

Universidad de las Ciencias Informáticas

Facultad # 3



**Gestión de Configuración del Software para el
Departamento de Desarrollo de Software.**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Carlos Manuel Galán González.

Tutor: Lic. Marcia García Alfonso.

La Habana, Junio del 2008

DECLARACIÓN DE AUTORÍA

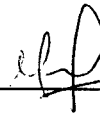
Declaro que soy el único autor de este trabajo y autorizo a la Facultad # 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 8 días del mes de julio del año 2008.



Carlos Manuel Galán González.

Autor.



Marcia García Alfonso.

Tutor.

Agradecimientos

A la universidad de las Ciencias Informáticas por brindarme en todos estos años la esencia y la información necesaria para poder llevar a cabo un buen estudio durante los cinco años de la carrera.

Antes que nada quisiera darle un millón de gracias a mi tutora la licenciada Marcia García Alfonso, por brindarme su ayuda en todo momento y por darle de una manera u otra un aporte tangible al trabajo de diploma.

A mis compañeros de la universidad por haberme brindado la amistad y entrega en todo momento sin importar nada, gracias a mis amistades cercanas por darme los mejores años de estudiante que alguna persona pudiera tener.

A mi familia en general por brindarme la paciencia y el cariño que en todos estos años me han entregado, en especial a mi madre y a mi abuela que son las personas que más amo en mi vida.

A mi novia Lisset, por ser tan buena siempre conmigo y por darme el soporte y el amor que jamás habría pensado tener.

Dedicatoria

Este trabajo de diploma se lo dedico principalmente a mi familia, a todas las personas que de una forma u otra me ayudaron en mis años de estudiante, también se lo dedico a los que no me ayudaron porque de una forma me hicieron una persona más fuerte y con ideas bien firmes.

Pero muy especialmente a la memoria de mi abuelo Orlando Galán Betancourt, que fue la persona que me inculcó todo lo valioso e importante que debe tener un ser humano para poder alcanzar sus temas.

Una y mil gracias Abuelo, siempre estarás en mi corazón.

RESUMEN

El presente trabajo aborda la Gestión de Configuración del Software en el Departamento de Desarrollo de Software del MININT, luego de exponer y analizar varios conceptos brindados por diferentes autores, se describen elementos básicos de importancia para la realización de las tareas de la Gestión de la Configuración del Software (GCS) las cuales son detalladas de forma conceptual, por último se expone las herramientas más factibles que permiten el trabajo con la configuración en un proceso de desarrollo de software, conformando así el estado del arte de la GCS. Se caracterizó el DDS y se compararon herramientas para el control de versiones seleccionándose las más óptimas para el mismo. Por último se propone un procedimiento propio de gestión de configuración de software, así como algunas acciones identificadas para ponerlo en práctica, el mismo está ajustado a las características actuales del departamento y debe resultar de gran importancia para un buen desempeño futuro en cualquier proyecto que se lleve a cabo dentro del departamento, permitiendo la visibilidad del proyecto de software a partir del control de sus elementos de configuración y del registro y la evaluación de los cambios sobre ellos y el control uniforme de sus diferentes versiones. El control central de los elementos reutilizables permitirá crear productos con mayor rapidez y calidad. Como fundamento para la creación del plan de configuración de software para el DDS se tuvo en cuenta el estándar brindado por la IEEE.

Palabras claves: Gestión de Configuración del Software (GCS), Departamento de Desarrollo de Software (DDS), Elementos de Configuración del Software, Línea base, Control de Versiones, Gestión de Cambios, Subversion, Visual SourceSafe, Repositorio, Plan, Procedimiento.

INDICE

AGRADECIMIENTOS II

DEDICATORIA..... III

RESUMEN IV

INDICE V

INDICE DE FIGURAS. VII

INTRODUCCIÓN 1

PROBLEMA 3

OBJETO DE ESTUDIO 3

OBJETIVO GENERAL 3

OBJETIVOS ESPECÍFICOS 3

CAMPO DE ACCIÓN 3

MÉTODO TEÓRICO: 4

MÉTODO EMPÍRICO: 4

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE LA GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE 6

1.1 INTRODUCCIÓN..... 6

1.3 BASES DE LA GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE. 8

1.3.1 EL CAMBIO 8

1.3.2 ELEMENTOS DE CONFIGURACIÓN DEL SOFTWARE. 9

1.3.3 LINEA BASE..... 14

1.3.4 BIBLIOTECAS DE SOFTWARE..... 16

1.4 LOS PROCESOS DE LA GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE. 17

1.4.1 IDENTIFICACIÓN DE LA CONFIGURACIÓN:..... 19

1.4.1.1 ESTABLECIMIENTO DE UNA JERARQUÍA PRELIMINAR DEL PRODUCTO SOFTWARE. 20

1.4.1.2 SELECCIÓN DE LOS ELEMENTOS DE CONFIGURACIÓN. 20

1.4.1.3 DEFINICIÓN DE LAS RELACIONES EN LA CONFIGURACIÓN. 21

1.4.1.4 DEFINICIÓN DE UN ESQUEMA DE IDENTIFICACIÓN. 21

1.4.1.5 DEFINICIÓN Y ESTABLECIMIENTO DE LÍNEAS BASE. 22

1.4.1.6 DEFINICIÓN Y ESTABLECIMIENTO DE BIBLIOTECAS DE SOFTWARE. 23

1.4.2 CONTROL DE VERSIONES EN LA CONFIGURACIÓN. 23

1.4.3 CONTROL DE CAMBIOS EN LA CONFIGURACIÓN.....	29
1.4.4 INFORMES DE ESTADO.....	33
1.4.5 AUDITORIA DE LA CONFIGURACIÓN.....	35
1.5 HERRAMIENTAS PARA LA GCS.....	36
1.6 CONCLUSIONES.....	37
CAPÍTULO 2: CARACTERIZACIÓN DEL DDS Y SELECCIÓN DE HERRAMIENTAS.....	38
2.1 INTRODUCCIÓN.....	38
2.2 CARACTERIZACIÓN DEL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.....	38
2.3 HERRAMIENTAS PARA EL CONTROL DE VERSIONES Y CAMBIOS.....	41
2.3.1 COMPARACIÓN DE SISTEMAS DE CONTROL DE VERSIONES.....	41
LICENCIA.....	42
OPERACIONES EN EL REPOSITORIO.....	43
ATOMIC COMMITS:.....	43
ARCHIVOS O DIRECTORIOS SE MUEVEN O RENOMBRAN:.....	44
COPIAS DE ARCHIVOS Y DIRECTORIOS:.....	45
REPLICA DEL REPOSITORIO REMOTO:.....	46
PROPAGAR CAMBIOS A LOS REPOSITORIOS PADRES:.....	46
PERMISOS DEL REPOSITORIO:.....	47
SOPORTES DE CAMBIOS:.....	48
SIGUIENDO EL HISTORIAL DE LA LÍNEA-SABIA DEL ARCHIVO:.....	48
CAPACIDAD DE TRABAJAR SOLO EN UN DIRECTORIO DEL REPOSITORIO:.....	49
SEGUIMIENTO DE CAMBIOS UNCOMMITTED:.....	50
DOCUMENTACIÓN:.....	51
FACILIDAD DE DESPLIEGUE:.....	51
PAQUETE DE COMANDOS:.....	52
SOPORTE DE RED:.....	53
PORTABILIDAD:.....	54
INTERFAZ WEB:.....	55
DISPONIBILIDAD DE INTERFACES GRÁFICAS DE USUARIOS:.....	55
VENTAJAS DE SUBVERSION.....	56
DESVENTAJAS DE SUBVERSION.....	57
VENTAJAS DE SOURCESAFE.....	58

DESVENTAJAS DE SOURCESAFE.....	58
2.4 CONCLUSIONES.....	59
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN Y VALIDACIÓN DE RESULTADOS.....	60
PROPUESTA DE SOLUCIÓN.....	60
3.3 VALIDACIÓN DE LA PROPUESTA DE PROCEDIMIENTO PARA LA GESTIÓN DE CONFIGURACIÓN EN EL DDS.....	75
3.4 CONCLUSIONES.....	76
CONCLUSIONES GENERALES.....	77
RECOMENDACIONES.....	78
BIBLIOGRAFÍA.....	79
ANEXOS.....	81
ANEXO DE EJEMPLO DE UN PLAN DADO POR IEEE.....	81

INDICE DE FIGURAS.

Ilustración 1 Esquema General.....	11
Ilustración 2 Objetos de configuración.....	12
Ilustración 3 Esquema de identificación.....	13
Ilustración 4 Líneas Base de un proyecto.....	15
Ilustración 5 Proceso de líneas base.....	16
Ilustración 6 Actividades de la GCS.....	19
Ilustración 7 Esquema de revisiones.....	24
Ilustración 8 Esquema de variantes.....	25
Ilustración 9 Grafo de evolución.....	26
Ilustración 10 Sistema de control de versiones.....	27
Ilustración 11 Proceso formal de revisión y cambios.....	30
Ilustración 12 Proceso de control de cambios.....	32
Ilustración 13 Procesos de informes de estado.....	34
Ilustración 14 Procedimiento de componentes.....	61
Ilustración 15 Procedimiento de bibliotecas de software.....	64

Ilustración 16 Esquema 1 de interfaz de configuración..... 66

Ilustración 17 Esquema 2 de interfaz de configuración..... 68

Ilustración 18 Esquema 3 de interfaz de configuración..... 73

Ilustración 19 Esquema 4 de interfaz de configuración..... 75

Introducción

La tecnología se ha vuelto imprescindible y omnipresente, pero el control y gerenciamiento de los proyectos está lejos de ser satisfactorio. Se trata de una fuerza irresistible, que todo lo llena, que todo lo abarca, pero que a menudo queda totalmente fuera de control, consumiendo todos los recursos y arrasando todo a su paso.

En estos tiempos nos encontramos que en cuanto al desarrollo de software todavía la producción es baja y los costos son muy elevados, esto se debe en su mayoría a la no aplicación de técnicas de ingeniería y gestión de software, y la no definición de roles y procesos adecuados en el desarrollo de software. Además existe una ingenuidad total en lo que a planificación de proyectos se refiere, cada año en el mundo fallan miles de proyectos informáticos ya sea por requisitos de usuarios o porque simplemente sobrepasan las expectativas de costo y tiempo, y una de las más importantes que no se toman experiencias de fallos anteriores. Otra potencial causa es la competencia que existe actualmente en el mundo donde si no obtienes un producto confiable simplemente las demás empresas productoras de software te roban el cliente por decirlo de alguna manera.

En el desarrollo de software los cambios, debidos principalmente a modificaciones de requisitos y fallos, son inevitables. Normalmente se trabaja en equipo por lo que es preciso llevar un control y registro de los cambios con el fin de reducir errores, aumentar la calidad y la productividad y evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o a las tareas realizadas por otros miembros del equipo de proyecto.

La ingeniería del software es la encargada de designar un conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la actividad de programación que conlleva. Por supuesto forman parte de esta disciplina, las ciencias computacionales y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada ingeniería informática.

Todo este proceso de ingeniería trae consigo un resultado que no sería otro que la creación del software, concebido después de un largo y cuidadoso proceso de desarrollo de software, hay que agregar que se considera también parte del software creado la documentación correspondiente. El principal objetivo del proceso de desarrollo de software es obtener el producto del software con la mayor calidad y el control posible.

El desarrollo exitoso de un software requiere de visibilidad del proceso productivo que permita ver las acciones que ocurren dentro del ciclo de vida para garantizar que la percepción del usuario sea vista por el desarrollador y que el responsable del proyecto pueda tomar las decisiones para planificar, dirigir y controlar el desarrollo del software basadas en información y no en suposiciones, además se requiere de coordinación para aumentar la productividad disminuyendo los errores. Las técnicas de GCS permiten hacer visible y coordinado el proceso de desarrollo. La integridad de un producto de software depende de la acción combinada de tres tipos de disciplinas. (Antonio, 2001).

-Desarrollo.

-Gestión.

-Control.

La Gestión de Configuración del Software (GCS) se encuentra dentro de las disciplinas de control que se encarga de llevar un control riguroso sobre los productos de software en su creación lo que permite que el proceso de desarrollo de software se realice con calidad y se logre mantener la integridad de los componentes del producto de software incluyendo también la documentación asociada al mismo.

En el contexto del perfeccionamiento del proceso de Informatización del Ministerio del Interior y del desarrollo de la Modernización Tecnológica del Enfrentamiento se hace necesario: Consolidar el proceso de desarrollo de software utilizando metodologías y herramientas asistidas por computadoras e implementar un sistema de calidad del software para este proceso.

Como parte de lo anterior se está creando el Departamento de Desarrollo de Software (DDS) a partir del Grupo de Proyectos Priorizados que existe desde el 2004, en el que se aplicó no en toda su extensión una metodología propia, ADOMET, basada en el análisis y diseño orientado a objetos, que hoy no se ajusta a las tendencias actuales para producir software con calidad.

El DDS se encuentra en una fase organizativa, de la que se pretende obtener como resultado, una estrategia de trabajo común y consecuente con las necesidades reales que tiene la producción de software en el MININT, basada fundamentalmente en el desarrollo de componentes y aplicaciones así como en las novedades tecnológicas que brinda SOA (arquitectura orientada a servicios) y BPMS (gestión y automatización de procesos de negocio) para algunos procesos de desarrollo.

La no existencia de un proceso de desarrollo de software formalizado, ni de un método de gestión de configuración de software, así como el desconocimiento de las herramientas que se necesitan para realizar un buen proceso de gestión de configuración de software implica que no aparece el control sobre las configuraciones del software en el DDS.

Problema

La no existencia de un proceso de gestión de configuración del software dentro del departamento de desarrollo de software para el control de la configuración del software.

Objeto de Estudio

La gestión de configuración del software en el proceso de desarrollo de software del DDS.

Objetivo General

Proponer un procedimiento para la Gestión de Configuración del Software dentro del DDS.

Objetivos Específicos

Caracterizar la situación actual del DDS.

Comparar y seleccionar las herramientas para la gestión de cambios y el control de versiones en el DDS como soporte a la gestión de configuración.

Diseñar e implementar un procedimiento para la gestión de configuración que tenga en cuenta la gestión de cambios y el control de versiones en el DDS.

Campo de Acción

Procedimiento para la Gestión de Configuración del Software en el Departamento de Desarrollo de software.

