propuesta de cómo estructurar dicho expediente en áreas de tener una mejor organización. Para ello se parte de la propuesta del departamento de calidad en la universidad pero adaptándolo a las características y necesidades del proyecto. El repositorio puede estar estructurado mediante una carpeta con el identificador del proyecto y dentro un subsistema de carpetas principales: Desarrollo Técnico. Sistema de Calidad. Seguridad Informática. Gestión de Proyecto. Gestión de Configuración. Línea Base Proyecto

Esta estrategia realiza las actividades de gestión de configuración de Identificación de la Configuración del Software, Gestión de Cambio de la Configuración, Control de Versiones de la Configuración del Software, Generación de Informes de Estado, Control Interno al proceso de GCS. Estas actividades fueron seleccionadas, primeramente, por la importancia que tiene su cumplimiento en el proceso de GCS. En la estrategia la ErGCS brinda una serie de técnicas y procedimientos sólidos encaminados a proporcionar mejoras en el proceso de GCS “.(18)

1.8.2 Estrategia para la Gestión de Configuración en el Proyecto Sistema de Gestión Penitenciaria

“Se elabora una estrategia para la Gestión de Configuración en el proyecto Sistema de Gestión Penitenciaria (SIGEP), con el propósito de establecer un control sobre los elementos que se generan durante el desarrollo del software, y mantener así la integridad del producto a entregar. Se proponen métricas para medir atributos de la configuración, que permiten observar el nivel de calidad alcanzado durante la ejecución de los procesos de Gestión de Configuración, con el propósito de establecer un control sobre la información referente a esfuerzos, cumplimiento, eficiencia y rendimiento entre otros parámetros. Para la descripción de la estrategia definieron procesos y una descripción de lo que ocurre en cada uno de ellos. En SIGEP, la metodología guía para el desarrollo del producto es RUP, la cual ha sido adaptada a las condiciones específicas del proyecto. Teniendo en cuenta las actividades que se desarrollan en cada uno de los flujos de trabajo que RUP define, se puede decir que esta estrategia se aplica en cada una de las fases, en cada iteración y en cada momento durante todo el ciclo de vida del producto software. Hacen establecimiento de seis procesos fundamentales a desarrollar dentro de la Gestión de Configuración del Software. En la ejecución de estos procesos se identifican diversas actividades, donde intervienen distintos roles que deberán responder por los artefactos que se generan. Los procesos definidos son:

1. Identificación de la configuración del software
2. Planificación de la gestión de configuración
3. Control de cambios en la configuración
4. Control de versiones en la configuración
5. Generación de informes de estado de la configuración
6. Auditorías a la configuración.

En estas actividades que se llevan a cabo se generan artefactos como la plantilla del plan de gestión de configuración y plantilla de control de cambios, las cuales son de gran importancia en este proceso para un buen funcionamiento.

La estructura del repositorio se desarrolla almacenando el trabajo de los desarrolladores. Para la estructura de la línea principal del desarrollo en el repositorio se basan en la que propone la comunidad de autores de Subversion o sea a través del uso de: branches, samples, tags, trunk para una buena organización en el repositorio, cada miembro del proyecto tendrá acceso restringido según sus responsabilidades dentro del proyecto.

Para una buena integración de eclipse con la herramienta de control de versiones se ha seleccionado Subversion y específicamente el cliente Tortoise SVN, que provee una buena integración con el explorador de Windows. Además Subversion por medio de subclipse puede integrarse a eclipse. Se establecen las políticas que se deben seguir para realizar una buena Gestión de Configuración. Se explica detalladamente el flujo de trabajo que se realiza en cada proceso especificando los roles que participan, las actividades que estos realizan y los artefactos que se generan, esta estrategia para una mayor calidad hace uso de las métricas para medir atributos como:

1. Esfuerzo individual de los miembros del equipo.
2. Estado de los elementos de configuración.
3. Elementos de configuración en cada versión del proyecto.
4. Versiones libres de defectos en una etapa.
5. Estado de las solicitudes de cambio.
6. Eficiencia en la resolución de las peticiones de cambios.
7. Rendimiento personal para implementar cambios.
8. Rendimiento del equipo para implementar cambios.
9. Cumplimiento personal para implementar cambios.
10. Cumplimiento por equipo para implementar cambios. "(19)

1.8.3 Comparación entre las estrategias

La Estrategia para la Gestión de Configuración del Software propuesta por el proyecto Registros y Notarías explica bastante detallado el proceso de Gestión de Configuración pero no llega a realizar todas las actividades que se llevan a cabo en los procesos, realizan una unión de los procesos: Planificación de la Gestión de Configuración e Identificación de la Configuración, eliminando muchas actividades

importantes de estos procesos, no se trata claramente el control de versiones que a pesar de que tiene mucha relación con el control de cambios no deja de ser importante. Se realiza una buena identificación de los elementos de configuración que es una de las actividades necesarias y preliminares dentro del proceso de GCS. Se lleva un buen control de los cambios. La herramienta que usan para el control de versiones es Subversion que hoy en día es una de las más usadas en la producción de software. Pero no explican muy claramente el trabajo con el repositorio. Utiliza estándares de calidad.

La estrategia propuesta para el proyecto SIGEP reúne las características más importantes de los procedimientos a seguir en la estrategia y puede ser adaptado a cualquier proyecto de software; es mucho más explícito a cómo aplicarse que las estrategias anteriores, presenta una propuesta de roles a asignar a cada una de las actividades organizándolas estructuralmente, se mide la calidad aplicando métricas, es la más completa de todas y reúne las características más importantes de los modelos anteriores, se ha probado en el proyecto haciendo que el mismo tenga una buena calidad y un buen desarrollo, toda esta información la hace propicia a la implantación en el proyecto de GDA.

1.9 Conclusiones

Luego de realizada la investigación sobre el estado del arte de esta disciplina, se puede concluir que no existe mucha diversidad en los criterios y definiciones planteados por diferentes autores. También es importante mencionar que uno de los procesos fundamentales de la gestión de configuración es la gestión de cambios y que ella determina en gran medida el éxito de un proceso de desarrollo de software. Por otra parte, atendiendo al estudio realizado a las empresas **Procyon** y **Softel** se puede alegar que a pesar de que no utilizan una estrategia propiamente, si utilizan otros métodos efectivos para controlar la gestión de configuración. Mientras que el estudio realizado a las estrategias seguidas por los proyectos RN y SIGEP arroja que estas estrategias no son lo suficientemente eficaces.

Capítulo 2

Definición de la estrategia de GCC para el grupo de proyecto GDA.

2.1 Introducción

En este capítulo inicialmente se realiza una propuesta de organización del equipo de trabajo, definiendo las funciones correspondientes a cada miembro del proyecto.

2.2 Situación de la GCS del proyecto GDA

La situación de grupo de proyecto será evaluada a través de un análisis exhaustivo de los problemas que presenta la GCS actual, teniendo en cuenta una serie de indicadores para la obtención de la calidad que son de interés su monitoreo. Estos son:

2.2.1 Uso de la herramienta para el control de versiones (SVN)

En el proyecto el uso de esta herramienta es bastante discreto, son muy pocos los miembros del proyecto que saben trabajar correctamente con la herramienta lo que tiene como consecuencia que la información no esté del todo actualizada.

2.2.2 Satisfacción del cliente

La satisfacción del cliente no ha sido la esperada por parte de los miembros del proyecto, esto se debe al poco conocimiento de la herramienta para el control de versiones, la falta de plantillas y de un formato determinado para ellas, incumplimiento con los requisitos y de las fechas de entrega pactadas.

2.2.3 Documentación y control de cambios de los ECS

La documentación de los cambios de los ECS ha presentado serias dificultades, debido a la inexistencia de los procedimientos formales que deben llevarse a cabo ante la realización de cualquier cambio, además del desconocimiento de la solicitud de cambio que se debe completar y enviar para solicitar así un cambio de un determinado ECS.

Los cambios son de suma importancia, pero también lo es el conocimiento de estos por parte de las personas indicadas, ya sean, personal de desarrollo, jefes y clientes. Para ello deben existir en el proyecto los informes de estado, documentación de los cambios, así como los seguimientos de los mismos y auditorías de comprobación.

2.2.4 Cantidad de elementos de configuración en desarrollo

En estos momentos se desconoce la cifra exacta ya que no se realiza ningún proceso para contabilizar dichos elementos.

2.2.5 Cantidad de elementos de configuración en revisión

Se desconoce dicha información pues no se realiza ningún proceso de revisión o análisis de los mismos.

2.2.6 Cantidad de elementos de configuración terminados

No existe un conocimiento de la cantidad de ECS que han sido terminados, debido a la falta de comunicación entre los miembros del proyecto.

2.2.7 Control de no conformidades o solicitudes de cambios realizadas por los clientes

No existe un mecanismo formal para controlar las no conformidades o solicitudes de cambios realizadas por los clientes por lo que se incurre en problemas como:

Software que no cubre las expectativas o que no cumple todas las funcionalidades establecidas, incomodidad del cliente al no ser atendida una no conformidad.

2.2.8 Cantidad de solicitudes de cambios en espera de ser revisadas

No existen las solicitudes de cambios pues no se han contabilizado los cambios realizados.

2.2.9 Cantidad de solicitudes de cambios en revisión

Nula.

2.2.10 Cantidad de solicitudes de cambios cerradas

Producto de la ausencia de un procedimiento para la solicitud de cambios, no se realiza el control de la cantidad de solicitudes de cambios cerradas.

2.2.11 Evaluación de los Procedimientos llevados a cabo en el proyecto

Actualmente no se tienen concebidos procesos para la evaluación de procedimientos.

2.3 Organización del equipo de trabajo

Para la organización del equipo de trabajo se ha creado una estructura con el objetivo de asignar las actividades de la GCC según los roles, de manera que queden bien establecidas las responsabilidades de cada integrante dentro del equipo:

Roles:

- ✓ Líder de factoría
- ✓ Arquitecto de Software Principal
- ✓ Analista de Software
- ✓ Administrador de Configuración
- ✓ Auditor
- ✓ Gestor de Configuración y cambios
- ✓ Interesado en el cambio
- ✓ Comité de Control de Cambios (CCC)

El Líder de factoría y el Analista de Software serán los encargados de guardar y mantener controlada la información del producto en el SVN, de manera tal que se mejore así el uso de la herramienta para el control de versiones y evitar males mayores como pérdida de información. Los demás miembros del equipo trabajarán en las actividades y tareas correspondientes a su rol.

2.3.1 Actividades que se realizan en cada rol

El **Líder de factoría** se encargará de:

- ✓ Emitir la aprobación del Cambio
- ✓ Definir una metodología de desarrollo.
- ✓ Aprobar las tecnologías a usar en el desarrollo del proyecto.
- ✓ Definir las bibliotecas de software.
- ✓ Coordinar y organizar todas las tareas que se asignan a todos los miembros del equipo de desarrollo.

El **Analista del software** se encargará de:

- ✓ Definir los artefactos que se obtendrán como resultado del análisis y la metodología que se sigue para obtenerlos.
- ✓ Supervisar y controlar el cumplimiento de la metodología para el análisis.
- ✓ Definir los sistemas, subsistemas y módulos en que se organiza la solución de software.

El analista se encargará además del desarrollo de los artefactos del producto (ECS) y su documentación, es decir, una vez que se establezcan los ECS, los analistas serán los que se responsabilicen con la realización de estos.

El **Arquitecto de Software Principal** se encargará de:

- ✓ Recomendar una metodología de desarrollo.
- ✓ Establecer la arquitectura del módulo, priorizar los casos de uso más críticos así como diseñar todo lo que se va a programar.

El **Administrador de Configuración** se encargará de:

- ✓ Mantener la configuración estable del producto.
- ✓ Analizar el impacto de los cambios propuestos por los clientes con los analistas.
- ✓ Llevar el control de cambios de las versiones estables.
- ✓ Integra el comité de control de cambios para la parte que tenga que ver con el equipo de desarrollo.

El **Auditor** se encargará de:

- ✓ Auditar la configuración.

El **Gestor de Configuración y Cambios** se encargará de:

- ✓ Velar por el cumplimiento de lo plasmado en el plan de gestión de configuración.
- ✓ Identificar y actualizar las versiones de cada ECS.
- ✓ Evaluar Trascendencia de Solicitud de Cambio.
- ✓ Registrar la Solicitud de Cambio
- ✓ Establecer tipos de auditorías.

El **Interesado en el cambio** realizará las actividades:

- ✓ Emitir solicitud de cambio
- ✓ Revisar estados de las solicitudes de cambio
- ✓ Actualizar la solicitud de cambio

El **Comité de Control de Cambio** realizará las actividades:

- ✓ Evaluar el impacto del cambio
- ✓ Emitir información final sobre la aprobación o negación del cambio.

2.4 Procesos de Gestión de Configuración en el proyecto GDA

En esta sección se definirán los 5 procesos fundamentales a desarrollar dentro de la estrategia, en la descripción de estos procesos se detallan las actividades que deben realizarse por cada rol.

Los procesos definidos son:

1. Definición y Establecimiento de la Configuración del Software.
2. Control de Versiones en la Configuración.
3. Control de Cambios en la Configuración.
4. Auditorías a la Configuración.
5. Generación de Informes de Estado de la Configuración.

Existen una serie de preguntas que conducen a la definición de los cinco procesos de la estrategia.

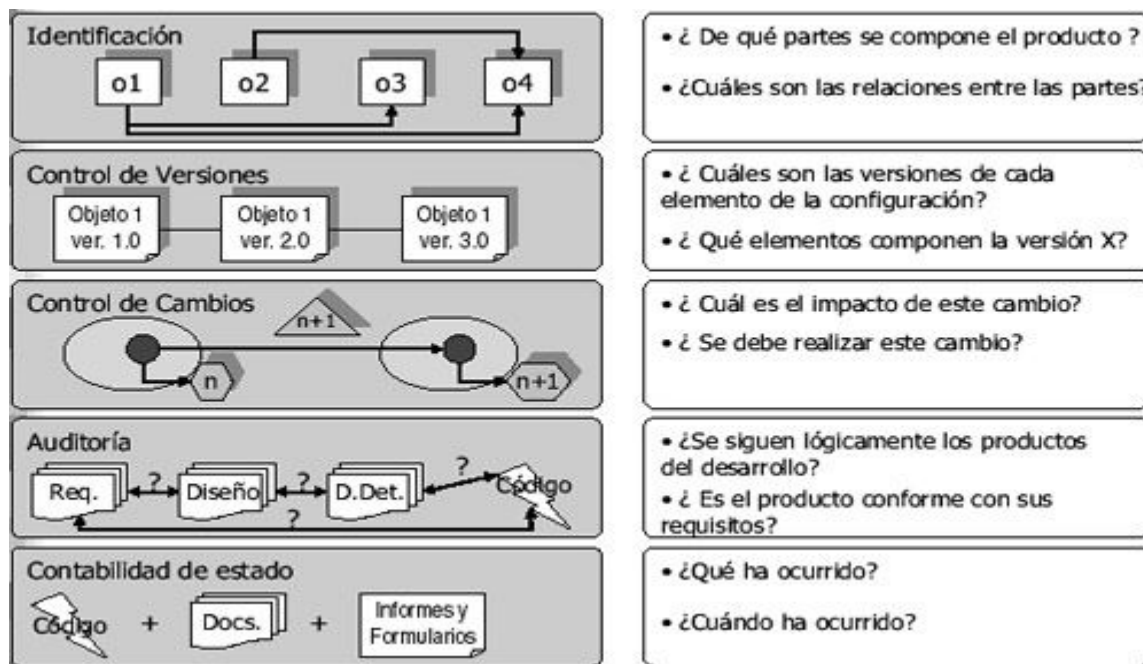


Figura 4 (12)

2.4.1 Definición y establecimiento de la configuración del software

La definición y establecimiento de la gestión de configuración del software tiene como objetivo definir la estructura del producto, así como la de sus componentes. Esta tarea es muy importante para la gestión de configuración, ya que consiste en identificar y asignar nombres a todos los elementos que forman parte del producto, que no son otros que los ECS. Este proceso será llevado a cabo por el rol de Administrador de Configuración.

El control y la gestión de elementos de configuración del software requerirán de:

1. Nombrar cada uno de los elementos por separado
2. Organizarlo mediante un enfoque orientado a objetos

Es posible identificar dos tipos de objetos:

1. Básicos: Unidad de información creada durante: análisis, diseño, código o pruebas. Por ejemplo: una sección de una especificación de requisitos, parte de un modelo de diseño, código fuente para un componente, o un conjunto de casos de prueba.
2. Agregados: Colección de objetos básicos y otros objetos agregados. Es un mecanismo para representar una versión completa de una configuración de software.

“Cada elemento deberá poseer una serie de características distintivas que lo identifican de manera exclusiva:

1. Identificador (número, letra, ambos. No ambiguo)
2. Nombre (descriptivo)
3. Tipo (documento, código, producto de terceros, etc.)
4. Localización
5. Fecha
6. Versión (mayor, menor, revisión)
7. Estado (Ej. Para un documento En elaboración, finalizado, revisado, aceptado)
8. Proyecto/producto”(3)

Es importante saber definir el número de ECS que se van a manejar pues si son demasiados, la documentación, especificación y descripción de las funciones puede llegar a ser inmanejable; si son pocos, el software pudiese escasear o carecer de información sustentable; es decir que se deben escoger los ECS necesarios, suficientes y más importantes.

2.4.1.2 Elementos de un sistema de gestión de la configuración

Susan Dart identifica cuatro elementos que deben estar presentes en el momento de desarrollar un sistema de gestión de la configuración eficaz:

1. **“Elementos de componentes:** conjunto de herramientas acopladas dentro de un sistema de gestión de archivos (ej.: Base de datos), permiten el acceso y la gestión de cada elemento de configuración del software
2. **Elementos de proceso:** serie de procedimiento y tareas que definen un enfoque eficaz con el cual gestionar el cambio
3. **Elementos de construcción:** conjunto de herramientas que automatizan la construcción del software al asegurar que se ha ensamblado un conjunto adecuado de componentes validados
4. **Elementos humanos:** la implementación de una GCS eficaz requiere que el equipo de software utilice un conjunto de herramientas y características de procesos”(20)

¿Cómo identificar un ECS?

La respuesta a esta interrogante se verá reflejada mediante un proceso de identificación que se llevó a cabo, el cual primeramente se basa en establecer responsables que van a ejecutar la actividad, posteriormente se establecen los estados de las versiones de los ECS y se definen las actividades que rigen este documento tales como:

Identificar los ECS y documentarlos, adicionar al artefacto ECS con la siguiente información:

- A. Identificador del ECS
- B. Descripción del ECS
- C. Autor(es) del ECS
- D. Listado de elementos dependientes del ECS
- E. Listado de los módulos que lo usan
- F. Tipo de ECS
- G. Localizaciones

Los ECS se podrán definir según los siguientes criterios:

- ✓ “Utilización múltiple: Número de elementos de su mismo nivel o niveles superiores que lo utilizan.
- ✓ Criticidad: Gravedad del impacto de un fallo en dicho componente.
- ✓ Número de personas implicadas en su mantenimiento.

- ✓ Complejidad de su interfaz: Las interfaces de un EC con otros deberían ser simples. Hay que minimizar el acoplamiento entre ECS.
- ✓ Singularidad del componente con respecto al resto.
- ✓ Reutilización: Si el componente se va a diseñar especialmente para ser reutilizado.

Si se trata de elementos reutilizados.

- ✓ Tipo de tecnología: Si el componente incorpora nuevas tecnologías no utilizadas en otros componentes.

Una vez que se aprueban los ECS estos pasarán automáticamente a ser controlados por la GCS, luego de documentar las relaciones existentes entre estos se almacenarán en el repositorio la versión de los ECS definidos, se saca a la luz el artefacto de ECS incluyendo las relaciones implantadas y se realizará la distribución del mismo, por último se considera que a la hora de seleccionar los ECS es sumamente importante analizarlos y organizarlos por categorías con el objetivo de minimizar el impacto de los cambios.

2.4.1.3 Relaciones entre los ECS

Se puede deducir que dentro del proceso de desarrollo del software es indispensable establecer y mantener bien definidas las relaciones entre los elementos de configuración de software, dentro de las cuales se pueden establecer del tipo:

Composición: Los elementos de configuración componentes, son dependientes de la existencia de los elementos de configuración compuestos, se establece una relación fuerte entre ellos donde la existencia o no de los elementos componentes depende en gran medida del elemento de configuración compuesto, éste último tiene la responsabilidad de la creación y/o destrucción de los elementos componentes.

Dependencia: Relación semántica que indica que un cambio en un elemento afecta a la semántica de otro elemento.

Sucesión: Relación que se establece dada las transiciones de los elementos de configuración de una versión a otro producto de los cambios que ocurren en el proyecto. Para ello es útil auxiliarse de un grafo de evolución que describa la historia de cambios de un elemento de configuración del software y su evolución.

Funcional: Se establecen cuando varios ECS responden o comparten la misma funcionalidad.

Derivación: A partir de qué se ha originado algo. Por ejemplo, el código objeto del código fuente, o una determinada traza de ejecución de un determinado caso de prueba con un determinado programa ejecutable.

Variante: Variación sobre un determinado elemento, con la misma funcionalidad, pero que, por ejemplo, funciona más rápido.

Equivalencia: Por ejemplo, cuando un determinado ECS que es un programa está almacenado en tres lugares diferentes (un disco maestro, una copia de seguridad en cinta y el diskette del programador), pero todas las copias corresponden al mismo programa.

Estas relaciones permitirán identificar los elementos de configuración del software que se verán afectados a la hora de implementar algún cambio sobre un elemento cualquiera, así como llevar un seguimiento de cada elemento dentro del proyecto.(21)

2.4.1.4 Procedimiento para establecer las líneas base

Para establecer la línea base del proyecto de GDA se hace necesario llevar a cabo un procedimiento el cual consta de los siguientes pasos:

- ✓ Definir los responsables encargados de llevar a cabo la actividad.
- ✓ Definir la fecha de elaboración.
- ✓ Definir una estrategia a seguir para implantar la línea base.
- ✓ Constatar la culminación de la iteración.

Actividades a desarrollar

- A. Establecer estado de las versiones de los ECS.
- B. Identificar ECS involucrados en el proceso.
- C. Aprobar ECS para la línea base.
- D. Aclarar iteración a la que pertenece la línea base.
- E. Verificar y establecer fase de realización.
- F. Elaborar la línea base.

Termina el procedimiento con la distribución del informe de línea base a todos los miembros del equipo.

2.4.1.5 Actividad de establecer las líneas bases

Se establece en la primera iteración como línea base, la versión 1.0 del producto en la fase de ambientación que incluye los siguientes ECS: Informe de la investigación preliminar, registro de la Investigación, cuadro de clasificación, modelos de flujo de procesos, especificación de requisitos, tabla de plazos de transferencia, tabla de acceso y seguridad, informe de evaluación de sistemas existentes, e informe de estrategias.

Ya para la segunda iteración se establece como línea base la versión 1.1 del producto, en las fases de Implementación/Pruebas, en la cual se incluyen los siguientes ECS: Diagrama de entidades de la empresa, diagrama de modelos de contenidos, diagrama de flujos de proceso, diseño del cuadro de clasificación, diseño de la tabla de plazos de transferencias, diseño de la tabla de acceso y seguridad, código fuente, manual de usuario, documento de integración, documento escenario(s) de prueba unitaria, archivo de bugs, documento plan de prueba, documento de gestión de bugs.

2.4.1.6 Actividad de verificación de las líneas bases

Los encargados de constatar que las líneas bases establecidas funcionen correctamente será el arquitecto de software principal conjuntamente con el gestor de configuración y cambios, para verificar esto, primeramente se debe tener claro que las líneas bases deben permitir identificar la parte funcional del sistema que ha sido revisado correctamente y pasado por una serie de pruebas que descartan cualquier fallo o defecto.