Para lograr los objetivos se llevaron a cabo las siguientes tareas de investigación:

- Investigación sobre el estado actual de la gestión de configuración en el mundo.
- Estudio sobre el control de versiones y la gestión de cambios dentro de la gestión de configuración del software.

-Estudio sobre las herramientas más utilizadas en la actualidad por la gestión de configuración del software para así definir un procedimiento y seleccionar las necesarias a utilizar para el control de versiones y la gestión de cambios en el departamento de desarrollo de software (DDS).

Para la realización de este trabajo se emplearon los siguientes métodos:

Método Teórico:

-Análisis Sintético: Se utilizó en la investigación para analizar las teorías y documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

Método Empírico:

-Observación: Este método permitió la recoger la mayor cantidad de información posible a partir de la observación del problema.

Este trabajo de investigación cuenta con una introducción general sobre el tema de estudio, tres capítulos, conclusión, referencia, bibliografía, glosario de términos.

Capitulo 1-Contiene la fundamentación teórica del trabajo, con un estudio de los antecedentes de la Gestión de Configuración del Software.

Capitulo 2-Contiene los principales objetivos que se abarcan dentro del DDS y sus características específicas. Además este capítulo abarca toda la investigación de las principales herramientas que se utilizan dentro de la Gestión de Configuración del Software, en específico dentro de la gestión de cambios y el control de versiones como basamento teórico determinando cuales son las más efectivas para el trabajo que se realiza dentro del DDS.

Capitulo 3-Contiene la descripción de la solución propuesta que no es más que la propuesta de un procedimiento para el proceso de Gestión de la Configuración del Software para el DDS.

Por último las Conclusiones del Trabajo, Recomendaciones, Bibliografía utilizada, Anexo y Términos importantes brindan información relevante de la investigación realizada.

CAPÍTULO 1: Fundamentación teórica sobre la gestión de configuración del software.

1.1 Introducción.

En este capítulo se aborda la "Gestión de Configuración del Software (GCS) tanto a nivel internacional como nacional. Se tratan términos más específicos dentro de los cuales se encuentran: el cambio, elementos de configuración, líneas base, bibliotecas de software y repositorios, todos estos englobados dentro de las bases de la gestión de configuración del software, además se abordan la gestión de cambios y control de versiones que son actividades del proceso de la gestión de configuración del software.

1.2 La Gestión de Configuración.

En la actualidad muchas empresas utilizan la gestión de configuración del software como un proceso de mejora continua para acelerar la calidad del software. Códice Software es una empresa española que ofrece una herramienta de Gestión de Configuración de Software basada en un innovador diseño y adaptada a las actuales tendencias del desarrollo. Para que se entienda mejor se trata de un sistema de control de versiones para desarrolladores de software que mantiene bajo control todos los elementos implicados en un desarrollo de software, como código fuente, documentación e imágenes y además coordina a los desarrolladores permitiéndoles trabajar en paralelo.

En nuestro país muchas empresas ya utilizan esta importante rama de la ingeniería de software, un ejemplo tangible de esto se lleva a cabo en la Universidad de las Ciencias Informáticas (UCI) donde diferentes proyectos utilizan la gestión de configuración del software como un pilar importante dentro del desarrollo de software, ejemplo de esto es la Intranet universitaria, el proyecto Registro y Notaria actualmente de los más grandes y completos de nuestra universidad.

A lo largo del ciclo de vida del proceso de software, los productos de software evolucionan. Desde la concepción del producto y la captura de requisitos inicial hasta la puesta en producción del mismo y posteriormente desde el inicio del mantenimiento hasta su retiro, se van realizando una serie de cambios,

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

tanto en el código como en la documentación asociada. La Gestión de Configuración del Software es una disciplina encargada del control de la evolución de los productos de software.

La GCS tiene gran importancia, en el desarrollo del software los cambios son inevitables, ya sea por cuestión de necesidad del cliente o porque el equipo de trabajo decide que se necesita para trabajar mejor. A medida que el producto en desarrollo va creciendo más difícil es controlar los cambios y ahí es donde juega un papel importante la gestión de configuración del software ya que esta rama de la ingeniería del software se encarga de establecer, organizar y controlar los cambios en el software.

Muchos expertos en la materia han dado su definición de lo que es la gestión de configuración del software, a continuación se exponen algunos conceptos del tema en cuestión:

“La Gestión de Configuración es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.” (Babich).

“La Gestión de Configuración del Software es el conjunto de procesos destinados a asegurar la validez de todo producto obtenido durante cualquiera de las etapas del desarrollo de un sistema de información a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo. (Anónimo, 2007) (Anónimo, 2007).

“La Gestión de la Configuración es el arte de identificar, organizar y controlar los cambios que sufre el software.” (Anónimo, 2007)

“La gestión de configuración del software no es un mantenimiento del software, el mantenimiento es la etapa final de la ingeniería hasta que se retire el producto del equipo, la gestión de configuración del software es un conjunto de actividades de seguimiento y control que comienzan cuando se inicia el proyecto de desarrollo del software y termina sólo una vez que el software queda fuera de circulación.” (Anónimo, 2007)

“La gestión de configuración del software es una actividad de garantía de calidad del software que se aplica en todas las fases del proceso de ingeniería del software.” (Armando Cabrera).

Existen más conceptos y planteamientos sobre el tema de la gestión de configuración y en particular uno de los más citados es el planteado por IEEE:

“La Gestión de Configuración es la disciplina que abarca todo el ciclo de vida de la producción de software y productos asociados. Específicamente, requiere de la identificación de los componentes a controlar y la estructura del producto, controla todos los cambios sobre los elementos y garantiza mecanismos para auditar todas las acciones.” (IEEE, 2007)

1.3 Bases de la Gestión de Configuración del Software.

1.3.1 El cambio

Para empezar a adentrarnos dentro de lo que son las bases de la gestión de configuración del software hay que tomar como punto de partida “el cambio”, variable importantísima dentro del proceso de creación del software. 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.

Entonces nos hacemos diferentes preguntas: ¿Por qué cambiar el sistema? ¿Que produce los cambios en el sistema? La respuesta a estas interrogantes se puede encontrar en cuatro aspectos fundamentales y a menudo muy tradicionales dentro del desarrollo del software:

Nuevos requisitos del negocio o condiciones que dictan los cambios en las condiciones del producto o en las normas comerciales.

Nuevas necesidades de los clientes que demandan la modificación de los datos producidos por un sistema basado en computadora.

Reorganización y/o reducción del volumen comercial que provoca cambios en las prioridades del proyecto o en la estructura del equipo de ingeniería del software.

Restricciones presupuestarias o de planificaciones que provocan una redefinición del sistema o del producto.

1.3.2 Elementos de configuración del software.

Un elemento de configuración es 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. (Anónimo)

Un elemento de configuración debe ser un elemento que se pueda definir y controlar de forma separada. Es decir, debe ser una unidad en sí mismo.

Los elementos de configuración del software incluyen:

- Ejecutables.
- Código Fuente.
- Modelos de datos.
- Modelos de procesos.
- Especificaciones de requisitos.
- Pruebas.

Y para cada uno de estos elementos se almacenará al menos:

- Nombre.
- Versión.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-Estado.

-Localización.

Son elementos de Configuración en un proyecto de software:

- El plan de proyecto.
- El plan de Gestión de Configuración.
- El documento de definición de requerimientos.
- Estándares de análisis, diseño, codificación, pruebas, y auditoría.
- Documentos de análisis del sistema.
- Documentos de diseño del sistema.
- Prototipos.
- Documentos de diseño de alto nivel.
- Documentos de diseño de bajo nivel.
- Especificaciones de prueba del sistema.
- El plan de pruebas del sistema.
- El Código fuente del programa.
- Código objeto y ejecutable.
- Especificaciones de pruebas de unidad.
- Planes de pruebas de unidad.
- Documentos de diseño de base de datos.
- Datos de prueba.
- Datos del proyecto.
- Manuales de usuario.

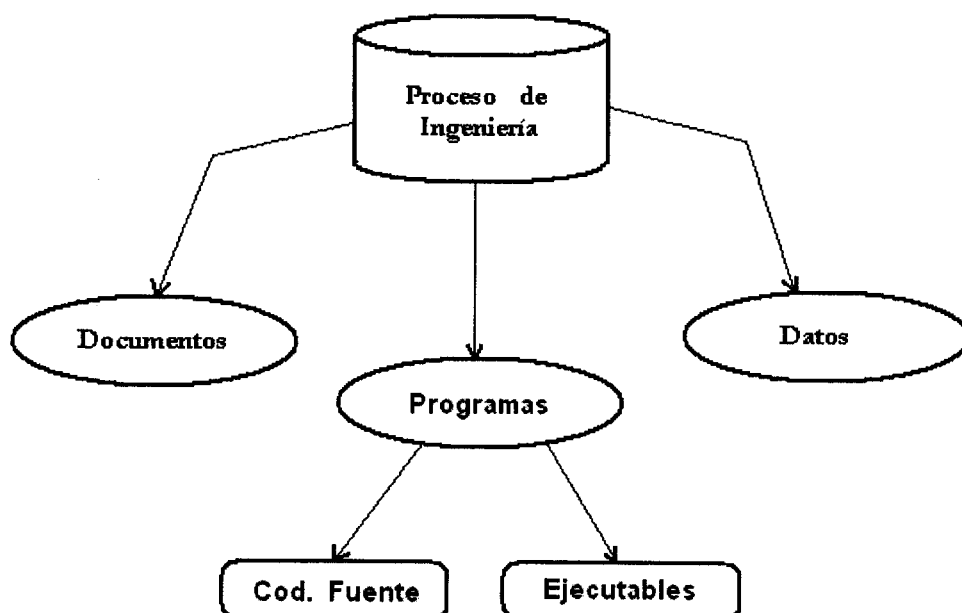


Ilustración 1 Esquema General

Los elementos de la configuración del software se organizan como objetos de configuración que deben ser catalogados por la base de datos del proyecto con un nombre único. Un elemento de la configuración del software tiene un nombre y atributos, y está conectado a otros objetos mediante relaciones.

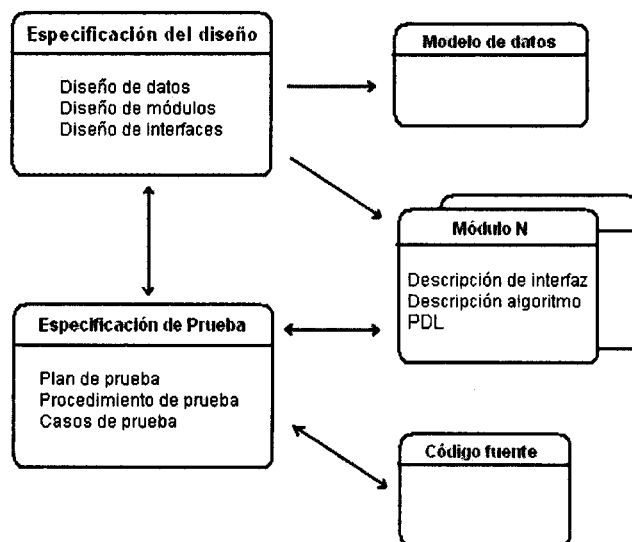


Ilustración 2 Objetos de configuración.

Como muestra la figura anterior cada elemento de configuración del software está relacionado con otro por lo que si se desea un cambio se necesita un estudio detallado sobre todos los objetos y sus relaciones.

En cualquier caso, para cada proyecto concreto se debe determinar qué se va a considerar como elemento de configuración.

Un elemento de configuración puede estar formado por otros elementos, de hecho el sistema en su conjunto como tal se puede determinar como un elemento de configuración. Cada elemento de configuración puede requerir su descomposición.

Existen dos tipos de elementos de configuración:

- Objetos Básicos.
- Objetos Compuestos.

Un objeto básico es una unidad de texto creada durante el análisis, diseño, codificación o prueba.

Un objeto compuesto es una colección de objetos básicos u objetos compuestos. Cada objeto tiene un conjunto de características que los identifican como únicos. El nombre del objeto es una cadena de caracteres que identifica al objeto sin ambigüedad. La descripción del objeto es una lista de elementos de datos que identifican:

El tipo de ECS (documento, programa, datos) que está representado por el objeto.

Un identificador del proyecto; y la información de la versión y/o el cambio.

El esquema de identificación de los objetos de software debe tener en cuenta que los objetos evolucionan a lo largo del proceso de ingeniería, por lo que se puede crear un *grafo de evolución*.

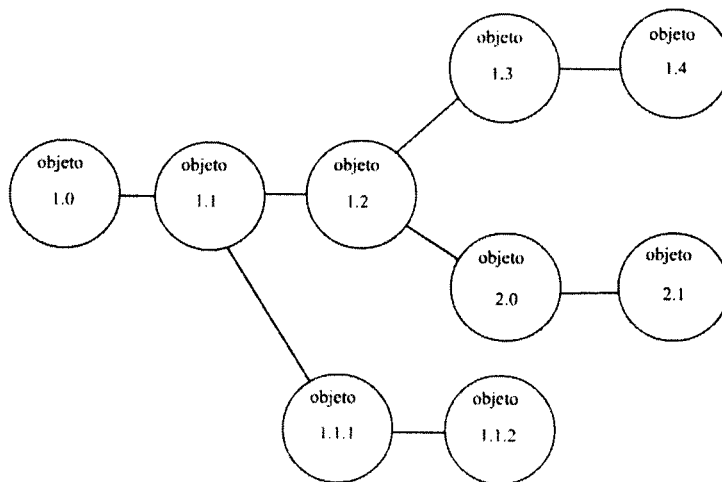


Ilustración 3 Esquema de identificación.

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.

Se puede decir que los elementos de configuración del software no es más que la información creada como resultado del proceso de Ingeniería del Software, que pueden ser código fuente, casos de prueba entre otras. Que se organizan como objetos de configuración, guardados y reconocidos por la base de datos como un elemento único, poseen nombre e identificador y están relacionados entre sí. Constituye la unidad de trabajo para la gestión de configuración del software.

1.3.3 LINEA BASE

Uno de los objetivos principales de la gestión de configuración del software va a ser el de gestionar los cambios que se producen en el sistema en desarrollo, para controlar los cambios sin impedir los cambios justificados se utiliza el concepto BASELINE o Línea Base.

La línea base se define como 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.