Así, una vez establecidas las líneas bases, se podrá chequear el cumplimiento de las planificaciones previstas en cuanto a tareas y artefacto a desarrollar se refiere. Estas serán las líneas bases que se usarán para continuar con el desarrollo del producto.

Es importante establecer una línea base para cada equipo de desarrollo al concluir cada fase de desarrollo y verificar así el cumplimiento de los objetivos de esta. De manera tal que a medida que transcurran las fases de desarrollo para un módulo en específico se defina una línea base que posea una mayor cantidad de elementos y funcionalidades.

Una vez finalizado el desarrollo de los módulos pertenecientes a un misma captura de requerimientos, y realizarse integraciones de estos, se establecerá una línea base que recoja la funcionalidad de todos en conjunto. Asimismo de manera unívoca se realizará la integración final del producto a entregar.

2.4.1.7 Actividad de definir bibliotecas de software

En el proyecto de GDA resulta imprescindible el uso de las bibliotecas de software para establecer un control sobre los elementos de configuración que se desarrollan a diario en este proyecto, así como la documentación de los artefactos que se generan. A su vez el uso apropiado de la biblioteca del software posibilita el almacenamiento de todo el trabajo de los desarrolladores a lo largo del proceso de elaboración del software además de realizar salvadas de proyecto.

Hoy en día las bibliotecas del software constituyen el punto culminante del almacenamiento de información ya que permite a un equipo de trabajo gestionar los cambios de manera eficiente.

2.4.2 Control de Versiones

“Este proceso combina procedimientos y herramientas para gestionar diferentes versiones de objetos de configuración que se crean durante el proceso del software.

Un sistema de control de versiones implementa cuatro grandes capacidades:

1. **Base de datos del proyecto:** guarda los objetos de configuración relevantes.
2. **Capacidad de gestión de la versión:** almacena todas las versiones de un objeto de configuración.
3. **Facilidad de hechura** que permita al ingeniero de software recopilar todos los objetos de configuración relevantes y construir una versión específica del software.
4. **Seguimiento de conflictos** (seguimiento de bugs): permiten al equipo registrar y hacer seguimiento del estado de todos los conflictos destacados – asociados con cada objeto de configuración.”(12)

2.4.2.1 Actividad de identificar versiones de elementos de configuración

Esta actividad se llevará a cabo por el gestor de configuración y cambios del proyecto. La necesidad de establecer una nueva versión está dada por la ocurrencia de un gran cambio y puede ser identificada al realizar las auditorías correspondientes a la configuración del software.

Se pretende con esta actividad identificar las versiones por las que suelen pasar los elementos de configuración con el objetivo de ser recuperados en cualquier momento.

2.4.2.2 Actividad de controlar seguimiento de versiones

A manera de controlar el seguimiento de las versiones que se generan gradualmente en nuestro proyecto, se realizará por cada ECS, en caso de ser necesario, un grafo de evolución, el cual permitirá entender con certeza las diferentes versiones o variantes por las que se ha encaminado el ECS. Esta actividad es el punto cúspide del rol de gestor de configuración y cambios. Una vez que surge una nueva versión se toman datos como:

- ✓ Fecha de creación
- ✓ Responsable
- ✓ Características que difieren con la versión anterior

Las versiones estarán definidas e identificadas por números con una estructura similar a: <<X.Y.Z...>>. Lo cual nos da una medida de cuan significativo podrá ser el cambio producido sobre el ECS.

2.4.3 Control de Cambios

Este proceso consistirá en evaluar y registrar todos los cambios que se hacen durante la configuración del software. James Bach reconoce que se enfrenta a un acto de equilibrio cuando dice que: “El control del cambio es vital. Pero las fuerzas que lo hacen necesario también lo tornan irritante. Nos preocupamos por los cambios porque una pequeña perturbación en el código puede crear una gran falla en el producto. Pero también puede resolver una gran falla o permitir maravillosas nuevas capacidades. Nos preocupamos por los cambios porque un solo desarrollador solitario podría hundir el proyecto; aunque en las mentes de dichos solitarios se originan ideas brillantes, y un proceso de control del cambio gravoso podría desalentarlos efectivamente de realizar trabajo creativo”(12). Demasiado control del cambio, crea problemas; poco, crea otros problemas.

2.4.3.1 Gestor de Configuración y Cambios. Sus funciones.

El rol de gestor de configuración y cambios es un rol que está compuesto por dos subroles los cuales son: gestor de cambios y gestor de configuración. Estos roles por separados juegan un papel protagónico dentro de la GCS por ejemplo:

El gestor de cambios es el encargado de velar por el seguimiento, estado y control de todas las solicitudes de cambios emitidas por los clientes o inclusive por los mismos miembros del equipo de desarrollo.

Por su parte el gestor de configuración establece las políticas de gestión de configuración, además se encarga de definir y establecer la gestión de configuración del software, proceso incluido dentro de la estrategia de GCS para el proyecto. Además es el encargado de definir las líneas bases y de actualizar el estado de los elementos modificados, no sin antes ser probados debidamente, por último y no menos importante, será el encargado de emitir la información obtenida en este proceso y reportarla a manera de informes.

El gestor de cambios y el gestor de configuración deben estar en constante intercambio de información y responsabilidades con el fin de llevar a cabo un control estricto sobre los elementos de configuración identificados. Por todo lo anteriormente descrito se ha llegado a la conclusión de que estos roles se verán reflejados e interpretados por un solo rol, el Gestor de Configuración y Cambios.

2.4.3.2 Clasificación de estados de las solicitudes de cambio

- ✓ **Emitida:** La Solicitud de Cambio ha sido emitida pero aún no se ha dado una aprobación o denegación de la misma.
- ✓ **Duplicada:** La Solicitud de Cambio ha sido emitida anteriormente por otro interesado y se encuentra en proceso.
- ✓ **Aceptada:** El cambio solicitado es aceptado.
- ✓ **En desarrollo:** La Solicitud de Cambio fue aceptada y está siendo implementada por los desarrolladores.
- ✓ **Postpuesta:** Una vez analizada la Solicitud de Cambio puede determinarse que el cambio puede ser tratado, pero no en el momento actual, pues no es prioritario según lo planificado para la iteración.
- ✓ **Rechazada:** La Solicitud de Cambio es rechazada, porque no se encuentra dentro de las prioridades y necesidades del proyecto, o no resulta realmente necesario realizar el cambio, o es demasiado costosa su implementación.
- ✓ **Rechazada por falta de información:** El rechazo puede estar dado por falta de información de la Solicitud de Cambio, de manera que el interesado en el cambio puede volver a emitir la Solicitud de Cambio con la información necesaria.
- ✓ **Cerrada:** Es el caso en que la Solicitud de Cambio fue aprobada en algún momento, se implementó el cambio, el equipo de calidad le realizó pruebas y los resultados arrojados fueron satisfactorios.

2.4.3.3 Actividad de emitir solicitud del cambio

Esta actividad será realizada por el interesado en el cambio, rol que puede ser procedente tanto del lado del cliente como del mismo equipo de desarrollo, el cual se encargará de emitir una solicitud de cambio,

informando el motivo por el cual emite dicha solicitud, además esta solicitud constará del cambio deseado con su respectiva justificación y el ECS al que el interesado en el cambio hace referencia.

2.4.3.4 Actividad de revisar estado de la solicitud de cambio

Esta actividad será llevada a cabo por el interesado en el cambio y básicamente se iniciará cuando este reciba una solicitud de cambio emitida por el gestor de configuración y cambios, la cual le resultará de vital importancia para el posterior desempeño en el proyecto, informándosele así del seguimiento y tratamiento del cambio.

2.4.3.5 Actividad de registrar la solicitud de cambio

Esta actividad será efectuada por el gestor de configuración y cambios y consiste en registrar la solicitud de cambio en el listado de solicitudes de cambios existente, actualizar el seguimiento de solicitud e informar al líder de factoría todo lo acontecido en dicha solicitud de cambio con el fin de que este emita su aprobación para gestionar o no dicha solicitud.

Antes de que el líder de factoría emita la solicitud de cambio esta pasará por un proceso de clasificación del cual dependerá totalmente su posterior trayectoria. La clasificación de los cambios es:

Cambio informal: Se le atribuye esta denominación cuando se hace referencia a un elemento de configuración del software que aún no ha pasado a la línea base de desarrollo del proyecto. La aprobación de la realización del cambio es emitida por el CCC.

Cambio semi-formal: Se le atribuye esta denominación cuando se hace referencia a un elemento de configuración del software que ha pasado a la línea base de desarrollo del proyecto, pero el cambio a desarrollar no afecta el trabajo del resto de los equipos de desarrollo y puede implementarse a escala local sin complicaciones. La aprobación de la realización del cambio es emitida por el CCC.

Cambio formal: Se le atribuye esta denominación cuando se hace referencia a un elemento de configuración del software que ha pasado a la línea base de desarrollo del proyecto, y un cambio sobre este elemento afectaría el trabajo del resto de los equipos de desarrollo, de manera que se necesita de la

aprobación del líder de de factoría, pero antes el comité de control de cambios debe realizar una evaluación del posible impacto del cambio.

2.4.3.6 Actividad de emitir la aprobación del cambio

El CCC por su parte será el encargado de llevar a cabo esta tarea la cual consiste en el análisis de la evaluación del impacto dada por él y emite la aprobación o no de la ejecución del cambio.

2.4.3.7 Actividad de evaluar el impacto del cambio

El rol encargado en desarrollar esta actividad es el CCC el cual deberá evaluar el impacto desde su respectiva área de desarrollo, donde posteriormente se realizará una evaluación integral del impacto en el proyecto.

2.4.3.8 Tratamiento de las no conformidades

El equipo de calidad compuesto por el responsable de calidad y el asegurador de esta, juegan un papel primordial, pues es el equipo que se va a encargar de monitorear que el producto a entregar cumpla con la calidad y eficiencia requerida por el cliente. Para cumplir este objetivo el equipo de calidad se deberá llevar una serie de revisiones técnicas formales a los distintos productos que se van desarrollando en el proyecto, en caso de que estas revisiones no cumplan con estos requerimientos de calidad se generará una no conformidad por parte del cliente la cual deberá ser atendida inmediatamente. Estas no conformidades pueden ser emitidas tanto por el cliente como por parte del mismo equipo de calidad mientras transcurre el periodo de pruebas, una vez emitida, esta pasa a manos de gestor de configuración y cambios el cual se encargará de llevar el seguimiento de la misma personalmente. Cada una de estas no conformidades deber ser analizadas por el líder de factoría conjuntamente con el equipo que llevó a cabo el desarrollo de los artefactos sujetos a las mismas, para así realizar una valoración de la factibilidad de desarrollarla y realizar el cambio que la satisfaga, o rechazar la no conformidad. A partir de entonces, si es aceptada la implementación de dicha no conformidad, comienza a tratarse como un cambio a un elemento de configuración determinado que ha de llevar un seguimiento y control.

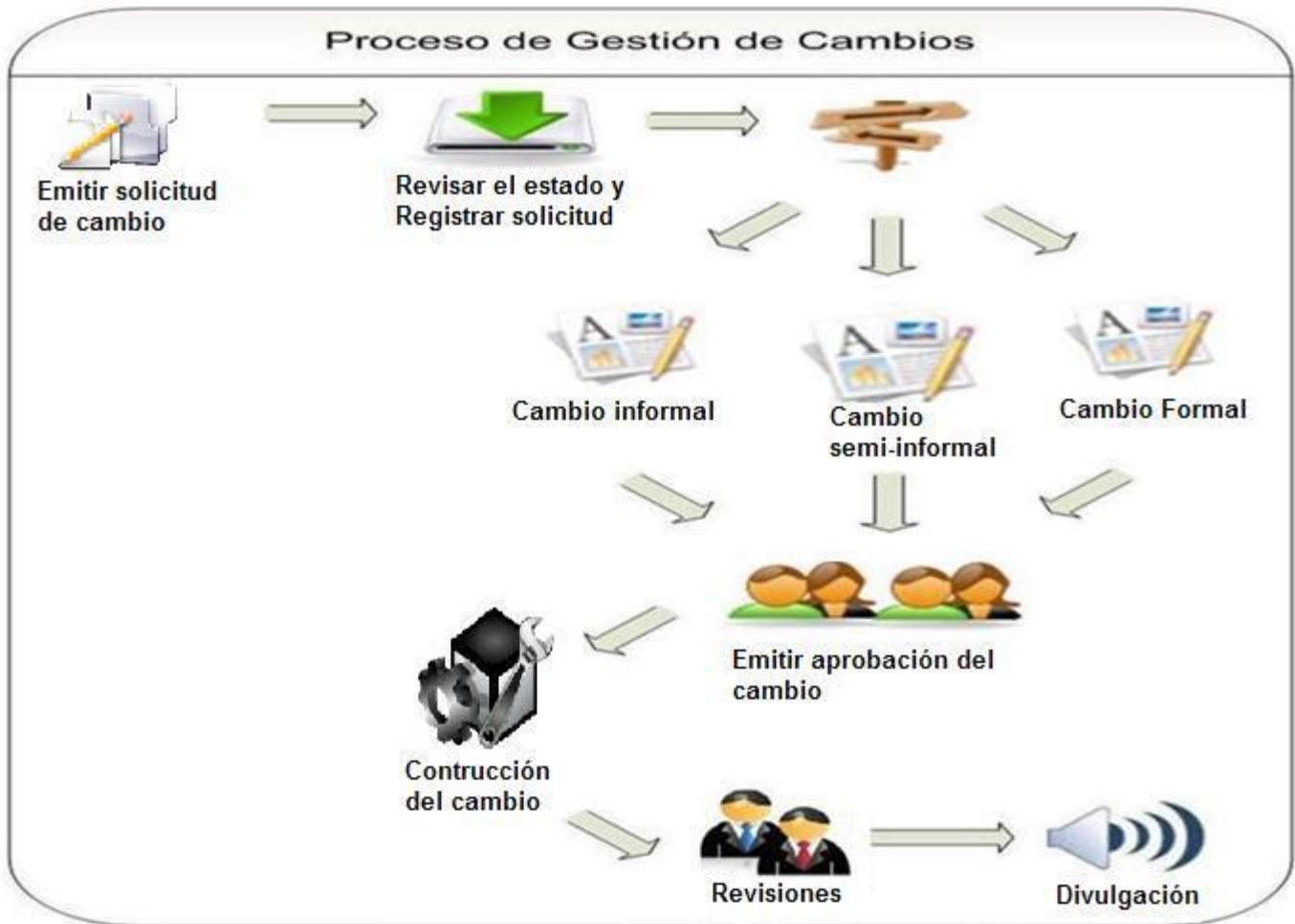


Fig. Proceso de Control de Cambios

2.4.4 Auditoría de la configuración

¿Cómo podemos asegurar que el cambio se implementó correctamente? La respuesta a esta interrogante se verá reflejada a través:

- ✓ Revisiones técnicas formales: corrección técnica del objeto de configuración que se ha modificado; se la debe realizar en casi la mayoría de los cambios triviales.
- ✓ Auditoría de la configuración del software: complementa la revisión técnica formal.

La auditoria de la configuración asegura que los ECS correctos se han incorporado en una construcción específica y que la documentación está actualizada y es consistente con la versión que se ha construido.

2.4.4.1 Actividad de establecer los tipos de auditorías

Para la realización de auditorías, el auditor será el encargado de establecer los tipos de auditorías que pueden ser realizadas según las necesidades que se apremian. Estos tipos de auditorías realizados son reflejados en el Plan de Gestión de Configuración.

2.4.4.2 Actividad de auditar la configuración

Esta actividad será protagonizada por el auditor el cual se encargará de emitir las auditorías pertinentes en dependencia del tipo de auditoría y emitirá los resultados obtenidos en esta. Con estas auditorías se podrán detectar problemas desconocidos hasta el momento e inclusive permitirá conocer el grado de calidad con que se desarrollan los procesos de GCS en nuestro proyecto.

2.4.5 Generación de Informes de estado

Responde a las siguientes preguntas:

1. ¿Qué pasó?
2. ¿Quién lo hizo?
3. ¿Cuándo pasó?
4. ¿Qué más se afectó?

La respuesta a estas interrogantes se podrá apreciar a través de un pequeño ejemplo.

En caso de que el interesado en el cambio emita una solicitud de cambio, en la cual se solicite un cambio de condiciones de acceso a los documentos esta solicitud deberá poseer una fecha con la estructura de día/mes/año en que se solicita el cambio. Para que una vez que el CCC emita la aprobación del cambio se pueda apreciar si se afectó o no algún ECS en este caso específico la Tabla de Acceso y Seguridad.

Básicamente la generación de informes de estado se encargará de producir los informes para todos los cambios y cosas que se le hagan al software por cada uno de los desarrolladores, así todos están al tanto

de lo que se va haciendo, de esta forma se evita el síndrome de la mando izq. ignora lo que hizo la derecha, y un desarrollador va a cambiar lo que otro ya hizo.

El flujo de información del proceso de IEC lo observamos en la Figura 4. Al asignarse una identificación nueva a un ECS se efectúa una entrada de IEC. Cada vez que el CCC aprueba un cambio se genera una entrada en el IEC. Al realizarse una auditoria los resultados se reportan como parte de la tarea de IEC. El resultado del IEC es posible colocarlo en una base de datos. Por último se genera un IEC con regularidad y su finalidad es mantener a los gestores y profesionales alertas ante los cambios importantes.

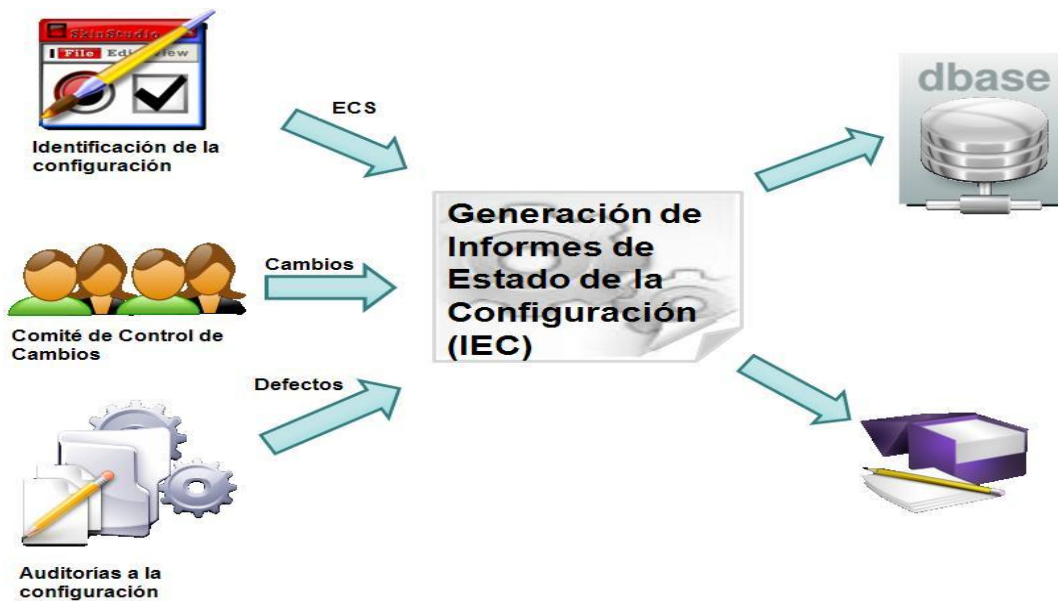


Fig. 4 Proceso de IEC (Informe de Estado de la Configuración)(12)

2.4.5.1 Actividad de establecer tipos de reportes

En esta actividad el gestor de configuración y cambios establecerá los tipos de reportes que pueden ser emitidos según las necesidades que urgen en el proyecto.

2.4.5.2 Actividad de realizar reportes

Los reportes se generarán en dependencia del momento en que se encuentre el desarrollo del proyecto y las informaciones que sean necesarias, sin subestimar en ningún momento la importancia y necesidad de la generación de cada uno de estos reportes de configuración. Esta actividad será realizada por el gestor de configuración y cambios.

2.5 Conclusiones

En este capítulo se realizó un análisis del estado actual de la GCS en el proyecto, consecuentemente se realizó una estrategia organizativa para el equipo de desarrollo del mismo. Se propuso la creación de varios roles no concebidos hasta el momento dentro del proyecto. Se conformó una estrategia a seguir para gestionar la configuración y se definieron sus procesos ligados intrínsecamente con sus actividades.

Capítulo 3

Implantación de la estrategia de GCC para el grupo de proyecto GDA

3.1 Introducción

Teniendo como base la estrategia anteriormente planteada se puede definir cómo va a quedar implantada la estrategia a seguir por el proyecto GDA. Se definen los procesos a seguir durante la GCC en el proyecto, así como las actividades, los responsables y los artefactos generados durante la configuración.

3.2 Metodología a usar

En el grupo de proyecto Gestión Documental y Archivística la GCC se verá identificada por la metodología DIRKS debido a las características peculiares que presenta este grupo de proyecto.

“Las siglas “DIRKS” viene del inglés: Designing and Implementing Recordkeeping System, lo que puede traducirse al español: Diseño e implementación de sistemas para almacenamiento y mantenimiento de registros.

Fue desarrollado en Marzo del 2000 como una guía por el Archivo Nacional de Australia.

Esta metodología se lleva a cabo en varios pasos y contiene una serie de buenas prácticas para desarrollar un sistema de mantenimiento de registros ya sea automatizado o no.

Constituye además la piedra angular en el conjunto de las mejores prácticas, normas y directrices de registros publicados por el Archivo Nacional de Australia que se inspira en las mejores prácticas descritas en la Norma Australiana 4390-1996, Gestión de Documentos y la Norma Internacional ISO 15489, Administración de Registros.

DIRKS es un enfoque riguroso y estructurado destinado a garantizar que los registros y la gestión de la información están firmemente enfocados en las necesidades de la organización.” (22)

3.3 ECS seleccionados

- ✓ Informe de la investigación preliminar
- ✓ Registro de la Investigación
- ✓ Cuadro de Clasificación
- ✓ Modelos de flujo de procesos
- ✓ Especificación de requisitos
- ✓ Tabla de Plazos de Transferencia
- ✓ Tabla de Acceso y Seguridad
- ✓ Informe de Evaluación de Sistemas Existentes
- ✓ Informe de estrategias
- ✓ Diagrama de entidades de la empresa
- ✓ Diagrama de modelos de contenidos
- ✓ Diseño del Cuadro de clasificación
- ✓ Diseño de la Tabla de Plazos de Transferencias
- ✓ Diseño de la Tabla de acceso y seguridad
- ✓ Diagrama de Flujos de Proceso
- ✓ Código fuente
- ✓ Manual de usuario
- ✓ Documento de integración
- ✓ Documento Escenario(s) de prueba unitaria
- ✓ Archivo de Bugs
- ✓ Documento Plan de prueba
- ✓ Documento de gestión de Bugs

Descripción de los ECS seleccionados

Para un mejor entendimiento de las funciones de estos elementos, se realizará una breve descripción de los ECS seleccionados, teniendo en cuenta los flujos en que se desarrollan y los encargados en ejecutarlas.

Flujo 1. Investigación preliminar

Descripción: Realizar investigación en la organización para obtener una comprensión del contexto en el que la organización desarrolla su actividad así como un análisis a fondo de las necesidades actuales del sistema.

ECS: Informe de la investigación preliminar

objetivo: Proporcionar la comprensión del contexto en el que la organización desarrollo su actividad, mediante un examen de la misión, visión, valores de la organización, política , estrategias, objetivos, estructura organizativa, regulaciones específicas a las cuales están sometidas sus actividades y de los factores de éxito relacionados con la gestión documental.

Rol que ejecuta

- ✓ Gestor de configuración y cambios
- ✓ Arquitecto de software principal

Rol que supervisa

- ✓ Líder de factoría
- ✓ Gestor de configuración y cambios
- ✓ Revisor técnico (del área productiva).