Una línea base se define como un elemento que ha sido revisado y aceptado, que va a servir como base para otros desarrollos posteriores y que solamente puede cambiarse a través de un proceso formal de control de cambios.

Una línea base es un concepto de gestión de configuración del software que nos ayuda a controlar los cambios sin impedir seriamente los cambios justificados. La IEEE 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.

Las líneas base más utilizadas son las que se encuentran en la figura siguiente:

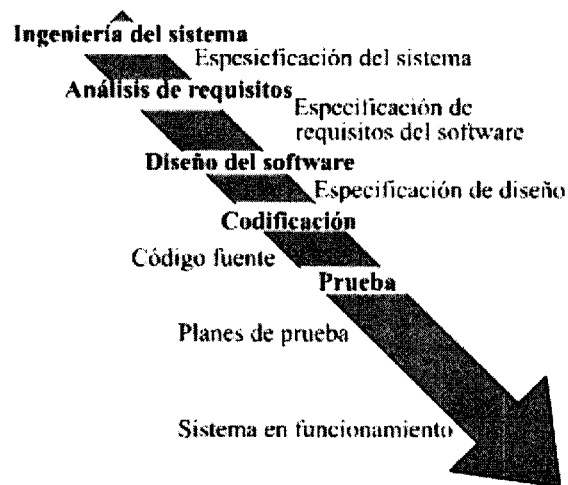


Ilustración 4 Líneas Base de un proyecto.

Otra forma de nombrar la línea base es de la forma siguiente: especificación o producto básico, el cual es el resultado de un consenso y que debe ser modificado solo en casos realmente necesarios y a partir de un proceso formal de control de cambio. Un proyecto tiene varias líneas bases.

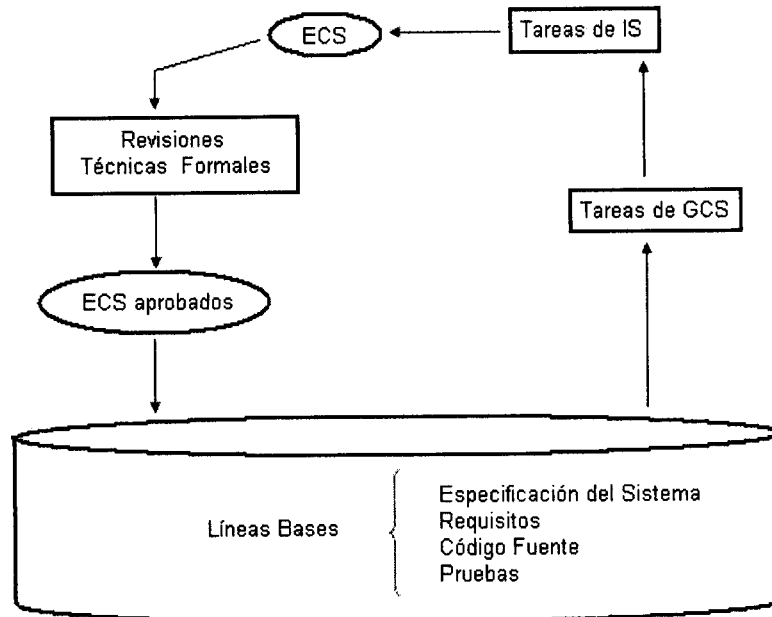


Ilustración 5 Proceso de líneas base.

1.3.4 BIBLIOTECAS DE SOFTWARE

Una biblioteca de software es una colección controlada de software y/o documentación relacionada cuyo objetivo es ayudar en el desarrollo, uso o mantenimiento del software. (Antonio, 2001)

En un proyecto se suelen establecer las siguientes bibliotecas de software:

-Biblioteca de Trabajo: Se establece al inicio del proyecto, y comprende el área de trabajo donde los analistas y diseñadores elaboran los documentos del proyecto y donde los programadores desarrollan el software, es decir, donde se realiza la codificación y pruebas unitarias. Una vez se han completado las revisiones o pruebas y el elemento de configuración en cuestión ha sido revisado y aprobado, se inicia la

transferencia del ECS a la biblioteca de soporte del proyecto. En esta biblioteca el control de cambios es informal.

-Biblioteca de soporte al proyecto: En esta biblioteca se almacenan los ECS aprobados y transferidos desde la Biblioteca de Trabajo o desde la Biblioteca de Integración. Cuando un elemento pasa a esta biblioteca, se encuentra sujeto a un control de cambios interno y semiformal.

-Biblioteca maestra: Se usa para almacenar ECS liberados para su entrega al cliente o distribución en el mercado. Los elementos en la biblioteca maestra se encuentran sujetos a un control de cambios formal y estricto. Y normalmente esta biblioteca tiene fuertes restricciones de acceso para escritura, aunque normalmente no los tiene para lectura. En esta biblioteca se almacenan las Releases del sistema.

-Repositorio de Software: Es la entidad en la que se archivan los ECS de un proyecto tras su cierre. Se transfieren a él desde la Biblioteca Maestra. Opcionalmente se puede identificar un segmento especial en el que se almacenarán los elementos reutilizables. Todo lo que se almacena en el repositorio debe estar adecuadamente identificado y catalogado, para facilitar su recuperación en caso de necesidad. Se supone que es un almacenamiento a largo plazo, por lo que puede ser de recuperación lenta (cinta). Es central y común a todos los proyectos, mientras que la biblioteca de Producción y la Maestra son individuales para cada proyecto.

-Biblioteca de Integración: Es de esta biblioteca de donde se toman los ECS para su integración en ECS de nivel superior del sistema.

-Biblioteca de Producción: Está compuesta por la Biblioteca de trabajo, la de integración y la Biblioteca de Soporte al Proyecto.

1.4 Los procesos de la Gestión de Configuración del Software.

Varios autores afirman que los procesos asociados a la gestión de configuración del software que aparecen en el estándar de la IEEE son suficientes para conseguir sus objetivos principales y son los siguientes:

1. Identificación de la Configuración.
2. Control de Cambios en la Configuración.
3. Generación de Informes de Estado.
4. Auditoria de la Configuración.

Por otra parte otro conjunto de autores después de una profunda investigación han planteado un conjunto de procesos que resulta nuevo en su composición:

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.

ISO sugiere de acuerdo a su criterio los siguientes procesos:

1. Identificación de la configuración.
2. Control de cambios en la configuración.
3. Informe del estado de la configuración.
4. Auditoria de la configuración.

Existe una similitud en cuanto a estos criterios y otros que se han dado por varios autores por lo que se puede llegar a la conclusión que se expone en la figura siguiente:

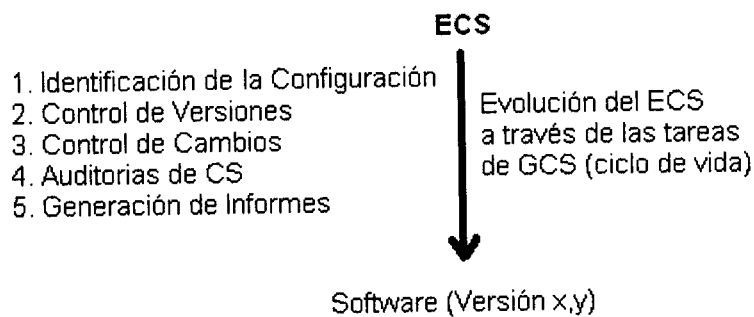


Ilustración 6 Actividades de la GCS.

1.4.1 Identificación de la configuración:

Este proceso consiste en identificar y asignar nombres significativos y consistentes a todos y cada uno de los elementos que forman parte del producto software, en cada fase de su desarrollo, es decir, a cada uno de los Elementos de Configuración del Software.

Este proceso engloba las siguientes actividades:

- Establecimiento de una jerarquía preliminar del producto software.
- Selección de los Elementos de Configuración.
- Definición de las relaciones en la configuración.
- Definición de un Esquema de Identificación.

- Definición y Establecimiento de líneas base.
- Definición y Establecimiento de bibliotecas de software.

A continuación se expone en que consiste cada una de estas actividades:

1.4.1.1 Establecimiento de una jerarquía preliminar del producto software.

Se trata de obtener una primera visión de la estructura y los elementos que tendrá el sistema software. Esta jerarquía facilitará la ejecución de otras actividades posteriores de gestión de configuración, como la selección de los elementos de configuración o la asignación de números de identificación a los documentos. Se realizará, opcionalmente, en los primeros compases del proyecto.

1.4.1.2 Selección de los Elementos de Configuración.

La selección de un número adecuado de elementos de configuración del software es muy importante. El tener demasiados puede provocar un número excesivamente elevado de especificaciones y documentos que al final resulta inmanejable. Otro problema es que mantener un elemento de configuración del software es costoso, ya que puede requerir:

- Especificación independiente
- Plan de pruebas
- Manuales independientes
- Revisión por parte del usuario
- Necesidad de aprobación de los cambios importantes por parte del usuario.
- Auditorías funcionales y físicas independientes.
- Número de identificación separado.
- Contabilidad de estado separada.

- Trazabilidad de su evolución.

1.4.1.3 Definición de las relaciones en la configuración.

Se puede considerar que los elementos de configuración son objetos, y están conectados con otros elementos de configuración del software mediante relaciones. Esta información ayudará al personal de gestión de configuración del software a comprender dónde se sitúa un elemento con respecto al resto. Hay que tener en cuenta que el personal de gestión de configuración del software suele ser externo al equipo de desarrollo y necesita este tipo de ayudas para poder asomarse a los productos y al proceso de desarrollo.

1.4.1.4 Definición de un Esquema de Identificación.

Permite archivar el elemento de configuración del software. Esta actividad brinda la información necesaria para utilizar el elemento en cualquier sistema que se esté realizando. Funciona como un etiquetado al elemento de configuración en cuestión de la siguiente manera:

- Número o código del Elemento de Configuración del Software.
- Nombre del Elemento de Configuración del Software.
- Descripción del Elemento de Configuración del Software.
- Autor/es del Elemento de Configuración del Software.
- Fecha de creación.
- Identificación del proyecto al que pertenece el Elemento de Configuración del Software.
- Identificación de la línea base a la que pertenece.
- Identificación de la fase y sub fase en la que se creó.
- Tipo de Elemento de Configuración (documento, programa, elemento físico (Cintas, discos, etc.)).
- Localización.

Después de tener bien archivado esta información de cada elemento de configuración del software se pasa a archivar lo que a versiones se refiere, debido a que un mismo elemento de configuración va a tener durante el ciclo de vida del software varias versiones para ello se utilizan los siguientes campos:

- Número de versión.
- Fecha de la versión.

1.4.1.5 Definición y Establecimiento de líneas base.

Una línea base se puede establecer de dos formas:

FÍSICAMENTE: Etiquetando cada elemento de configuración del software y almacenándolos en un archivo o biblioteca de proyecto.

LÓGICAMENTE: Publicando un documento de identificación de la configuración, que identifica el estado actual del producto en dicho punto del proceso de desarrollo. En cualquier caso, al comienzo de cada nuevo proyecto se debería determinar de forma precisa:

- cuál es el ciclo de vida a utilizar
- cuáles son sus fases
- cuáles son los productos de cada fase
- en qué puntos se van a establecer líneas base
- cuáles van a ser los elementos de configuración del software que contenga cada línea base.

1.4.1.6 Definición y Establecimiento de bibliotecas de software.

Una biblioteca de software es una colección controlada de software y/o documentación relacionada cuyo objetivo es ayudar en el desarrollo, uso o mantenimiento del software.

Facilitan la tarea de gestión de configuración del software, especialmente en cuanto al control de cambios y la contabilidad de estado.

1.4.2 Control de Versiones en la Configuración.

Antes de explicar a qué se dedica esta actividad de la gestión de configuración se debe de tomar como punto de partida el concepto de versión.

Una versión es 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. (Antonio, 2001)

A las distintas versiones que aparecen en el tiempo, según se va avanzando en el desarrollo de un elemento, se les suele llamar también REVISIONES.

Entre las distintas revisiones de un elemento de configuración se pueden establecer relaciones de sucesión temporal. Una representación posible para las diferentes revisiones de un elemento y sus relaciones de sucesión es mediante un GRAFO DE EVOLUCIÓN o grafo de revisión.

La manera más fácil de crear una nueva revisión de un elemento de configuración del software es realizar una modificación sobre la revisión más reciente. De esta manera las revisiones van formando una cadena, a la que se suele llamar CADENA DE REVISIÓN. Cada revisión en la cadena es una actualización, y viene a sustituir a la revisión anterior. Normalmente se trabajará sobre la última revisión de la cadena, que es la más actual, aunque las anteriores también deben ser accesibles.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Cada una de las revisiones de un elemento de configuración del software se debe poder identificar de manera única, siendo común utilizar un esquema numérico, donde cada nueva versión recibe un número sucesivo.

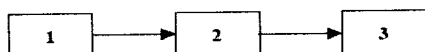


Ilustración 7 Esquema de revisiones.

La Gestión de Configuración debe permitir también especificar y gestionar distintas variantes de los elementos de configuración. Las variantes son versiones de un elemento de configuración que coexisten en un determinado momento y que se diferencian entre sí en ciertas características. Las variantes representan la necesidad de que un objeto satisfaga distintos requisitos al mismo tiempo. A diferencia con las revisiones, que son estrictamente secuenciales, y sólo existe una como revisión actual, las variantes se desarrollan en paralelo, y puede haber varias sobre las que se esté trabajando simultáneamente.

Una variante no reemplaza a otra, como ocurre con las revisiones, sino que abre un nuevo camino de desarrollo. Las variantes se reconocen fácilmente en el grafo de evolución, puesto que aparecen como una ramificación de éste.