ECS: Registro de la Investigación

Objetivo: Identificar todas las fuentes consultadas en la investigación preliminar.

Rol que ejecuta

- ✓ Especialista (del área productiva).

Rol que supervisa

- ✓ Líder de factoría (de la dirección de la fábrica).
- ✓ Especialista principal de Gestión Documental (del área productiva).
- ✓ Revisor técnico (del área productiva).

Flujo 2. Análisis de las actividades de la organización.

Descripción: Adquirir adecuado conocimiento de las actividades de la organización para así entender mejor las necesidades de la administración de registros.

ECS: Cuadro de Clasificación.

Objetivo: Mostrar las funciones, actividades y operaciones que generan documentos.

Rol que ejecuta

- ✓ Especialista de Gestión Documental (del área productiva).
- ✓ Técnico de Gestión Documental (del área productiva).

Rol que supervisa

- ✓ Líder de factoría (de la dirección de la fábrica).
- ✓ Especialista principal de Gestión Documental (del área productiva).
- ✓ Revisor técnico (del área productiva).

ECS: Modelos de flujo de procesos.

Objetivo: Permite visualizar qué documentos se producen en cada proceso, como se transmite la información de una unidad a otra, y quién es responsable en cada momento del manejo y custodia de los documentos.

Rol que ejecuta

- ✓ Ingeniero Industrial (del área productiva)
- ✓ Especialista de Gestión Documental (del área productiva).

Rol que supervisa

- ✓ Líder de factoría (de la dirección de la fábrica).
- ✓ Especialista Principal de Gestión Documental (del área productiva)
- ✓ Revisor técnico (del área productiva).

Flujo 3. Identificación de los requisitos.

Descripción: identificar todos los requerimientos del sistema que se tienen que cumplir para crear y mantener documentos que evidencien las actividades de la organización.

ECS: Especificación de requisitos.

Objetivo: Listar y detallar las características que el cliente desea en su producto.

Rol que ejecuta

- ✓ Analista de software (del área productiva).

- ✓ Especialista de Gestión Documental (del área productiva).

Rol que supervisa

- ✓ Líder de factoría (de la dirección de la fábrica).
- ✓ Especialista Principal de Gestión Documental (del área productiva)
- ✓ Revisor técnico (del área productiva).

ECS: Tabla de Plazos de Transferencia.

Objetivo: Determinar los plazos durante los cuales se tiene que preservar o eliminar la documentación.

ECS: Tabla de Acceso y Seguridad.

Objetivo: Establecer las condiciones de acceso a los documentos.

Rol que ejecuta

- ✓ Especialista de Gestión Documental (del área productiva).
- ✓ Técnico de Gestión Documental (del área productiva).
- ✓ Analista de Software (del área productiva).

Rol que supervisa

- ✓ Revisor técnico (del área productiva)
- ✓ Especialista principal de Gestión Documental (del área productiva)

Flujo 4. Evaluación de los sistemas existentes.

Descripción: Conocer el sistema utilizado en la actualidad antes de desarrollar uno nuevo, la finalidad es comprobar si la organización posee algún sistema de gestión de documentos, cualquier sistema de gestión de información que proporcione evidencia de sus transacciones o ninguno.

ECS: Informe de Evaluación de Sistemas Existentes.

Objetivo: Mostrar los resultados del análisis realizado teniendo en cuenta los puntos fuertes y débiles de los sistemas existentes para determinar los nuevos requisitos en materia de gestión de documentos.

Rol que ejecuta

- ✓ Analista de Software (del área productiva).
- ✓ Especialista de Gestión Documental (del área productiva).

Rol que supervisa

- ✓ Jefe de la factoría (de la dirección de la fábrica)
- ✓ Especialista principal de Gestión Documental (del área productiva)
- ✓ Revisor técnico (del área productiva).

Flujo 5 Identificación de estrategias.

Descripción: Establecen las estrategias para la administración de registros las cuales afectarán el sistema actual de la organización y por lo tanto el sistema a desarrollar. En este flujo se determinan políticas, normas, procedimientos que serán adoptadas y qué herramientas, tanto informáticas como documentales, hace falta diseñar e implementar con el fin de asegurar la creación y el mantenimiento de los documentos que reflejan la actividad de la organización.

ECS: Informe de estrategias.

Objetivo: Mostrar un informe en el cual se planteen las posibles estrategias de actuación y los beneficios que se esperan obtener.

Rol que ejecuta

- ✓ Especialista de Gestión Documental (del área productiva).
- ✓ Analista de Software (del área productiva).

Rol que supervisa

- ✓ Jefe de la factoría (de la dirección de la fábrica).
- ✓ Especialista principal de Gestión Documental (del área productiva).
- ✓ Revisor técnico (del área productiva).

Flujo 6. Diseño del SGDA.

Descripción: Se describe cómo va a ser el sistema propuesto.

ECS: Diagrama de entidades de la empresa.

Objetivo: Mostrar el diseño de las entidades presentes en el Modelo de flujos de procesos (flujo 2).

ECS: Diagrama de modelos de contenidos.

Objetivo: Mostrar el diseño de los modelos de contenidos.

ECS: Diseño del Cuadro de clasificación

Objetivo: Mostrar el diseño del Cuadro de clasificación.

ECS: Diseño de la Tabla de Plazos de Transferencias.

Objetivo: Mostrar el diseño de la Tabla de Plazos de Transferencias.

ECS: Diseño de la Tabla de acceso y seguridad.

Objetivo: Mostrar el diseño de la Tabla de Plazos de Transferencias.

ECS: Diagrama de Flujos de Proceso.

Objetivo: Diseño del flujo de procesos dentro de la organización.

Rol que ejecuta

- ✓ Diseñador de Software (del área productiva).

Rol que supervisa

- ✓ Arquitecto de Software (del área productiva).
- ✓ Arquitecto de Software Principal (área productiva).
- ✓ Analista de Software (del área productiva).
- ✓ Especialista de calidad (área de calidad).
- ✓ Revisor técnico (del área productiva).

Flujo 7. Implementación

Descripción: Se codifica el sistema diseñado.

ECS

- ✓ Código fuente.
- ✓ Manual de usuario.
- ✓ Documento de integración.

Objetivo: Se muestran los elementos de implementación que integran.

Rol que ejecuta

- ✓ Desarrollador (área productiva).

Rol que supervisa

- ✓ Analista de Software (área productiva).
- ✓ Arquitecto de Software Principal (área productiva).
- ✓ Especialista de calidad (área de calidad).

Flujo 8. Revisión posterior a la implementación

Descripción: Realizar un análisis posterior del proceso de desarrollo del sistema y de los resultados obtenidos después de la implementación.

ECS

- ✓ Documento Escenario(s) de prueba unitaria.
- ✓ Archivo de Bugs.
- ✓ Documento Plan de prueba.
- ✓ Documento de gestión de Bugs.

Objetivo Detallar las discrepancias observadas y proponer las acciones adecuadas para corregir las disconformidades que se hayan encontrado y adoptar las oportunidades de mejora que no se hayan previsto en las etapas anteriores.

Rol que ejecuta

- ✓ Especialista de prueba

Rol que supervisa

- ✓ Especialista principal de calidad (área de calidad).
- ✓ Especialista Principal de Gestión de Calidad (área productiva).
- ✓ Arquitecto de información (área productiva).
- ✓ Jefe de factoría (dirección de la fábrica).

3.4 Proceso de Gestión de Configuración del Software en el proyecto GDA

En la siguiente figura se puede observar las distintas actividades que se llevan a cabo en el proceso, conjuntamente con los encargados en desarrollarlas y los artefactos generados a lo largo del ciclo de vida del proceso, los cuales deben cumplir con los estándares de calidad definidos para que al culminar el proceso se obtengan resultados satisfactorios. Es válido destacar que el proceso de GCS será tratado por separado de la GCC.

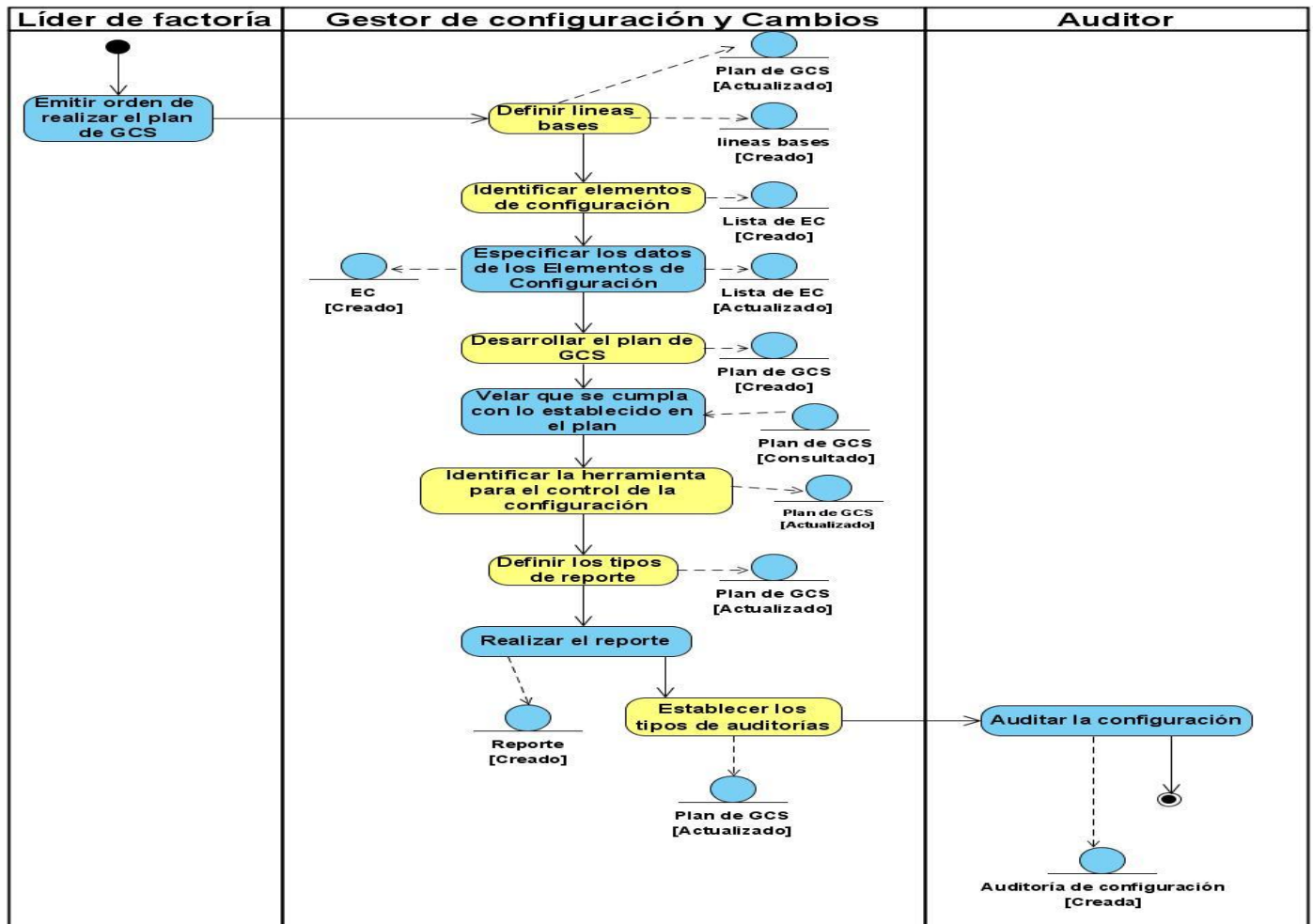


Figura Flujo de trabajo de la GCS en el proyecto.

3.5 Proceso de control de cambios en el proyecto

En la estrategia propuesta en el capítulo anterior están definidos los roles para llevar a cabo el proceso de Control de Cambios estos son:

- ✓ Líder de factoría
- ✓ El comité de control de cambios
- ✓ El interesado en el cambio
- ✓ El gestor de cambios y configuración.

Estos roles son los encargados de llevar a cabo un estricto control de cambios en el proyecto y el procedimiento se verá reflejado en un diagrama de actividades en donde se detallarán todas las actividades llevadas a cabo por estos roles.

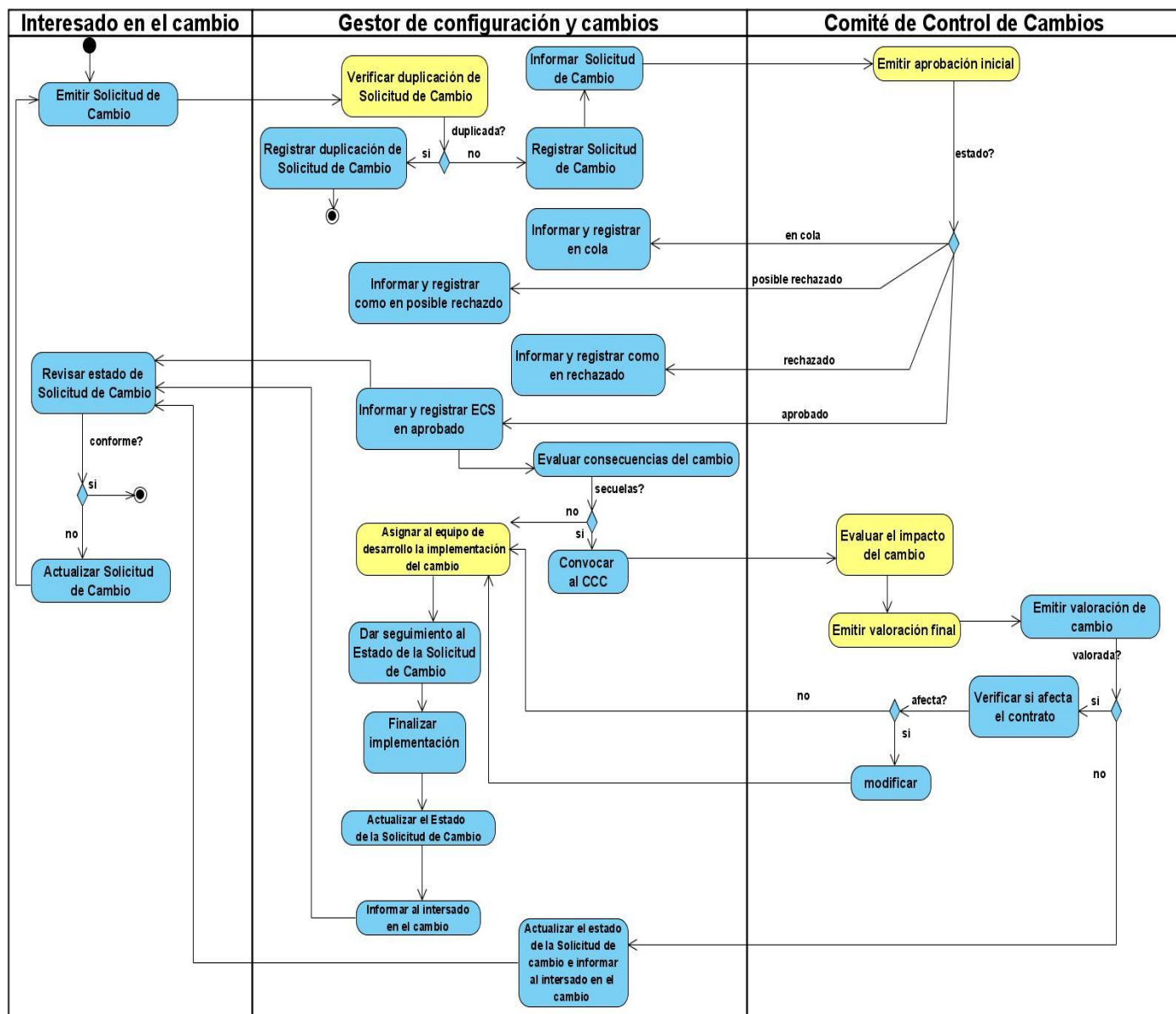


Fig. Flujo de trabajo para controlar los cambios en el proyecto

3.5.1 Estados definidos por los que pasa un cambio en el proyecto GDA

- ✓ **Aprobada:** Solicitudes de cambio que han sido aprobadas por el CCC, pero no están desarrollándose.
- ✓ **Cerrada:** Solicitud de cambio que ya fue resuelta.

- ✓ **En cola:** Solicitud de cambio por analizar por el CCC.
- ✓ **En desarrollo:** Solicitud de cambio aprobada por el CCC y está siendo desarrollada.
- ✓ **Posible rechazada:** Solicitud de cambio en espera de los datos solicitados.

En el proyecto GDA se aplicará la visión de realizar todo un proceso formal y serio con los cambios sobre los elementos cuya funcionalidad y estructura sean aprobadas tanto por parte de los desarrolladores como por parte del cliente, pero no se limitará a la implementación de esta idea dejando que los cambios sobre el resto de los elementos que aún se encuentran en desarrollo y no han sido probados finalmente, sean desarrollados sin tener en cuenta la importancia de los mismos. Si bien son importantes los primeros, los últimos también han de serlo, pues un cambio sobre un elemento que aún no ha sido sometido a revisiones puede implicar cierto retroceso en el desarrollo, afectando de esta manera la planificación concebida en un inicio e incluso las estimaciones realizadas con anterioridad por el líder de factoría. Por esta razón se hace necesario e importante buscar una manera de dar un seguimiento a estos últimos cambios a los cuáles se hace referencia, pero siempre tratando de mantener la mayor fluidez posible.

3.5.2 El comité de control de cambios. Su funcionamiento

El comité de control de cambios está conformado por miembros del equipo de desarrollo que cumplen diferentes roles dentro del proyecto, incluso puede darse el hecho de que un representante de la parte del cliente se encuentre dentro del comité de control de cambios. Además se recomienda que el líder de factoría esté siempre presente dentro del comité de control de cambios. El objetivo principal de este comité radica en realizar una valoración dada una Solicitud de Cambio determinada y emitir un criterio sobre si el cambio debe ser implementado o no. Cada integrante del Comité de Control de Cambios, analiza desde su posición dentro del proyecto la repercusión del cambio solicitado y cómo se ve afectado, y luego emite la evaluación con las posibles afectaciones y mejoras. Posteriormente, con las valoraciones individuales se realiza una evaluación en conjunto, que será la portadora de la información que utilizará el líder de factoría, para emitir la decisión final de si se realiza o no el cambio.

3.5.2.1 Comité de Control de Cambios (CCC). Estructura

El Comité de Control de Cambios en el proyecto estará constituido por: El Líder de factoría, El Gestor de Configuración y Cambios, el Arquitecto de software principal del Proyecto y el Analista de software.

Para facilitar el manejo de la GCS se han repartido los roles que conforman el CCC por las distintas líneas de investigación del proyecto, estos roles a su vez, serán los jefes de líneas que conjuntamente con los responsables de la GCS de la línea correspondiente gestionarán la configuración. Esto trae como beneficio que en el momento de evaluar un cambio se cuente con una visión de la afectación que podría conllevar a cada línea de investigación.

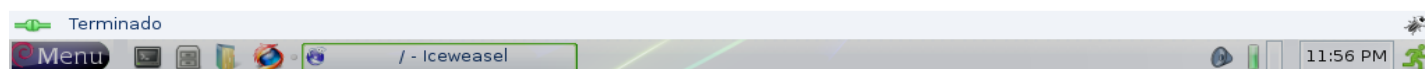
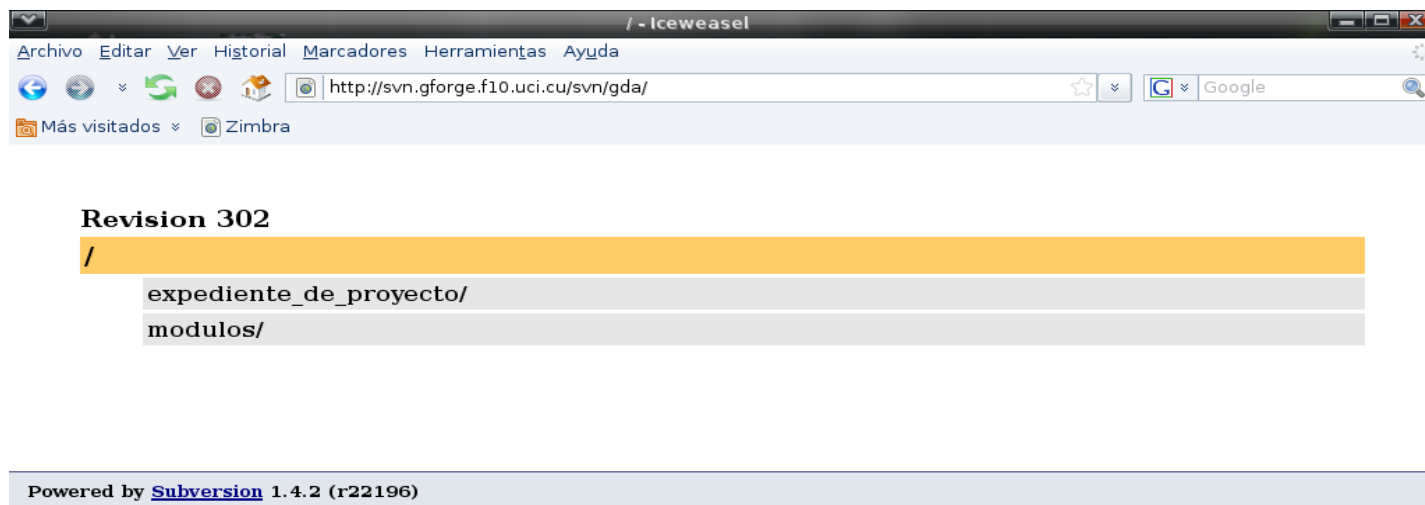
Líneas de Investigación	Jefe de Línea	Responsable de la GCS
Investigar y desarrollar sobre Gestión de Información y el Conocimiento	Líder de factoría	Administrador de Configuración
Investigar y desarrollar sobre Repositorios de Información (Estándares e Implementación)	Analista de Software	Analistas
Investigar y desarrollar sobre Gestión de Contenido Web	Arquitecto de Software Principal	Especialista Principal de Gestión Documental
Investigar y desarrollar sobre Plataformas ECM (Enterprise Content Management)	Gestor de CC	Desarrollador
Investigar y desarrollar sobre Plataformas CMS (Content Management System)	Arquitecto de Software Principal	Desarrollador
Reconocimiento de caracteres (OCR)	Líder de factoría	Desarrollador
Indexado de documentos	Gestor de CC	Especialista Principal de Gestión Documental

3.6 Control de Versiones

Para el control de versiones se utiliza la herramienta Subversion (svn), por el abanico de facilidades que este brinda. En la facultad 10 los todos los proyecto utilizan esta herramienta y realizan la gestión de proyectos a través de la herramienta Gforge, en la cual está incluido el repositorio de versiones de cada proyecto en específico, para acceder a este solo se debe ser integrante del proyecto al cual se desee acceder, en el caso específico del proyecto GDA la dirección para acceder a este es:

<http://svn.gforge.f10.uci.cu/svn/gda/>

En esta dirección se encuentra el svn del proyecto GDA, una vez que se accede a esta dirección aparece una interfaz como la que se representa a continuación.