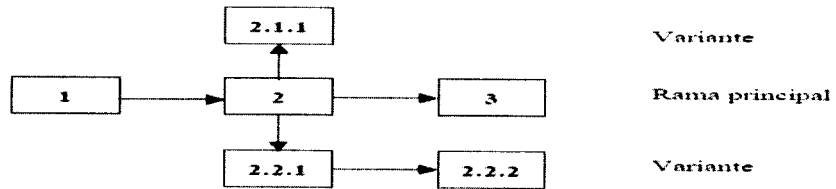
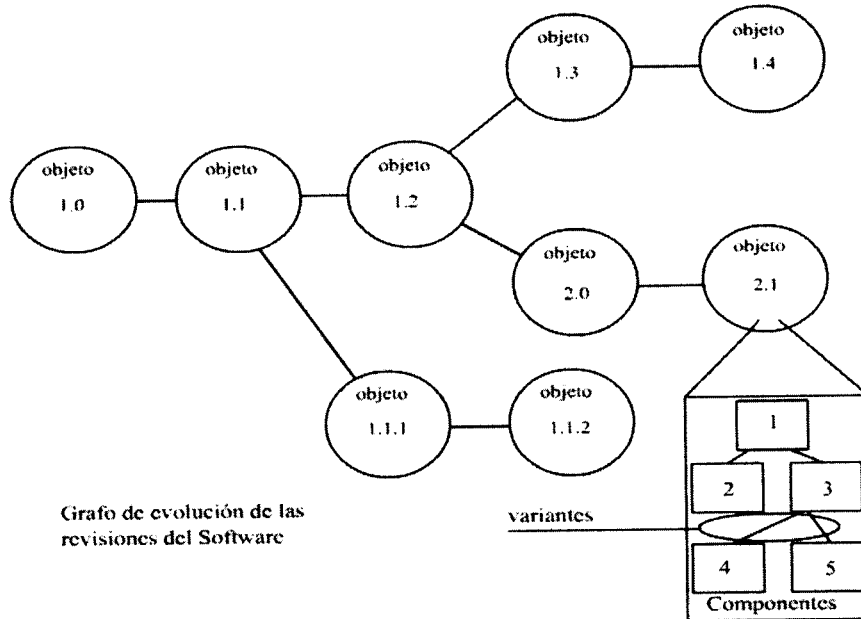


Ilustración 8 Esquema de variantes.

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.

La gestión de configuración permite a un usuario especificar configuraciones alternativas del sistema de software mediante la selección de las versiones adecuadas. Esto se puede gestionar asociando atributos a cada versión del software y permitiendo luego especificar y construir una configuración describiendo el conjunto de atributos deseado.

Los *atributos* pueden ser tan sencillos como un número específico de versión asociado a cada objeto o tan complejos como una cadena de variables lógicas que especifiquen tipos de cambios funcionales aplicados al sistema.



Grafo de evolución de las revisiones del Software

Ilustración 9 Grafo de evolución.

Para construir la variante adecuada de una determinada versión de un programa, a cada componente se le asigna una *tupla de atributos*. A cada variante se le asigna uno o más atributos. Otra forma de establecer los conceptos de la relación entre componentes, variantes y versiones es representarlas como un fondo de objetos.

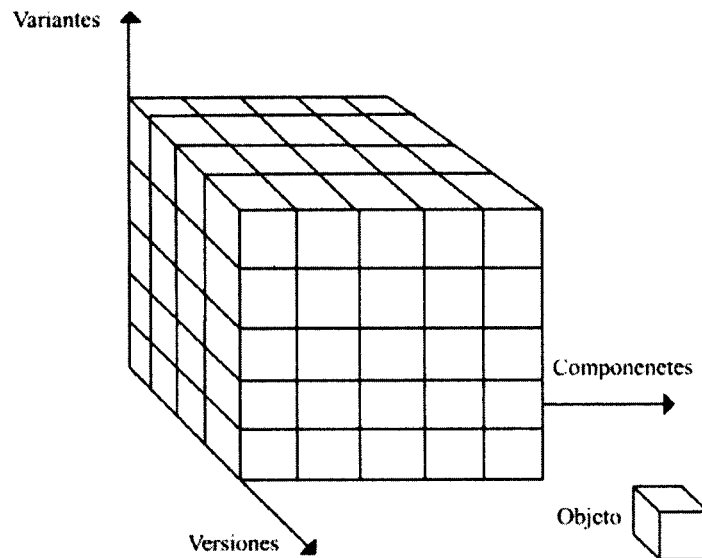


Ilustración 10 Sistema de control de versiones.

Un sistema de control de versiones debe proporcionar:

- Mecanismo de almacenaje de los elementos que deba gestionar (ej. archivos de texto, imágenes, documentación).
- Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).
- Aunque no es estrictamente necesario, suele ser muy útil la generación de informes con los cambios introducidos entre dos versiones, informes de estado, marcado con nombre identificativo de la versión de un conjunto de ficheros, etcétera.

La principal clasificación que se puede establecer está basada en el almacenamiento del código:

Centralizados: existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a cambio de reducir la potencia y flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable. Algunos ejemplos son CVS y Subversión .

Distribuidos: se aumenta la capacidad de decisión distribuida. Esto da más flexibilidad pero puede dificultar bastante la sincronización. Ejemplos: GIT y GNU Arch.

Todos los sistemas de control de versiones se basan en disponer de un repositorio que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados.

Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su utilización. Es posible duplicar la última versión o cualquier versión almacenada en el historial. Este proceso se suele conocer como *check out* o *desproteger*. Para modificar la copia local existen dos semánticas básicas:

Exclusivos: para poder realizar un cambio es necesario marcar en el repositorio el elemento que se desea modificar y el sistema se encargará de impedir que otro usuario pueda modificar dicho elemento.

Colaborativos: en el que cada usuario se descarga la copia la modifica y el sistema automáticamente mezcla las diversas modificaciones. El principal problema es la posible aparición de conflictos que deban ser solucionados manualmente o las posibles inconsistencias que surjan al modificar el mismo fichero por varias personas no coordinadas. Además, esta semántica no es apropiada para ficheros binarios.

Tras realizar la modificación es necesario actualizar el repositorio con los cambios realizados. Habitualmente este proceso se denomina *commit*, *check in* o *proteger*.

1.4.3 Control de Cambios en la Configuración.

Los cambios son normales en la evolución de un sistema. Un control de cambios apropiado es la esencia de una buena administración de la configuración. El control de cambios de la configuración del software evalúa cambios propuestos a elementos de configuración y coordina la implementación de los cambios aprobados.

Es la actividad de Gestión de Configuración más importante y su objetivo es proporcionar un mecanismo riguroso para controlar los cambios, partiendo de la base de que los cambios se van a producir. Normalmente combina procedimientos humanos y el uso de herramientas automáticas.

El proceso de control de cambios analiza descripciones de problemas, decide cuáles (si es necesario) serán cambiados, los retira de la librería maestra para ser cambiados, y devuelve los cambiados a la librería maestra para almacenarlos. EL proceso también entrega descripciones de los cambios efectuados.

Los cambios deben ser verificados antes de que los siguientes cambios sean implementados. Si se efectúa un grupo de cambios, cada cambio debe ser verificable por separado.

Proceso Formal de Revisión y Cambios en los ECS.

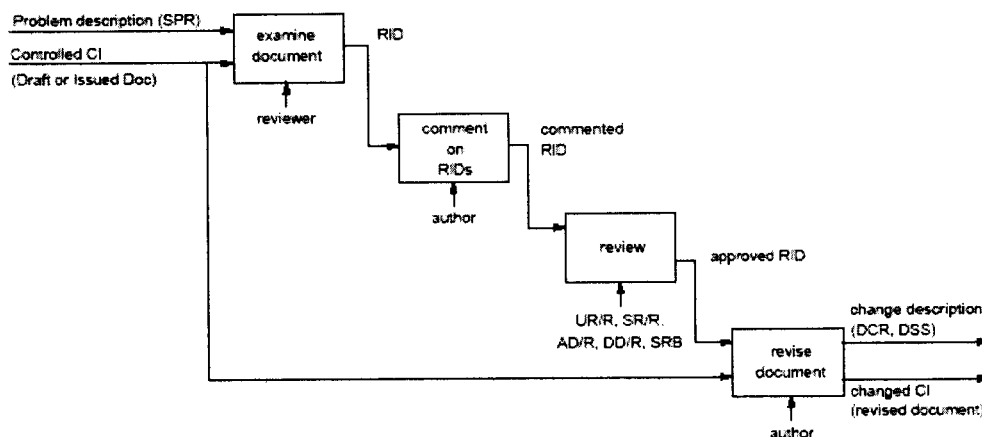


Ilustración 11 Proceso formal de revisión y cambios.

Dentro del proceso de control de cambios no hay ningún estándar que diga como se debe llevar a cabo los cambios dentro de un proyecto como tal, pero a la hora de llevar el control de cambios de manera formal si existen. A continuación se les muestra las etapas típicas dentro de un proceso formal, el cual habría que seguir para hacer un cambio sobre una línea base.

-Iniciación del Cambio: se presenta una solicitud de cambio, que puede venir provocada por un problema que se ha detectado o por un cambio en los requisitos.

-Clasificación y registro de la solicitud de cambio.

- Aprobación o rechazo inicial de la solicitud de cambio. De ello suele ser responsable el *Comité de Control de Cambios*.

- Evaluación de la solicitud de cambio, si ha sido aprobada, para calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y el coste estimado del cambio. Como resultado se obtiene un *Informe de Cambio*.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

- Se presenta el Informe de Cambio al Comité de Control de Cambios. Si se considera que el cambio es beneficioso se genera una *Orden de Cambio* (también llamada Orden de Cambio de Ingeniería), que describe el cambio a realizar, las restricciones que se deben respetar y los criterios de revisión y de auditoría. Esta Orden de Cambio es asignada a alguno de los ingenieros de software para que se encargue de llevarlo a cabo. En este momento, el objeto a cambiar se da de baja en la Biblioteca de Soporte al Proyecto.

- Se realiza el cambio, entrando en un proceso de seguimiento y control.

- Una vez finalizado el cambio, se certifica, mediante una revisión, que se ha efectuado correctamente el cambio y con ello se ha corregido el problema detectado o bien se han satisfecho los requisitos modificados. El objeto se devuelve a la Biblioteca de Soporte al Proyecto.

- Se notifica el resultado al originador del cambio.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

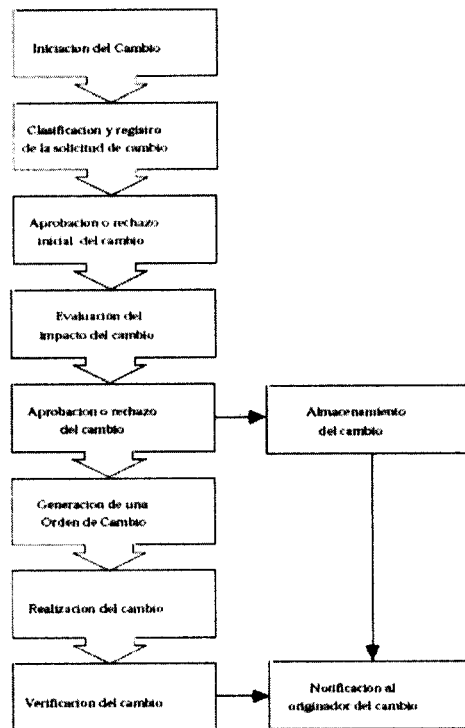


Ilustración 12 Proceso de control de cambios.

Las tareas principales a desarrollar dentro del control de cambios se pueden resumir en:

- 1-Actualizar la copia local.
- 2-Generar un mecanismo para llevar el control de cambios.
- 3-Protocolos de actuación para personas.
- 4-Herramientas de automatización de cambios.
- 5- Aplicación de políticas de gestión de cambios informales salvo en el proceso de conversión de ECO a línea base.
- 6-Procesos a controlar.

7-Baja del objeto en la biblioteca.

8-Alta del objeto en la biblioteca.

9-Aplicación de la política de versiones sobre los objetos (Control de Versiones).

10-Elementos a definir:

10.1-Control de acceso al repositorio (seguridad y permisos).

10.2-Control de sincronización. (Concurrencia de acceso al repositorio)

10.3-Orden de Cambio.

1.4.4 Informes de Estado.

Es la tercera de las cuatro tareas básicas de Gestión de Configuración. A veces también denominada Contabilidad de Estado. El objetivo es mantener a los usuarios, a los gestores y a los desarrolladores al tanto del estado de la configuración y su evolución. En definitiva, pretende dar respuesta a la pregunta "¿Qué ocurrió?", y también a la pregunta "¿Cuándo ocurrió?".

Ayuda también a mejorar los problemas de comunicación entre los participantes en un proyecto.

Esto se va a conseguir registrando toda la información necesaria acerca de lo que va ocurriendo y generando los informes necesarios. Esta tarea implica, por tanto, la realización de tres actividades básicas:

- Captura de la información
- Almacenamiento de la información
- Generación de informes



Ilustración 13 Procesos de informes de estado.

Importancia de la tarea de Informes de Estado:

-Para mantener la continuidad del proyecto. Se trata de permitir que el proyecto siga adelante cuando, por ejemplo, el jefe de proyecto deja la empresa.

-Para evitar la duplicidad del trabajo. Si no se guarda información acerca de lo que ya se ha hecho, se puede estar repitiendo el trabajo ya hecho.

-Para evitar caer en los mismos errores una y otra vez.

-Para ser capaces de repetir aquello que salió bien.

-Puede ayudar a encontrar las causas de un fallo.

La generación de informes de estado de la configuración es una tarea de GCS que responde a las siguientes preguntas:

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

La generación de informes de estado de la configuración desempeña un papel vital en el éxito del proyecto de desarrollo de software. Cuando aparece involucrada mucha gente es muy fácil que no exista

una buena comunicación. Pueden darse errores entre las personas desarrolladoras del software. El IEC ayuda a eliminar esos problemas, mejorando la comunicación entre todas las personas involucradas.

1.4.5 Auditoría de la Configuración.

Una auditoría es una verificación independiente de un trabajo o del resultado de un trabajo o grupo de trabajos para evaluar su conformidad respecto de especificaciones, estándares, acuerdos contractuales u otros criterios. La auditoría de la Configuración es la forma de comprobar que efectivamente el producto que se está construyendo es lo que pretende ser.

Esta función a veces se considera fuera de la Gestión de Configuración y dentro de la Garantía de Calidad. También tiene relación con las actividades de Validación y Verificación. En realidad es un punto de intersección entre todas ellas.

Es la actividad de GCS más costosa. Requiere de personal experimentado, y con un gran conocimiento del proceso de desarrollo. Sin embargo, debe ser realizada por personal ajeno al equipo de desarrollo técnico para mantener la objetividad de la auditoría.

¿Cómo podemos asegurar que el cambio se ha implementado correctamente?

-Revisiones técnicas formales.

-Auditorías de configuración del software.

Las revisiones técnicas formales se centran en la corrección técnica del elemento de configuración que ha sido modificado. Los revisores evalúan el elemento de configuración para determinar la consistencia con otros elementos de configuración, las omisiones o los posibles efectos secundarios.

Una auditoría de configuración del software complementa la revisión técnica formal al comprobar características que generalmente no tiene en cuenta la revisión. La auditoría se plantea y responde con las siguientes preguntas:

- ¿Se ha hecho el cambio especificado en la OCI? ¿Se han incorporado modificaciones adicionales?

- ¿Se ha llevado a cabo una revisión técnica formal para evaluar la corrección técnica?

- ¿Se han seguido adecuadamente los estándares de ingeniería de software?
- ¿Se han "recalcado" los cambios en el elemento de configuración del software? ¿Se han especificado la fecha del cambio y el autor? ¿Reflejan los cambios los atributos del objeto de configuración?
- ¿Se han seguido procedimientos para señalar el cambio, registrarlo y divulgarlo?
- ¿Se han actualizado adecuadamente todos los elementos de configuración del software relacionados?

La auditoria de configuración es realizada por el grupo de calidad del software.

1.5 Herramientas para la GCS

En entornos de desarrollo complejos, especialmente, si se integran diversas herramientas de ingeniería de software, se hace imprescindible la incorporación de una herramienta capaz de gestionar la configuración de los sistemas.

Este tipo de herramienta ofrece cuatro tipos de capacidades:

Control de versiones, es decir, la capacidad de proporcionar almacenamiento y acceso controlado a los datos, así como de registrar los cambios sobre los mismos y poder recuperar versiones anteriores.

Construcción, que consiste en automatizar las tareas asociadas a la compilación y enlace de un sistema con el fin de generar los ejecutables.

Trazabilidad de requisitos y análisis de impacto, que permiten rastrear un requisito hasta su implementación y conocer los elementos del sistema que se ven afectados ante un cambio.

Capacidades generales, apartado en el que se puede englobar: gestión del proceso, capacidades de trabajo en grupo, interfaces con otras herramientas, generación de informes, consultas, etc.

Las propuestas más robustas las ofrecen soluciones propietarias, con un alto precio en licencias. Es por esto, que como alternativa, los equipos de desarrollo deben seleccionar un conjunto de herramientas libres que por separado brinden las funcionalidades de las herramientas privativas.

Como ejemplo de un conjunto de herramientas representativas del mundo del software propietario se pueden citar: IBM Rational ClearCase, Visual Studio TeamSystem, Plastic SCM1, Telelogic, Surround SCM. Mientras que CVS1, Subversion (SVN), Darcs y Bit son ejemplo de herramientas libres.

1.6 Conclusiones

La finalidad de la Gestión de Configuración del Software es conocer la estructura de procesos y herramientas para aplicar dentro de la construcción del software, ellas nos ayudan a controlar los cambios y las versiones. Es importante considerar ciertas modificaciones que pueden ocurrirle al software dentro de todo el proceso de ingeniería para asegurar su control y calidad.

El control de versiones es un conjunto de procedimientos y herramientas que se utilizan para gestionar el uso de los objetos. El control de cambios es más actividad procedimental que asegura la calidad y la consistencia a medida que se realizan cambios en los objetos de la configuración. El proceso de control de cambios comienza con una petición de cambio, lleva a una decisión de proseguir o no con el cambio y culmina con una actualización controlada del ECS que se ha de cambiar. La auditoria de configuración es una actividad de aseguramiento a la calidad durante la realización de los cambios. Los informes de estado proporcionan información sobre cada cambio a aquellos que tienen que estar informados.

Capítulo 2: Caracterización del DDS y Selección de Herramientas.

2.1 Introducción.

En este capítulo se caracteriza el Departamento de Desarrollo de Software con el objetivo de poder seleccionar las herramientas factibles para el trabajo de control de versiones. El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente, sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, entre otras opciones. Los sistemas de control de versiones se pueden manejar de forma manual pero es aconsejable utilizar herramientas que faciliten esta gestión, aquí es donde juega un papel importante dentro del proceso de desarrollo de software los sistemas de control de versiones, por lo que en este capítulo se hará un estudio profundo sobre cuál o cuáles son los sistemas de control de versiones más factibles a utilizar.

2.2 Caracterización del Departamento de Desarrollo de Software.

El Departamento de Desarrollo de Software (DDS) se encuentra en formación por lo que puede variar el orden y la complejidad de objetivos y funciones dentro del mismo, en principio se propone su estructura como una organización mixta compuesta por áreas generales para el desarrollo del software (análisis, desarrollo, soporte, calidad) y los equipos de proyectos (formados por integrantes de las áreas mencionadas).

El DDS es un entorno complejo para el desarrollo de software, está en fase de creación y organización, contendrá polos de desarrollo en provincias que se subordinan a él, hasta el momento están creados en:

Villa Clara

Santiago de Cuba

Holguín

Las Tunas

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Esta característica hace que se requiera de un proceso de desarrollo ordenado, conciso y eficiente que garantice crear software rápido y con calidad.

Se propone que el DDS esté designado para ejecutar las funciones de regulación y supervisión técnica de la actividad de desarrollo de software que se realicen para el MININT, la dirección de la actividad productiva interna y de las actividades de colaboración y coordinación en la actividad productiva que se desarrollan externamente; estableciendo practicas en su estrategia organizativa y tecnológica que garantice el alineamiento de las tecnologías de información con las actividades vitales de la institución, de manera que se posibilite la innovación y desarrollo de sus procesos.

Entre los objetivos que están propuestos para su mejor organización interna se encuentran:

1-Acelerar el desarrollo del software en función de la Seguridad y el Orden Interior, adoptando estrategias organizativas y tecnológicas que permitan optimizar procesos, simplificar la creación y garantizar la seguridad.

2-Obtener productos más efectivos, seguros y fáciles de mantener, adoptando tecnologías, prácticas y herramientas que permitan reducir los ciclos de desarrollo, insertando la seguridad en todo el ciclo de vida.

3-Promover y aplicar herramientas tecnológicas que permitan adaptabilidad a los cambios, el desarrollo a partir del modelado, monitorización y control, de manera que se obtengan sistemas, aplicaciones y servicios gestionables y productivos.

4-Alcanzar estándares de calidad en el proceso de desarrollo del software adoptando progresivamente, un modelo de mejora evolutivo, utilizando la formación previa y la experiencia de expertos.

Después de observar el DDS se pudo determinar lo siguiente:

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

No existe un repositorio de software.

Carencia de personal especializado y el que existe es mínimo, pero este personal si cuenta con la experiencia de los procesos propios del MININT.

El control del software producido es bajo, así como de los que están en producción.

El posible re uso se dificulta y no es adecuado pues no están documentados los componentes que existen.

La capa intermedia de aplicación está desarrollada con tecnología Microsoft y base de datos ORACLE en todo el MININT, siendo esto un gran volumen de aplicaciones que son imprescindibles para el trabajo del mismo. Otras aplicaciones están desarrolladas totalmente con las herramientas propias de ORACLE. El traspaso de estas aplicaciones a código abierto sería un proceso lento y engorroso y no está previsto en la organización.

Necesidad de sincronizar y ordenar las actividades de desarrollo entre el DDS y los polos de desarrollo y dentro del DDS entre los equipos de proyectos.

Esta investigación está centrada en el área de soporte, muchos intelectuales que se han adentrado en el tema de la Gestión de Configuración del Software abordan que la misma se debe controlar dentro del área de soporte aunque no quitan que se podría encontrar también en el área de calidad, debido a que dentro de un proyecto de software ayuda en cierta medida en este aspecto.

El DDS cuenta con herramientas para el desarrollo de software y se está haciendo un estudio profundo sobre que tendencia aplicar dentro del mismo, por el momento cuenta con las siguientes interfaces de desarrollo:

1-Visual Studio.Net(C#)

2-Plataforma Oracle

2.3 Herramientas para el Control de Versiones y Cambios.

Uno de los elementos claves dentro de la Gestión de Configuración del Software (GCS) es la selección de sus herramientas para el control de versiones y la gestión de cambios.

Para lograr controlar todas las actividades de la GCS, es decir, identificar y definir los elementos del sistema, controlar sus cambios, registrar y reportar el estado de los elementos se necesitan las herramientas más óptimas que se adapten a las características exactas.

En este capítulo se dará una panorámica general de las herramientas de control de versiones y gestión de cambios más utilizadas en el mundo y después de una profunda investigación y comparación se obtendrá la más óptima para las actividades de la GCS dentro del DDS.

2.3.1 Comparación de Sistemas de Control de Versiones.

Existen muchos aspectos para comparar las herramientas de control de versiones entre los que se encuentran:

-Licencia.

-Operaciones en el Repositorio.

-Mejoras.

-Estado Técnico.

-Interfaces de Usuarios.

LICENCIA

Nuestro país se encuentra bloqueado varios años, todo lo que se ha hecho en cuanto a desarrollo e informatización de la sociedad cubana es gracias al esfuerzo de la revolución y su pueblo. Para el desarrollo del software dependemos de muchos aspectos por ejemplo, el uso gratuito del software y que sea de código abierto. Se tendrá en cuenta el tipo de software que se construirá dentro del DDS (libre/propietario).

CVS - GNU GPL (Código Abierto).

AccuRev – Propietario, licenciado nombrado-usuario.

Aegis - GNU GPL (Código Abierto).

Arch - GNU GPL (Código Abierto).

Bitkeeper – Propietario, pagar por licencia de uso con una opción de licencia sin gastos reveladores del software de abrir-fuente.

Clearcase – Propietario, soportado por la licencia flotante.

CMSynergy – Tasa negociable con el vendedor

Co-Op – Propietario, llave corta del texto.

Darcs - GNU GPL (Código Abierto).

Mercurial - GNU GPL (Código Abierto).

Monotone - GNU GPL (Código Abierto).

OpenCM - GNU GPL (Código Abierto), pero moviéndose pronto al DEB o completo.

Perforce - Propietario, binario solamente.

PureCM - Propietario, binario solamente.

Subversion – Licencia APACHE/BSD-Style (Código Abierto).

Superversion – GNU GPL (Código Abierto).

SVK - Licencia Perl (Código Abierto).

Vesta - GNU LGPL (Código Abierto).

Visual SourceSafe – El VSS envía con MSDN y puede también ser comprado independiente o con otras herramientas.

A partir de ahora solo se trataran las herramientas que pueden ser de mayor importancia para el trabajo de control de versiones dentro del DDS.

Otro punto de vital importancia es el trabajo con el repositorio donde se almacena todos los productos de un proyecto de software

OPERACIONES EN EL REPOSITORIO.

ATOMIC COMMITS:

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Esta es una operación dentro del repositorio que significa que si se está trabajando en una operación en el repositorio y se interrumpe a mitad, el repositorio no será dejado en un estado inconsistente.

CVS - Los cvs commits no son atómicos.

Aegis - los commits son atómicos.

Arch - los commits son atómicos.

Darcs - los commits son atómicos.

Mercurial - los commits son atómicos.

Monotone - los commits son atómicos.

OpenCM - los commits son atómicos.

Subversion - los commits son atómicos.

Superversion - los commits son atómicos.

SVK - los commits son atómicos.

Vesta - los commits son atómicos.

Archivos o Directorios se mueven o renombran:

Esto significa que el sistema permite el movimiento de un archivo o directorio a una localización diferente manteniendo el historial de archivo.

CVS - No permite el renombrar y un re nombramiento manual dividiría el historial en dos.

Aegis - Soporta re nombramiento.

Arch - Soporta re nombramiento.

Darcs - Soporta re nombramiento.

Mercurial - Soporta re nombramiento.

Monotone - Soporta re nombramiento.

OpenCM - Soporta re nombramiento.

Subversion - Soporta re nombramiento.

Superversion – No soporta re nombramiento.

SVK - Soporta re nombramiento.

Vesta - Soporta re nombramiento.

Copias de Archivos y Directorios:

Esto significa que se puede hacer una copia a un archivo o directorio a un lugar diferente en el repositorio manteniendo el historial.

CVS - No soporta copias.

Aegis - No soporta copias.

Arch - No soporta la estructura de copia de archivos o directorios.

Darcs - No soporta la estructura de copia de archivos o directorios.

Mercurial - Si soporta copias.

Monotone - Si soporta copias.

OpenCM - No soporta copias.

Subversion - Si y es una operación muy barata (0 (1)), esto también se utiliza para ramificar.

Superversion - No soporta copias.

SVK - Si soporta copias al igual que Subversion.

Vesta - Si soporta copias.

Replica del repositorio remoto:

Esto significa que el software permite reproducir un repositorio remoto para conseguir una copia funcionalmente equivalente en el sistema local. Eso se debe hacer sin ningún acceso especial al servidor remoto a excepción del acceso normal al repositorio.

CVS - Indirectamente, usando Cvsup.

Aegis - SI.

Arch - SI.

Darcs - SI.

Mercurial - SI.

Monotone - SI.

OpenCM - NO.

Subversion - Indirectamente.

Superversion - SI.

SVK - SI.

Vesta - SI, la réplica es una parte fundamental del diseño.

Propagar cambios a los repositorios padres:

Esto significa que puede el sistema propagar cambios de un repositorio a otro.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE
DESARROLLO DE SOFTWARE.

CVS - NO.

Aegis - SI.

Arch - SI.

Darcs - SI.

Mercurial - SI.

Monotone - SI.

OpenCM - No.

Subversion - SI.

Superversion - NO.

SVK - SI.

Vesta - SI.

Permisos del Repositorio:

Esto significa que es posible definir permisos para el acceso a diferentes partes de un repositorio remoto o es accesible abiertamente para todos.

CVS - Limitado.

Aegis - SI.

Arch - SI.

Darcs - NO.

Mercurial - SI.

Monotone - SI.

OpenCM - Los permisos se definen sobre una base del árbol de rama.

Subversion - SI.

Superversion - NO.

SVK - SI.

Vesta - SI.

Soportes de Cambios:

Esto significa que el repositorio puede permitir soportes de cambios. Estos son una manera de agrupar un número de modificaciones que sean relevantes para cada uno en un paquete atómico, que puede ser cancelado o propagado según se necesite.

CVS - NO.

Aegis - SI.

Arch - SI.

Darcs - SI.

Mercurial - SI.

Monotone - SI.

OpenCM - SI.

Subversion - Soporte Parcial.

Superversion - Soporte Parcial.

SVK – Igual a Subversion.

Vesta – No exactamente.