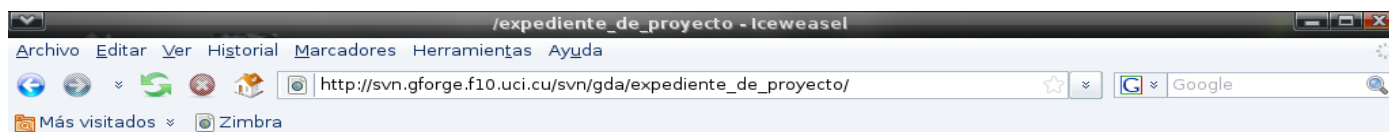


Como se puede apreciar el svn del proyecto GDA consta de dos raíces principales:

- ✓ Expediente de proyecto
- ✓ Módulo

El expediente de proyecto está organizado de la siguiente manera:

- ✓ Cronograma de trabajo
- ✓ Gestión de proyecto
- ✓ Ingeniería
- ✓ Soporte



Revision 302

/expediente_de_proyecto

[\[Parent Directory\]](#)

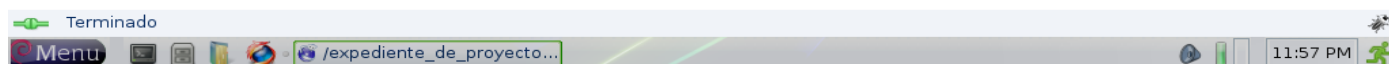
[cronograma de trabajo/](#)

[gestion de proyecto/](#)

[ingenieria/](#)

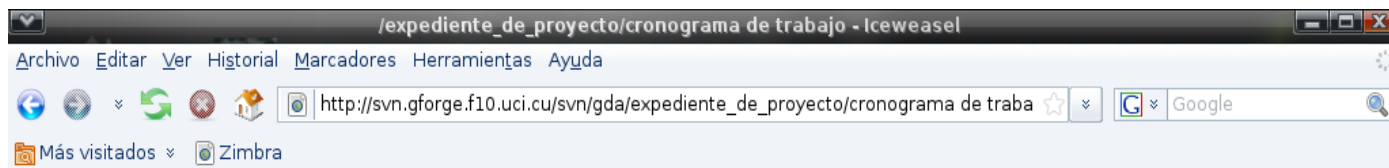
[soporte/](#)

Powered by [Subversion](#) 1.4.2 (r22196)



http://svn.gforge.f10.uci.cu/svn/gda/expediente_de_proyecto

Cada raíz incluye una serie de subraíces o subcarpetas en las cuales están incluidos una serie de datos importantes para el proyecto. Por ejemplo dentro de la carpeta cronograma de trabajo aparece un documento Excel con el cronograma de trabajo del proyecto, como se muestra a continuación.



Revision 320

/expediente_de_proyecto/cronograma de trabajo

[\[Parent Directory\]](#)

[cronograma de trabajo.xls](#)

Powered by [Subversion](#) 1.4.2 (r22196)



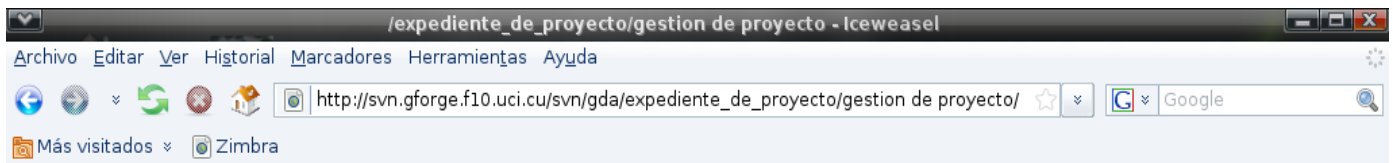
[http://svn.gforge.f10.uci.cu/svn/dga/expediente de proyecto/cronograma de trabajo](http://svn.gforge.f10.uci.cu/svn/dga/expediente_de_proyecto/cronograma de trabajo)

Continuamente otra de las carpetas contenidas por la carpeta raíz es la de gestión de proyecto la cual contiene una serie de sub carpetas las cuales son:

- ✓ Contratación
- ✓ Informacion_de_clientes

- ✓ Plan_del_proyecto
- ✓ Recurso
- ✓ Reuniones
- ✓ Riesgos

La interfaz es la que se muestra a continuación.



Revision 320

/expediente_de_proyecto/gestion de proyecto

[\[Parent Directory\]](#)

[contratacion/](#)

[informacion_de_clientes/](#)

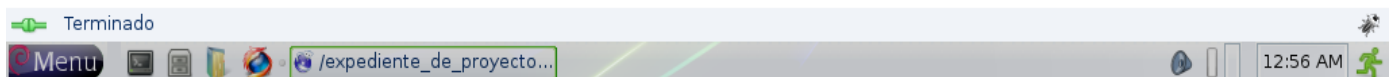
[plan_del_proyecto/](#)

[recursos/](#)

[reuniones/](#)

[riesgos/](#)

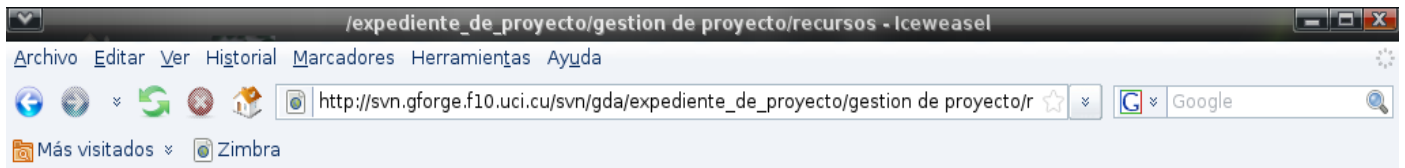
Powered by [Subversion](#) 1.4.2 (r22196)



http://svn.gforge.f10.uci.cu/svn/dga/expediente_de_proyecto/gestion_de_proyecto

De aquí es válido mencionar el contenido de las carpetas recursos y reuniones.

Dentro de la carpeta recurso aparecen un conjunto de sub carpetas y documentos tales como: cursos de capacitación impartidos en el proyecto, el Horario de Producción establecido, así como el documento de Recurso Humanos y dos versiones de los documentos Roles y Responsabilidades y documento Plan de capacitación para los miembros del proyecto, en ambos casos la versión más actualizada es la 1.2. Como se muestra en la siguiente figura.



Revision 320

/expediente_de_proyecto/gestion de proyecto/recursos

[Parent Directory]

cursos de capacitacion/

Horario de Producción.doc

Plan de capacitación 1.2.doc

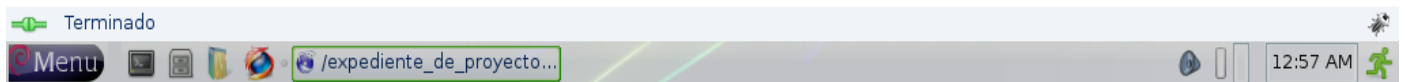
Plan de capacitación.doc

Recursos Humanos.xls

Roles y responsabilidades 1.2.doc

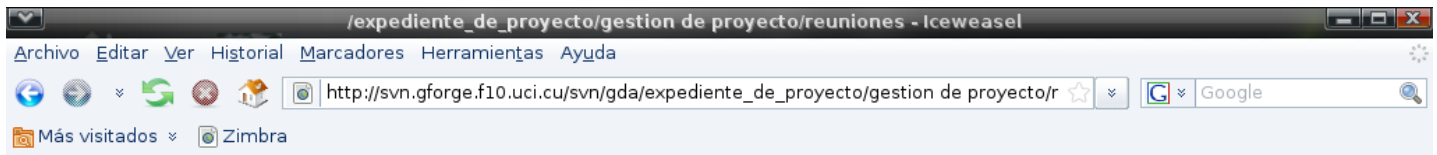
Roles y responsabilidades.doc

Powered by [Subversion](#) 1.4.2 (r22196)



http://svn.gforge.f10.uci.cu/svn/dga/expediente_de_proyecto/gestion de proyecto/recursos

La subcarpeta reuniones contiene un conjunto de documentos en formato Word llamados minutas de reuniones realizadas en distintos periodos de tiempo como muestra la siguiente figura.



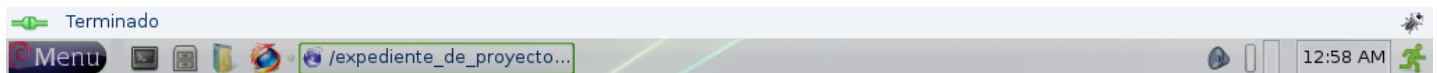
Revision 320

/expediente_de_proyecto/gestion de proyecto/reuniones

[Parent Directory]

- Minuta Reunión en el día 05 09 08.doc
- Minuta Reunión en el día 05.doc
- Minuta Reunión en el día 08 09 08.doc
- Minuta Reunión en el día 15 09 08.doc
- Minuta Reunión en el día 19 09 08.doc
- Minuta Reunión en el día 22 09 08.doc
- Minuta Reunión en el día 26 09 08.doc
- Minuta Reunión en el dia 01 10 08.doc
- Minuta Reunion en el día 29 09 08.doc

Powered by [Subversion](#) 1.4.2 (r22196)



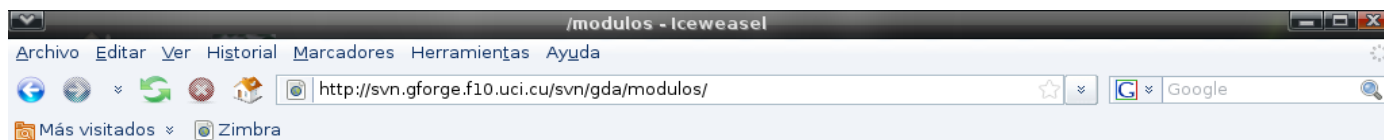
http://svn.gforge.f10.uci.cu/svn/dga/expediente_de_proyecto/gestion de proyecto/recursos

Por otro lado aparece la carpeta denominada módulo en la cual como su nombre lo indica aparecen una serie de módulos desarrollados por el proyecto los cuales en su gran mayoría están siendo implementados en estos momentos, estos módulos son:

- ✓ DAD
- ✓ Digitalización
- ✓ Drupal_Alfresco

- ✓ EditorXML
- ✓ Firma digital
- ✓ Mensajería
- ✓ Módulo_correspondencia
- ✓ Servicios
- ✓ ServicioSDK
- ✓ Tipología_datos

La interfaz es la siguiente:



Revision 302

/modulos

[\[Parent Directory\]](#)

DAD/

digitalizacion/

drupal_alfresco/

editorXML/

firma_digital/

mensajeria/

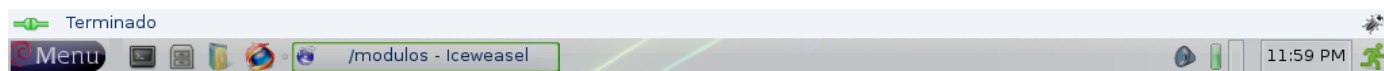
modulo_correspondencia/

services/

servicios_sdk/

tipologia_dato/

Powered by [Subversion](#) 1.4.2 (r22196)

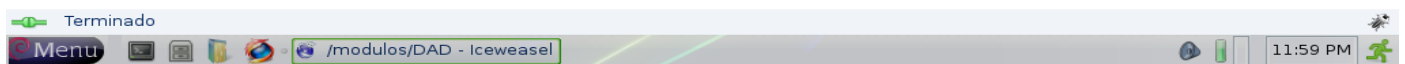


<http://svn.gforge.f10.uci.cu/svn/dga/modulos>

Los módulos a su vez contienen una serie de subcarpetas las cuales en la inmensa mayoría son:

- ✓ Código fuente
- ✓ Documentación

Por ejemplo en el módulo DAD quedaría de la siguiente manera.