Siguiendo el historial de la línea-sabia del archivo:

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Esto significa que si el sistema de control de versiones tiene una opción para seguir el historial del archivo línea por línea. Es decir: ¿para cada línea muestra en qué revisión fue cambiada más eficientemente y por quién?

CVS - SI.

Aegis - SI.

Arch - NO.

Darcs - SI.

Mercurial - SI.

Monotone - SI.

OpenCM - Se desconoce, probablemente no.

Subversion - SI.

Superversion - NO.

SVK - SI.

Vesta - NO.

Capacidad de trabajar solo en un directorio del repositorio:

Esto significa que el sistema de control de versiones puede comprobar solamente un directorio del repositorio. O restringir los registros a un directorio solamente.

CVS - SI.

Aegis - NO.

Arch - Es posible.

Darcs - Es posible.

Mercurial - Es posible.

Monotone - Es posible.

OpenCM - NO.

Subversion - SI.

Superversion - NO.

SVK - SI.

Vesta - SI y NO.

Seguimiento de cambios Uncommitted:

Esto significa que si el software tiene la habilidad de seguir los cambios en la copia de trabajo que no ha sido confiada al repositorio.

CVS - SI.

Aegis - SI.

Arch - SI.

Darcs - SI.

Mercurial - SI.

Monotone - SI.

OpenCM - SI.

Subversion - SI.

Superversion - SI.

SVK - SI.

Vesta - SI.

Documentación:

Esto significa que tan buena es la documentación del sistema. Cuán fácil es utilizar.

CVS – Excelente.

Aegis – Medio.

Arch – Medio.

Darcs – Bueno.

Mercurial – Muy Bueno.

Monotone – Bueno.

OpenCM – Bien Documentado.

Subversion – Muy Bueno.

Superversion – Bastante Pobre.

SVK – Relativamente Pobres.

Vesta – Absolutamente a fondo.

Facilidad de despliegue:

Esto significa cuán fácil es desplegar el sistema. ¿Qué son las dependencias y como pueden ser eliminadas?

CVS - Bueno.

Aegis - No es muy portable.

Arch - Excelente.

Darcs - Muy Bueno.

Mercurial - Excelente.

Monotone - Excelente.

OpenCM - Muy Bueno.

Subversion - Los servicios de Subversion requieren instalación de un modulo apache 2 o su propio servidor propietario. El cliente requiere solamente la lógica Subversion específica y la biblioteca de neón de WEBDAV .la instalación de componentes es absolutamente directa, pero requerirá un cierto trabajo, Subversion no viene pre embalado para su sistema.

Superversion - Si Java 1.4 está instalado, el despliegue de Superversion toma generalmente dos tecleos.

SVK - Además de instalar Subversion, los usuarios requieren instalar los atascamientos del Perl de Subversion y algunos módulos de CPAN.

Vesta - Más o menos.

Paquete de comandos:

Esto significa que es un paquete de comandos. Cuan compatibles es con los comandos del CVS.

CVS - Un sistema de comando simple que incluye tres comandos más de uso general (los cvs confía, actualización de los cvs y comprobación de los cvs) y otros.

Aegis - Un sistema de comando complejo que implica muchas operaciones apenas para conseguir un comienzo. No es compatible con CVS (no obstante la ayuda para tales operaciones básicas es contemplada) observar que la tutela es un sistema de gerencia de la configuración del software y no apenas un sistema de control simple de la versión, que puede justificar esta complejidad adicional.

Arch - Muchos comandos son compatibles con CVS o BitKeeper. Sin embargo, hay muchos otros comandos para él hacia diversas aplicaciones.

Darcs - El sistema de comando es bastante compacto y los comandos de la base son fáciles de entender. Sigue a CVS en algunos lugares, pero puesto que el modelo es diferente la mayoría de los comandos son únicos.

Mercurial - Intenta seguir las convenciones de CVS, pero se desvía donde hay un diverso diseño.

Monotone - Intenta seguir las convenciones de CVS, pero se desvía donde hay un diverso diseño.

OpenCM - A CVS-como el sistema de comandos que es familiar a los usuarios existentes de CVS.

Subversion - A CVS como el sistema de comando que es fácil de conseguir.

Superversion - Hay poca necesidad de memorizar un sistema de comandos porque todas las acciones ocurren en un GUI. Una parte de la terminología utilizada se pide prestada de CVS.

SVK - A CVS-como el sistema de comando que es fácil de conseguir.

Vesta - El sistema de comandos no tiene relación con CVS. La mayor parte del tiempo, los usuarios utilizan cerca de 5 comandos. Pocos necesitan saber más de 20 comandos.

Soporte de Red:

Esto significa cuan buena es la integración de la red en el sistema. Compatibilidad con infraestructuras y protocolos existentes.

CVS – Bueno.

Aegis – Pobre.

Arch – Excelente.

Darcs – Bueno.

Mercurial – Excelente.

Monotone – Bueno.

OpenCM – Bueno.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Subversion – Muy Bueno.

Superversion – Bueno.

SVK – Muy Bueno.

Vesta – El establecimiento de una red es inherente al sistema, el depósito exporta una interfaz del NFS y otra del RPC.

Portabilidad:

Esto significa cuan portable es el sistema de control de versiones para diferentes sistemas operativos, arquitecturas de sistemas y otros tipos.

CVS – Bueno.

Aegis – Medio.

Arch – Bueno.

Darcs – Muy Bueno.

Mercurial – Excelente.

Monotone – Excelente.

OpenCM – Bueno.

Subversion – Excelente.

Superversion – Excelente.

SVK – Bueno.

Vista – Bueno.

Interfaz Web:

Esto significa que si el sistema tiene una interfaz basada en www que se pueda utilizar para hojear el árbol y las diferentes revisiones de los archivos, etc.

CVS - SI.

Aegis - SI.

Arch - Existen archzoom que son trabajos en marcha.

Darcs - darcs.cgi se incluye en la distribución.

Mercurial - SI.

Monotone - NO.

OpenCM - NO.

Subversion - SI.

Superversion - NO.

SVK - SI.

Vesta - SI.

Disponibilidad de interfaces gráficas de usuarios:

Esto significa que cual es la disponibilidad de las interfaces graficas de usuario para el sistema. Cuantos clientes GUI están presentes para esto.

CVS – Muy Bueno.

Aegis – Hay tkaegis.

Arch – Hay tlator, octopy, y posiblemente otros en el desarrollo.

Darcs – No se habla de ninguno.

Mercurial – Mas fácil.

Monotone – No hay GUIs disponibles.

OpenCM – No hay GUIs disponibles.

Subversion – Muy Bueno.

Superversion – Está integrado.

SVK – No hay GUIs disponibles.

Vesta - No hay GUIs disponibles.

Después de analizar los sistemas de controles más óptimos se llegó a la conclusión que "Subversion" es un sistema de control de versiones que tiene varios de los requisitos para utilizarse dentro del DDS para futuros proyectos de desarrollo de software.

Subversion es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. 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. 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. En el anexo 1 se exponen los principales comandos de esta herramienta.

Ventajas de Subversion.

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- 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 sólo 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.
- Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).
- 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, etc.).

Desventajas de Subversion.

- 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).

Clientes

- TortoiseSVN. Provee integración con el explorador de Windows. Es la interfaz más popular en este sistema operativo. En el anexo 2 se exponen las imágenes de este cliente.
- Subclipse. "Plugin" que integra Subversion al entorno Eclipse.
- Subversive. "Plugin" alternativo para Eclipse.
- ViewVC. Interfaz web, que también trabaja delante de CVS.
- Para Mac, pueden emplearse los interfaces SvnX, RapidSVN y Zigversion.

- KDESvn. Provee integración con el escritorio KDE, muy parecido en apariencia/funcionamiento/características a TortoiseSVN.

Existe otra herramienta que no fue planteada en la comparación de las operaciones con el repositorio pero que ha resultado eficaz y fácil de utilizar para el trabajo de control de versiones, "Visual SourceSafe". Se aconseja que se utilice cuando el proyecto esta soportado bajo el Sistema Operativo Windows y si el proyecto es relativamente pequeño.

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.

SourceSafe fue creado por la compañía americana ONE TREE Software a principios de los años noventa. Su primera versión fue la 3.1 coincidiendo con el mismo número de versión del sistema operativo Windows. Microsoft adquirió los derechos sobre SourceSafe que era un programa de 16 bits y lo liberó en 1995 como un programa de 32 bits, dándole la versión 4.0. SourceSafe es un sistema basado en un equipo anfitrión a diferencia de la mayoría de los programas de control de versiones que son basados en Cliente-Servidor donde el repositorio de control de cambios reside en el equipo servidor y los clientes toman de allí la última versión para modificarla y posteriormente ingresarla con las modificaciones realizadas. Como en todos los programas de control de versiones, se basa en obtener una copia de trabajo ("check-out" o "desproteger"), realizar cambios sobre la copia y reingresarla al repositorio ("check-in" o "proteger"). Para lograr el acceso compartido al repositorio, VSS emplea el protocolo de archivos compartidos SMB lo cual crea algunos inconvenientes.

Ventajas de SourceSafe.

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.

Desventajas de SourceSafe.

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 éstos 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.

A pesar de ese inconveniente ha tenido buenos resultados en proyectos productivos de la Universidad de las Ciencias Informáticas donde la mayoría de los proyectos de desarrollo de software donde se trabaja con Visual Studio se aplica el Visual SourceSafe para su trabajo con el control de versiones de los códigos fuentes y ejecutables.

2.4 Conclusiones

En este capítulo se caracterizó el objeto de estudio y se realizó una revisión de las principales herramientas que se utilizan para el control de versiones y la gestión de cambios en proyectos de desarrollo de software y se obtuvo como resultado herramientas factibles para el trabajo en el Departamento de Desarrollo de Software (DDS) con las cuales se propone realizar todas las actividades de la gestión de configuración del software que conforman el proceso de GCS.

Capítulo 3: Propuesta de Solución y Validación de Resultados.

Propuesta de Solución.

El DDS necesita crear productos reutilizables que respondan a funcionalidades generales y que sean utilizados en las aplicaciones de gestión que se ejecuten. Todo software debe ser producido con rapidez y con calidad.

Ante la problemática existente en el DDS caracterizada en el capítulo 2 se plantea como procedimiento de trabajo para incluir en el proceso de desarrollo que se defina, así como las acciones identificadas que permitan ponerlo en vigor:

Realizar la administración de configuración de software del DDS de forma central en las primeras etapas, hasta alcanzar las experiencias que permitan tomar nuevas decisiones.

Acciones identificadas:

-Designar a nivel del DDS un administrador de configuración del software (ACS) que pertenezca al área de soporte tecnológico y se encargue de gestionar la configuración del software en todo el departamento. Este administrador debe:

-Crear un servidor Windows2003 para el repositorio de software, este repositorio es central y común a todos los proyectos y debe contener los elementos de configuración reutilizable y los productos software que componen un proyecto de gestión concluido.

-Instalar la herramienta Subversión para el trabajo con el repositorio.

-Establecer los permisos de acceso al repositorio previa coordinación con los equipos de proyectos que existan y autorización de la dirección del departamento.

-Mantener la integridad del repositorio, a partir de la incorporación de nuevos elementos reutilizables.

-Ejecutar la salva periódica del repositorio teniendo en cuenta las tecnologías de salva actuales.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Contar con un repositorio centralizado que contenga los productos de software reutilizables y su historial, estos serán utilizados en los diferentes proyectos y serán creados centralmente a partir de las funcionalidades generales que se determinen.

Acciones identificadas:

-Debe tener acceso de solo lectura para los equipos de proyectos.

-El ACS con la autorización de la dirección del DDS luego de la culminación de un proyecto que construya este tipo de producto o debido a un cambio formal sobre dichos elementos o cuando culmina un proyecto para almacenar todos sus elementos de configuración definidos en las líneas bases es quien escribe en el repositorio.

Componentes reutilizables

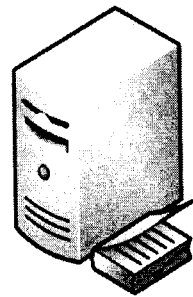
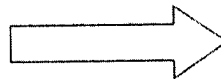
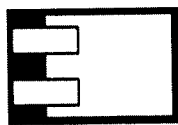


Ilustración 14 Procedimiento de componentes.

La gestión de configuración del software debe hacerse a partir del marco de trabajo propio que se defina para el proyecto de desarrollo del software.

Acciones identificadas:

-El jefe del equipo debe definir en cada proyecto las responsabilidades para realizar las diferentes actividades de la gestión de configuración del software.

-El Jefe del Equipo debe interactuar con el ACS a nivel de departamento para utilizar productos reutilizables del repositorio y en el caso de proyectos que desarrollan productos reutilizables al finalizar los mismos para incluirlos en el repositorio central.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Utilizar la herramienta Subversion para realizar la Gestión de Configuración con el propósito de mantener actualizado y bien documentados los archivos y componentes asociados al proceso de desarrollo del software.

Acciones identificadas:

-Designar en el área de soporte un especialista que asimile la herramienta Subversion para capacitar de forma ágil a los especialistas de los equipos de proyecto que realicen las tareas de gestión de configuración.

Establecer como elementos de configuración de los proyectos software son:

- Especificación de requisitos de software.
- Plan de gestión de configuración de software.
- Plan del proyecto.
- Plan de aseguramiento a la calidad del software.
- Plan de pruebas del sistema.
- Modelo funcional del sistema.
- Modelo de datos del sistema.
- Diseño detallado del sistema.
- Código fuente.
- Ejecutables.
- Manual de usuario.
- Casos de pruebas y resultados.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Establecer dentro del proceso de desarrollo las siguientes líneas bases que permiten gestionar el desarrollo software facilitando la consecución real de hitos intermedios para pasar desde las ideas al código:

-Línea base funcional: se establece al finalizar la fase de análisis y contendrá los requisitos del sistema, el plan de desarrollo, el plan de aseguramiento a la calidad, el plan de gestión de configuración y el plan de pruebas de aceptación.

-Línea base de asignación: se establece al finalizar la fase de diseño de alto nivel y contendrá la arquitectura del sistema, la especificación de interfaces de subsistemas y el plan de pruebas del sistema.

-Línea base de diseño: se establece al finalizar la fase de diseño detallado y contendrá el diseño detallado de los subsistemas y el plan de pruebas de integración.

-Línea base de producto: se establece al finalizar la codificación y las pruebas de integración y contendrá el código fuente, objeto y ejecutable, los resultados de las pruebas de integración así como una versión preliminar del manual de usuario.

-Línea base de explotación: se establece al finalizar la implantación e incluye los resultados de las pruebas del sistema, las de aceptación y la documentación de usuario.

En los proyectos basados en la tecnología .Net se puede utilizar la herramienta Visual SourceSafe para el trabajo con los ejecutables y sus versiones auxiliados con la herramienta Subversion para el control de la documentación asociada.

Establecer en cada proyecto la utilización de las bibliotecas de trabajo propias para cada miembro del equipo.

Acciones identificadas:

-Los ECS y documentos asociados a estos a nivel de puestos de trabajo se almacenan en esta biblioteca.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-Los cambios serán informales, los desarrolladores podrán cambiar cuanto deseen los ECS antes de ser subidos a la Biblioteca de Soporte del Proyecto.

Establecer en cada proyecto la utilización de una biblioteca de soporte del proyecto.

Acciones identificadas:

-El jefe del equipo almacena en esta biblioteca los ECS que se han subido desde la biblioteca de trabajo.

-Los cambios son semi informales, si se desea hacer algún cambio en los ECS que están guardados el jefe de proyecto da la autorización para poder bajar el ECS hacia la biblioteca de trabajo y realizar el cambio.

-En dependencia del tipo de ECS (componentes o documentos) puede ser manejada con la herramienta Visual SourceSafe o Subversion.

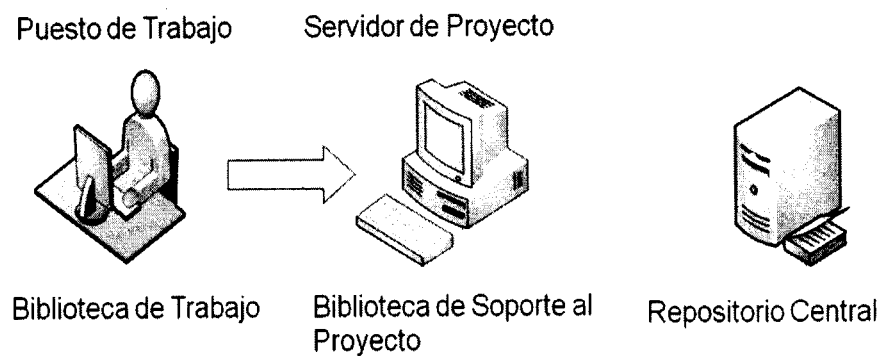


Ilustración 15 Procedimiento de bibliotecas de software.

La auditoría de configuración debe realizarla el Área de Calidad del DDS.

Acciones identificadas:

-El Jefe del Equipo debe definir con el Jefe del Área de Calidad del DDS quién y en qué momentos se realizará la actividad de auditoría de la configuración.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-En el Plan de la Configuración debe quedar reflejado esta responsabilidad.

El proceso de desarrollo de software tiene que incluir un proceso de gestión de configuración que garantice el control y coordinación sobre todos los artefactos o elementos de configuración del proyecto que se ejecute, que debe ser garantizado con una interfaz de configuración en la metodología que adopte el DDS.

La interfaz debe contener las siguientes actividades en dependencia del tipo de producto software que se construya y de la etapa del desarrollo en que se encuentre:

Definición de los requisitos de Gestión de Configuración: Permite identificar las necesidades de gestión de la configuración del sistema, definiendo los requisitos generales de gestión de la configuración y determinando los procesos de control que se van a llevar a cabo para mantener la integridad de los productos que se obtengan a lo largo de los procesos principales: control de versiones, control de estados, control de los cambios y otros que se determinen.

Productos de entrada:

-Identificación de requisitos

Productos de salida:

-Requisitos de gestión de la configuración

Participantes:

-Jefe de proyecto

-Equipo de proyecto

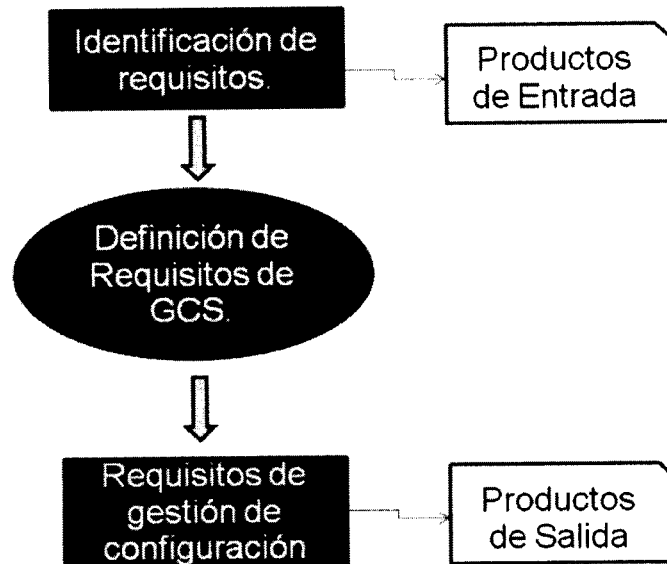


Ilustración 16 Esquema 1 de interfaz de configuración.

Establecimiento del Plan de Gestión de Configuración: Permite definir las políticas, estándares y procedimientos que se realizarán en el proyecto, debe ser flexible para cada proyecto. Se define el Plan de gestión de configuración para la solución propuesta y especificar el entorno tecnológico de soporte a la gestión de configuración.

Una vez establecido el Plan de gestión de la configuración, se irán registrando los productos que se obtengan en los procesos del desarrollo del software y que se hayan determinado en el plan como productos a incluir en el sistema de gestión de la configuración. En función de los requisitos de gestión de configuración obtenidos, es necesario establecer cuáles son estas necesidades y elaborar un plan de gestión de la configuración propio para el sistema.

Los aspectos que debe contemplar el plan son:

- Identificación de todos los productos que deben ser controlados, su clasificación y relaciones entre ellos, así como el criterio o norma de identificación.

- Ubicación y localización de los productos.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-Definición del ámbito y alcance del control de la configuración, describiendo los procesos incluidos en él.

-Definición de las reglas de versionado de los productos y los criterios de actuación para cada caso, teniendo en cuenta el motivo por el cual se realiza el cambio de versión.

-Definición del ciclo de estados para cada tipo de producto y los criterios de trazabilidad entre los mismos.

-Descripción de funciones y responsabilidades.

-Identificación de la información necesaria de control para auditoría.

Una vez obtenido el plan de gestión debe definirse el entorno tecnológico de soporte a la gestión de configuración determinando los componentes hardware y software que van a permitir la mecanización de los procesos y controles que establece el plan.

Productos de entrada

-Requisitos generales de gestión de configuración

-Solución propuesta

Productos de salida:

-Plan de gestión de la configuración

-Entorno tecnológico

Participantes:

-Jefe de proyecto

-Equipo de proyecto

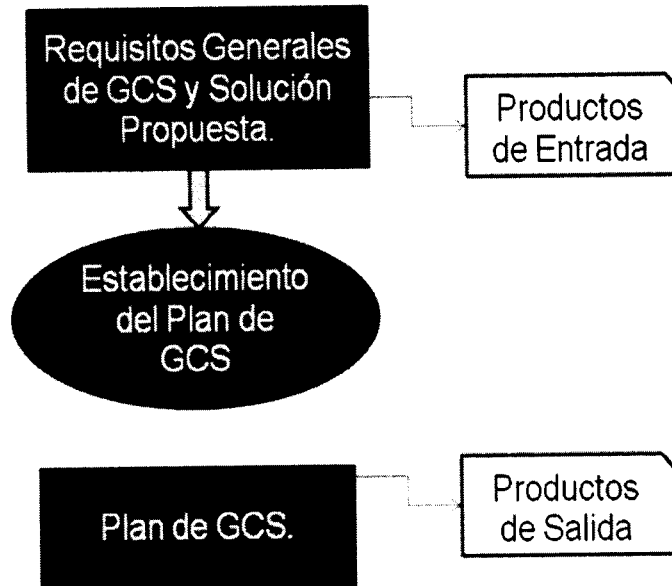


Ilustración 17 Esquema 2 de interfaz de configuración.

A partir del análisis encontrado en la tesis "Gestión de la Configuración del Software de Intranet II de la UCI" se parte de la propuesta que hace IEEE de Plan de Gestión (anexo 2) para dar un formato de plan ajustado a las características del departamento:

Sección 1: Introducción

1.1 Propósito: De forma breve describir el sistema al que se aplica el Plan de GCS.

1.2 Alcance: Aquí se define el alcance que va a tener la gestión de configuración en el proyecto, es decir:

-A que proyectos o programas se aplica

-Cuales serán los elementos de configuración que se van a gestionar, tanto código como documentación.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

1.2 Definiciones y acrónimos:

-Términos, definiciones y acrónimos propios de este documento, tales como líneas base, elementos de configuración.

Ejemplo:

D	Defectos Descubiertos.
D	
D	Defectos Resueltos.
R	
E	Elementos de Configuración del Software.
CS	
G	Gestión de Configuración del Software.
CS	
IE	Institute of Electrical and Electronics
EE	Engineers.
L	Línea Base.
B	
M	Momento del ciclo de vida en que son
D	Detectados los defectos.
M	Momento del ciclo de vida en que son
R	Resueltos los defectos.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

Sección 2: Gestión, Calendario y Recursos de la GCS.

2.1. Responsabilidades:

-Asignará la responsabilidad para las diferentes actividades de GCS a los miembros del equipo y para la actividad de auditoría a un miembro del área de calidad.

2.2. Calendario y Recursos.

- La programación para las revisiones y auditorías de Gestión de Configuración.
- La gestión de la configuración de herramientas de desarrollo de software, pruebas y soporte.

Sección 3: Actividades de la Gestión de Configuración.

3.1- Identificación de la Configuración:

- Establecimiento y enumeración de las líneas base que se van a establecer, y para cada una de ellas descripción de:

- El momento en el que se establecerá
- Los elementos de configuración que incluirá.

- Las relaciones entre los ECS

- Bibliotecas y repositorios software que se utilizarán. Descripción de los procedimientos que se utilizarán para situar físicamente los elementos de configuración en las bibliotecas software, para recuperarlos, protegerlos, etc.

3.2- Control de la Configuración:

-Definición de los mecanismos de evaluación e implementación de cambios: Procedimientos y formularios a utilizar.

- Mecanismos de Gestión de Problemas que se utilizarán.
- Mecanismos para Control de Versiones.

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

3.3- Contabilidad del Estado de la Configuración.

- Definición de los registros que se van a mantener y los informes que se van a generar.

3.4. Auditoria de la Configuración.

- Descripción de cada una de las auditorias y revisiones que se van a realizar, indicando:

- Objetivos.
- Elementos de configuración a auditar o revisar.
- Listas de comprobación y cuestionarios a utilizar.
- Registro del resultado de la auditoría.

Identificación y registro de los productos:

Cada producto, en función de su naturaleza, va pasando por diferentes estados. El conjunto de estados por los que va pasando un producto debe quedar registrado, según vaya evolucionando el mismo, hasta alcanzar el estado de producto final aprobado. El conjunto de estados posibles para cada producto son: en elaboración, finalizado, revisado y aceptado; y otros que se consideren necesarios.

En el momento en que un participante genere un producto, seleccionado como elemento de configuración en el plan de gestión de la configuración, deberá identificarlo y registrarlo. Del mismo modo, cuando modifique un producto que ya está registrado en el repositorio central, el administrador de la configuración deberá actualizar la información del registro correspondiente al producto modificado, indicando su versión y estado.

En la biblioteca de soporte al proyecto deben quedar identificados y registrados los productos globales que se obtienen a lo largo del desarrollo de los procesos principales y una vez puesto en producción deben quedar registrados en el repositorio central por el administrador de la configuración.

Al finalizar los procesos de:

- Análisis y diseño del Sistema, (AS)
- Diseño Detallado del Sistema (DS)
- Construcción del Sistema (CS)

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-Implantación y Aceptación (IA)

En cada momento se registrará el producto correspondiente que se corresponden con las líneas bases definidas:

- Especificación de requisitos de software, diseño de alto nivel y la arquitectura
- Diseño detallado del sistema
- Sistema
- Sistema implantado.

Una vez que el sistema está implantado y aceptado se registra el paso a producción de la versión de dicho sistema en el repositorio central por parte del ACS de cara a facilitar su posterior mantenimiento.

Productos de entrada:

- Información sobre el producto
- Registro de los productos creados o modificados
- Identificación
- Versión
- Estado
- Localización

Productos de salida:

- Registro de los productos creados o modificados
- Identificación
- Versión
- Estado
- Localización

Participantes:

- Administrador de la configuración
- Jefe de proyecto
- Equipo de proyecto

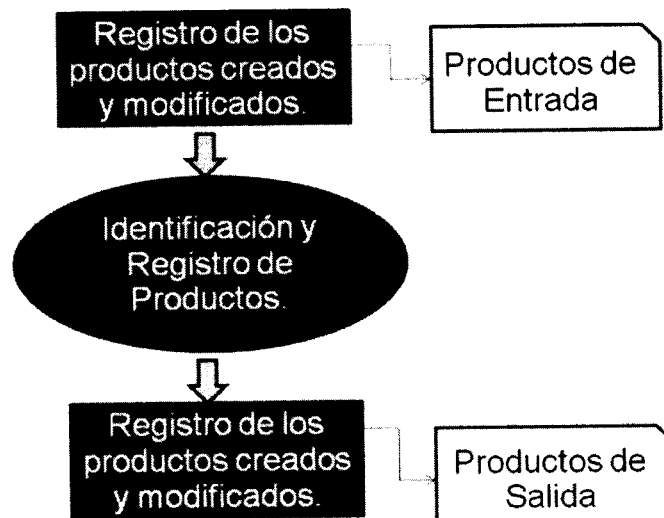


Ilustración 18 Esquema 3 de interfaz de configuración.

Mantenimiento del Sistema: Permite conservar la integridad del sistema cuando se producen cambios en el mismo, ya sea por la realización de mantenimiento correctivo o evolutivo. En este proceso la gestión de la configuración produce los beneficios siguientes:

- Reducción del tiempo de localización de los problemas.
- Minimización de la reproducción de errores
- Control y seguimiento de los estados por los que va pasando la petición de mantenimiento.

De esta manera se puede conocer en cada momento la situación en la que se encuentra cada cambio en particular y el sistema en general. Para realizar la petición de un cambio se solicita a la interfaz de

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

gestión de configuración información para identificar las versiones de los sistemas afectados por la petición.