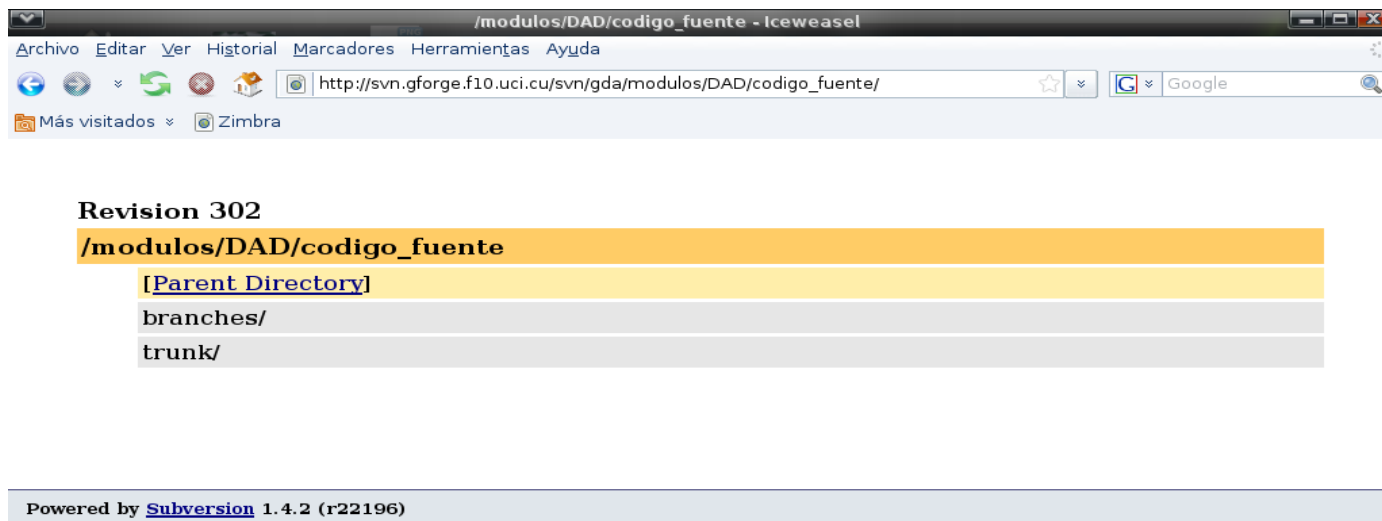


<http://svn.gforge.f10.uci.cu/svn/dga/modulos/DAD>

También se mantiene con un formato bastante estándar las carpetas creadas dentro de codigo_fuente la cual está integrada de dos subcarpetas:

- ✓ Trunk
- ✓ Braches

Por ejemplo en el módulo DAD, dentro de la carpeta código_fuente estarían estas dos subcarpetas como se muestra a continuación.



La carpeta denominada truck (tronco), es el lugar donde se deberá almacenar la línea base de proyecto. Cada desarrollador tiene una carpeta denominada braches (ramas), donde se verán reflejadas las distintas tareas desarrolladas. Este branches solamente podrá ser modificado por el desarrollador al cual pertenece. En caso de que algún desarrollador necesite llevar a cabo algún cambio en otro branches, lo lleva a cabo en una carpeta denominada unstable (inestable), la cual se utiliza solamente para realizar los cambios por desarrolladores, carpeta en la cual varía mucho la información y es muy inestable. Luego el Jefe de Línea y el responsable de la Gestión de Configuración de la Línea verán si son pertinentes estos

cambios o no, para luego de ser revisadas por el Jefe de Línea ser subidos al tronco. En el unstable no se va a llevar a cabo el proceso de Gestión de Cambios, porque es una carpeta de trabajo inestable.

3.7 Auditorías a la Configuración

Con el fin de garantizar que las tareas de la GCS se realicen apropiadamente dentro del proyecto, el equipo de calidad y el gestor de configuración y cambios, deberán cuidar por ello valiéndose de las auditorías de configuración.

Este proceso es sumamente importante y se considera que nunca se debe pasar por alto, por lo que se debe garantizar la realización del mismo por parte de proyecto.

Las auditorías a realizar son las siguientes:

Auditorías a los cambios: Estas auditorías se realizan para verificar que los cambios hayan sido implementados correctamente.

Auditorías a los elementos de configuración: Cuando se realizan auditorias a los elementos de configuración estas se realizan para conocer si estos elementos cumplen con las funcionalidades requeridas y así poder reconocer la estabilidad de los mismos.

3.8 Generación de Informes de Estado de la Configuración

En el transcurso de este proceso se desarrollan un conjunto de actividades las cuales proporcionan diversas facilidades, las cuales tienen como objetivo en común el de mantener informado a los interesados el estado en que se encuentra la configuración. Estas actividades son las de Realizar reportes y Establecer tipos de reportes, ambas ya descritas en el capítulo anterior. Los tipos de reportes a generar en el proyecto deben aparecer en el Plan de Gestión de Configuración y serán los siguientes:

Reporte de modificaciones a ECS: En este reporte se recogen las distintas modificaciones registradas a elementos de configuración en un periodo determinado, además de los responsables y las fechas de modificación.

Reporte de incidencias y soluciones: Este reporte por su parte recoge las incidencias ocurridas en el proyecto durante el ciclo de vida del mismo así como las respuestas dadas a las mismas en caso de que hayan sido solucionadas.

Reporte de las Solicitudes de Cambio: Se recoge en este reporte un resumen de todas las solicitudes de cambio emitidas y el estado en que se encuentran en estos momentos las mismas con sus responsables y fechas en que se alcanzó el estado.

Reporte inventario de los elementos de configuración del software: Recoge un resumen de las características más importantes de cada elemento de configuración en un momento determinado.

Reporte de versiones: Informa sobre determinados elementos de configuración, las versiones por las que han transcurrido estos elementos hasta llegar a la versión actual, especificando las características que marcan la diferencia entre una versión y otra.

Cualquier otro reporte solicitado: Tanto el cliente como cualquier otra persona del proyecto, pueden necesitar y solicitar cualquier otro tipo de información referente a la configuración del mismo.

3.9 Conclusiones

En este capítulo quedó establecida completamente la estrategia a seguir por el proyecto, con el fin de gestionar la configuración y los cambios así como los pasos a seguir para implantar todos los procesos ya definidos con anterioridad en esta estrategia.

Conclusiones Generales

A lo largo de esta investigación se pudo definir e implantar una estrategia de 5 procesos y una serie de actividades para la gestión de configuración y cambios en el proyecto de GDA, se pudo además con esta investigación saber el estado actual del proyecto referente a la GCC y establecer así esta estrategia que se adapta a las necesidades del mismo. Además se estableció y se aprobó satisfactoriamente el Plan de Gestión de Configuración, artefacto que identifica esta disciplina y permite en gran medida la organización del trabajo. Es válido destacar que también se definieron todos los ECS cuya evolución y desarrollo se deben monitorear continuamente, también se establecieron las líneas bases tomando como guía los elementos de configuración previamente seleccionados, se definieron roles específicos con el fin de mejorar la organización y gestionar la configuración desde las mismas líneas de investigación de proyecto, a su vez se logró la aclimatación de los miembros de proyecto en cuanto a las herramientas usadas dígase SVN o Gforge.

Por todo lo anteriormente expuesto se logró darle cumplimiento al objetivo general así como a los distintos objetivos específicos pues fue desarrollado íntegramente el modelo que permite establecer una adecuada GCC al proyecto, por lo que el mismo constará con una estrategia de GCC bien definida y organizada

Recomendaciones

1. Realizar un estudio para evaluar los resultados de la estrategia implantada.
2. Realizar un estudio de factibilidad para comprobar si una vez aplicada la estrategia se economiza el tiempo, los recursos y el esfuerzo invertido en el trabajo.

Bibliografía Citada

1. DESCONOCIDO. **Gestión de Configuración del Software**. 2006 2007, n° Disponible en:
<http://www.histaintl.com>.
2. ANTONIO, A. D. **La Gestión de la Configuración del Software**. 59 p.
3. JAVIER TUYA, M. J. S.-C., ADENSO DIAZ. **Gestión de la Configuración del Software en un proyecto de I+D**. Julio 2000 n° Disponible en: <http://www.di.uniovi.es/~tuya/pub/jcs-2000.html>.
4. PRESSMAN, R. S. **Ingeniería del Software. Un Enfoque práctico**. 2005. vol. 1, 614 p.
5. HASS, A. M. **Configuration Management Principles And Practice**. Addison-wesley Professional, 2002.
6. IEEE. **Standard Glossary of Software Engineering Terminology**. diciembre 1990, n°
7. ---. **Standard for Software Project Management Plans**. Enero de 1987, n°
8. FEBLES, A. **Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software**. La Habana, Cuba: 2004.
9. NAVARRO, A. **Gestión de la configuración software**. n°
10. DANIELE, M. **Teoría GCS**. Teoría 6 ISO 2007, 2007.
11. DESCONOCIDO. **ROM**. Junio de 2005. 13 p.
12. CABRERA, I. A. **Gestión de Cambio**. Mayo del 2007, n° Disponible en:
<http://www.mygnet.net/articulos/software/1082/>.
13. DESCONOCIDO. **Rational ClearCase** Disponible en: http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=es_ES&synkey=Z012568L96063G61.

14. ---. **ClearQuest** Disponible en: http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=es_ES&synkey=W108274Q56011S81.
15. ALTAMIRANDA, M. J. **Manual de usuario para el uso del Gforge**. Mérida, Venezuela: 2006.
16. DESCONOCIDO. **Resumen General de ISW**. n°
17. LORIÉ, Y. Y. S. y SEGURA, D. H. **La Gestión de Configuración y Cambios en el proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE)**. Investigativa, UCI, 2007.
18. CARBOL, Y. F. **Estrategia para la Gestión de Configuración de Software del Proyecto Registro y Notaria**. Universidad de las Ciencias Informática., 2007.
19. BENÍTEZ., M. Y. N. G. Y. R. **Estrategia para la Gestión de Configuración y Cambios en el Centro UCID**. Universidad de las Ciencias Informáticas, 2008.
20. DART, S. **Gestión de configuración - El eslabón perdido en Ingeniería Web**.
21. RODRÍGUEZ, A. M. G. **Propuesta de Estrategia para la gestión de configuración en el proyecto sistema de gestión penitenciaria**. Universidad de las Ciencias Informáticas, 2007.
22. KATIA SANTANA, L. **Propuesta de fábrica de software para el grupo de proyecto de GDA**. Universidad de las Ciencias Informáticas, 2009.

Bibliografía Consultada

DESCONOCIDO. **Gestión de Configuración del Software**. 2006 2007, n° Disponible en:
<http://www.histaintl.com>

CABRERA, I. A. **Gestión de Cambio**. Mayo del 2007, n° Disponible en:
<http://www.mygnet.net/articulos/software/1082/>.

RODRÍGUEZ, A. M. G. **Propuesta de Estrategia para la gestión de configuración en el proyecto sistema de gestión penitenciaria**. Universidad de las Ciencias Informáticas, 2007

BENÍTEZ., M. Y. N. G. Y. R. **Estrategia para la Gestión de Configuración y Cambios en el Centro UCID**. Universidad de las Ciencias Informáticas, 2008

ALTAMIRANDA, M. J. **Manual de usuario para el uso del Gforge**. Mérida, Venezuela: 2006.

PRESSMAN, R. S. **Ingeniería del Software. Un Enfoque práctico**. 2005. vol. 1, 614 p.

FEBLES, A. **Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software**. La Habana, Cuba: 2004.

LORIÉ, Y. Y. S. y SEGURA, D. H. **La Gestión de Configuración y Cambios en el proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE)**. Investigativa, UCI, 2007.

NAVARRO, A. **Gestión de la configuración software**. n°

DESCONOCIDO. **Gestiondelaconfiguracion.pdf**

MINISTERIO DE ADMINISTRACIONES PÚBLICAS. **Gescon.pdf**

VALENCIA, M, E. **Gestión de la Configuración del Software**

DESCONOCIDO. **Universidad de Rey Juan Carlos**

Anexos

Anexo 1: Formulario de Solicitud de cambio

<p>Formulario de Solicitud de Cambio</p> <p>Nombre del proyecto: _____</p> <p>Producto: _____</p> <p>Versión: _____</p> <p>Solicitado por: Nombre, apellidos: _____</p> <p>Rol Desempeñado: _____</p> <p>Título del cambio: _____</p> <p>Descripción: _____</p> <p>Elementos de Configuración Afectados: _____</p> <p>Beneficios o razones para el cambio _____</p> <p>Impacto: _____</p> <p>Estado: ___ Rechazado ___ Posible rechazado ___ Aprobado ___ En cola</p> <p>Prioridad:</p> <ul style="list-style-type: none"><input type="checkbox"/> Máxima<input type="checkbox"/> Media<input type="checkbox"/> Mínima <p>Aprobado por: _____</p> <p>Rechazado por: _____</p>
--