Una vez que el cambio ha sido aceptado y realizado se registra por el ACS en el repositorio central de productos, la nueva versión de los sistemas y de los productos según el criterio de versionado establecido en el plan de gestión de la configuración del sistema.

La información mantenida en este registro permite en todo momento efectuar una traza de la evolución del sistema y los productos que lo integran desde su puesta en producción como consecuencia de la realización de cambios.

Productos de entrada

- Catálogo de peticiones
- Registro de la versión correspondiente de los productos que van a ser modificados
- Registro de la versión del sistema de la que parte el cambio

Productos de salida:

- Registro del cambio
- Registro de la versión correspondiente de los productos que van a ser modificados
- Registro de la versión del sistema de la que parte el cambio

Participantes:

- Administrador de configuración
- Equipo de proyecto para el mantenimiento

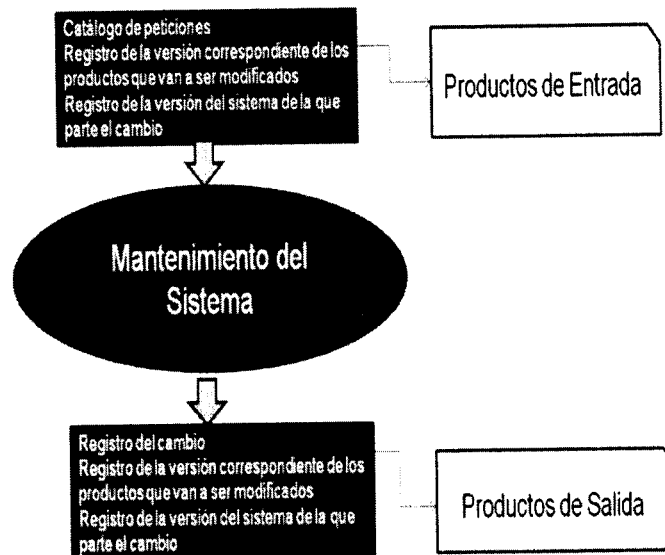


Ilustración 19 Esquema 4 de interfaz de configuración.

3.3 Validación de la propuesta de procedimiento para la gestión de configuración en el DDS.

Este procedimiento fue validado mediante la técnica de consultas a expertos, realizándose una entrevista a seis especialistas del DDS y el MININT, con el objetivo de evaluar la propuesta en cuestión. Tuvo la particularidad de que se les explicó primero su esencia así como se encuentra el tema en el DDS. Se alcanzó una buena comunicación y cooperación con los entrevistados.

Del total de los consultados el 50 % están categorizados como investigadores auxiliares y el 100% poseen más de 15 años de servicio directo en el desarrollo de software.

Los indicadores que se midieron fueron:

- Utilidad
- Novedad
- Necesidad

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE PARA EL DEPARTAMENTO DE DESARROLLO DE SOFTWARE.

-Factibilidad

-Valor práctico

En este cuestionario el 100% de los especialistas plantearon estar de acuerdo con la utilidad de la propuesta, pues se le da una introducción a los temas que abarcan la gestión de configuración y se brindan elementos esenciales a tener en cuenta por los equipos de proyectos para obtener software visible, íntegro y bien controlado.

De igual forma la totalidad acepta que es necesario, factible y posee valor práctico la aplicación del procedimiento que se propone.

Consideran además que debe ser sistemáticamente ajustado en dependencia de la experiencia que se obtenga y en la medida que se formalice el proceso de desarrollo de software, creando las condiciones para que sea del conocimiento de otros centros que desarrollen software a través de conferencias, seminarios, u otras jornadas de preparación con el objetivo de compartir experiencias y ampliar el uso de las herramientas propuestas.

3.4 Conclusiones

Al concluir este capítulo se le brinda al Departamento de Desarrollo de Software (DDS) un procedimiento propio de gestión de configuración de software, ajustado a las características actuales del mismo y que resulta de gran importancia para un buen desempeño futuro en cualquier proyecto que se lleve a cabo dentro del departamento, permitiendo la visibilidad del proyecto de software a partir del control de sus elementos de configuración y del registro y la evaluación de los cambios sobre ellos y el control uniforme de sus diferentes versiones. El control central de los elementos reutilizables permitirá crear productos con mayor rapidez y calidad. Como fundamento para la creación del plan de configuración de software para el DDS se tuvo en cuenta el estándar brindado por la IEEE.

Conclusiones Generales.

El autor considera haber alcanzado el objetivo general trazado en la investigación al proponer al DDS un procedimiento que de soporte a la Gestión de Configuración en el proceso de desarrollo de software, encaminado a fortalecer las mismas.

Al caracterizar el estado actual del DDS se pudo constatar que no existe un proceso de desarrollo de software formalizado que permita obtener software ágil y con calidad, así como la no aplicación del posible re uso que es clave para lo anterior y el desconocimiento del tema de la gestión de configuración.

El estudio comparativo de las principales herramientas que se utilizan para el control de versiones y la gestión de cambios en proyectos de desarrollo de software obtuvo como resultado herramientas factibles para el trabajo en el Departamento de Desarrollo de Software (DDS) con las cuales se propone realizar todas las actividades de la gestión de configuración del software que conforman el proceso de GCS.

Aplicando los métodos y técnicas diseñados, se obtuvo un procedimiento propio de gestión de configuración de software, ajustado a las características actuales del DDS y que resulta de gran importancia para un buen desempeño futuro en cualquier proyecto que se lleve a cabo dentro del departamento, permitiendo la visibilidad del proyecto de software a partir del control de sus elementos de configuración y del registro y la evaluación de los cambios sobre ellos y el control uniforme de sus diferentes versiones, así como el control central de los elementos reutilizables que permitirá crear productos con mayor rapidez y calidad.

Los expertos coincidieron que la propuesta tiene utilidad, valor práctico y es factible, además de su novedad en el DDS, considerando necesaria, oportuna y objetiva su implementación.

Recomendaciones

-Instrumentar en el DDS la propuesta que se aborda en esta investigación, comenzando de forma inmediata con las acciones de capacitación de los especialistas del área de soporte, en especial del Administrador de la Configuración que se designe.

-De igual forma dar conocimiento de esta propuesta a todo el personal que se requiera

-Continuar la identificación de las acciones que permitan perfeccionar el procedimiento propuesto.

-Continuar profundizando en esta investigación para comprobar la efectividad de la propuesta, la cual requiere de un continuo estudio:

En el orden tecnológico, en la asimilación de nuevas herramientas que resuelvan las desventajas de la que se seleccionaron y que están planteadas en la tesis.

En el orden de la capacitación, a partir del desarrollo vertiginoso de la tecnología, se exige una continua y permanente superación de los especialistas dedicados al desarrollo de software.

Bibliografía

Anónimo. *Ayuda de Tortoiseshvn.*

—. 2007. [html.rincondelvago.com. html.rincondelvago.com/configuracion-de-software.html](http://html.rincondelvago.com/html.rincondelvago.com/configuracion-de-software.html). [En línea] 2007. [Citado el: 14 de enero de 2008.] <http://html.rincondelvago.com/configuracion-de-software.html>.

—. 2006. *Propuesta Arquitectónica, Instituto superior politécnico "José Antonio Hechevarría"*. Ciudad Habana : s.n., 2006.

—. [www.csi.map.es. www.csi.map.es/csi/metrica3/gescon.pdf](http://www.csi.map.es/www.csi.map.es/csi/metrica3/gescon.pdf). [En línea] [Citado el: 16 de enero de 2008.] <http://www.csi.map.es/csi/metrica3/gescon.pdf>.

—. [www.histaintl.com. www.histaintl.com/soluciones/configuracion/configuracion.php](http://www.histaintl.com/www.histaintl.com/soluciones/configuracion/configuracion.php). [En línea] [Citado el: 13 de enero de 2008.] <http://www.histaintl.com/soluciones/configuracion/configuracion.php>.

—. 2007. [www.pbmiki.com. www.pbmiki.com/Gesti%C3%B3n%20de%20la%20Configuraci%C3%B3n](http://www.pbmiki.com/www.pbmiki.com/Gesti%C3%B3n%20de%20la%20Configuraci%C3%B3n). [En línea] 2007. [Citado el: 13 de enero de 2008.] <http://www.pbmiki.com/Gesti%C3%B3n%20de%20la%20Configuraci%C3%B3n>.

Antonio, Angélica de. 2001. *La Gestión de Configuración del Software*. 2001.

Armando Cabrera, Fernanda M.Soto. *Gestión de Configuración*.

Babich. [www.um.es. www.um.es/atika](http://www.um.es/www.um.es/atika). [En línea] [Citado el: 13 de enero de 2008.] <http://www.um.es/atika/mnes-seccion-de-metodologia>.

Bersoft, Henderson, Siegel. 1980. *Software Configuration Management*. Prentice-Hall : s.n., 1980.

Febles, Aylin. 2004. *Mconfig.PM , Modelo de referencia para la gestión de configuración en la pequeña y mediana empresa de software*. Ciudad Habana : s.n., 2004.

Febles, Carbó. 2007. *Estrategia para la gestión de configuración de software del proyecto registro y notarias*. Ciudad Habana, UCI : s.n., 2007.

Franco, José Antonio. 2006. *Entorno Unificado para la gestión de configuración*. Ciudad Habana : s.n., 2006.

IBM. 2003. *Rational Unified Process*. 2003.

IEEE. 1990. www.ieee.org. www.ieee.org/ni/aboutieee.htm. [En línea] 1990. [Citado el: 13 de enero de 2008.] <http://www.ieee.org/ni/aboutieee.htm>.

—. 1998. www.ieee.org. www.ieee.org/ni/aboutieee.htm. [En línea] 1998. <http://www.ieee.org/ni/aboutieee.htm>.

—. 2007. www.ieee.org/ni. www.ieee.org/ni/aboutieee.htm. [En línea] 2007. [Citado el: 15 de enero de 2008.] <http://www.ieee.org/ni/aboutieee.htm>.

L.Díaz. 2007. *Gestión de la configuración del software de intranet II de la UCI*. Ciudad Habana, UCI : s.n., 2007.

Pressman. 2001. *Ingeniería del Software, un enfoque práctico*. Madrid : s.n., 2001.

Ramírez, Alejandro. *Tutorial del sistema de control de versiones subversion*.

Ramsés Delgado, Emilio glez. *ConfigCase 3.0 , Herramienta de apoyo a la gestión de configuración*.

Zeledón, Collins. www.fernandoflores.cl. www.fernandoflores.cl/node/137. [En línea] [Citado el: 15 de enero de 2008.] <http://www.fernandoflores.cl/node/137>.

Anexos

Anexo de ejemplo de un plan dado por IEEE

1. Introducción

1.1 Propósito del Plan: En esta parte se explica cual va a ser el propósito y a quien va dirigido principalmente, esencialmente esto se hace para establecer y documentar las políticas, estándares y procedimientos para el plan de configuración de los elementos de configuración a desarrollar.

1.2 Alcance: Aquí se define el alcance que va a tener la gestión de configuración en el proyecto , es decir:

- A que proyectos o programas se aplica

- Cuales serán los elementos de configuración que se van a gestionar, tanto código como documentación.

- Limitaciones (recurso, tiempo).

1.3 Definiciones y acrónimos:

- Términos, definiciones y acrónimos propios de este documento, tales como líneas base, elementos de configuración, comité de control de cambios, etc.

1.4-Referencias:

- Todo tipo de referencias que se hagan para realizar con calidad el proyecto.

1.5-Definición de alto nivel del proceso de GCS.

- Principales interacciones de las personas y las actividades de GCS, indicando los flujos de información y los repositorios que se utilizarán.

2. Especificaciones de Gestión.

2.1. Organización:

-Descripción del contexto organizativo en el que se desarrollarán las actividades de GCS, definiendo las funciones e interfaces entre ellas. Definirá la relación de la Gestión de Configuración con la Gestión del Proyecto, y las relaciones de dependencia y autoridad.

2.2. Responsabilidades:

-Asignará la responsabilidad para las diferentes actividades de GCS a ciertas personas u organizaciones. Así, por ejemplo, definirá:

2.3. Implementación del Plan de Gestión de Configuración.

- El establecimiento del Comité de Control de Cambios
- El establecimiento de cada una de las líneas base
- La programación para las revisiones y auditorías de Gestión de Configuración.
- La gestión de la configuración de herramientas de desarrollo de software, pruebas y soporte.

3. Actividades de la Gestión de Configuración.

3.1- Identificación de la Configuración:

- Descripción del esquema de identificación que se utilizará para los elementos de configuración.

- Enumeración de las líneas base que se van a establecer, y para cada una de ellas descripción de:

- El momento en el que se establecerá
- Los elementos de configuración que incluirá
- Bibliotecas y repositorios software que se utilizarán. Descripción de los procedimientos que se utilizarán para situar físicamente los elementos de configuración en las bibliotecas software, para recuperarlos, protegerlos, etc.

3.2- Control de la Configuración:

-Definición de los mecanismos de iniciación de cambios: Procedimientos y formularios a utilizar.

- Definición de los mecanismos para evaluación de los cambios: Procedimientos y criterios a utilizar.

- Definición de los mecanismos para aprobación o rechazo de un cambio: Quién tiene la autoridad para ello y los procedimientos que se utilizarán.

- Definición de los mecanismos para evaluación de cambios: Procedimientos para verificar e implementar un cambio aprobado.

- Mecanismos de Gestión de Problemas que se utilizarán.

- Mecanismos para Control de Versiones.

-Identificación y registros de productos así como el registro del cambio en el sistema.

3.3. Contabilidad del Estado de la Configuración.

- Definición de los registros que se van a mantener y los informes que se van a generar.
- Definición de los procedimientos de captura, almacenamiento y procesamiento de la información.

3.4. Auditoría de la Configuración.

- Descripción de cada una de las auditorías y revisiones que se van a realizar, indicando:
 - Objetivos.
 - Elementos de configuración a auditar o revisar.
 - Programación de las tareas de la auditoría o revisión.
 - Procedimientos para llevar a cabo la auditoría o revisión.
 - Participantes.
 - Documentación que debe estar disponible para realizar la auditoría o revisión.
 - Listas de comprobación y cuestionarios a utilizar.
 - Procedimiento para registrar deficiencias.
 - Criterios de aprobación del elemento de configuración.