

**Universidad de las Ciencias Informáticas**

**Facultad 2**



**TÍTULO**

**Procedimiento de control y aseguramiento de la calidad en la  
Administración de la Configuración y Cambio.**

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

**Autores:** Yulien Gallego Parra.  
Diana Ibett Arias Herrera.

**Tutor:** Ing. Wilfredo González Vidal.

Ciudad de La Habana, Julio 2007

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Yulien Gallego Parra

\_\_\_\_\_  
Diana ibett Arias Herrera

\_\_\_\_\_  
Wilfredo González Vidal

## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Procedimiento de control y aseguramiento de la calidad en la Administración de la Configuración y Cambio**, fue realizado en la **Universidad de las Ciencias Informáticas**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente  
 Parcialmente en un \_\_\_\_ %

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

---

---

---

---

---

---

---

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a **<valor en MN o USD del efecto económico>**

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Representante de la entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño

## **OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA**

**Título:** Procedimiento de control y aseguramiento de la calidad en la Administración de la Configuración y Cambio.

**Autores:** Yulien Gallego Parra.

Diana Ibett Arias Herrera.

El tutor del presente Trabajo de Diploma considera que durante su ejecución las estudiantes mostraron las cualidades que a continuación se detallan.

En el trabajo de referencia considero de MUY ALTA la actitud y aptitudes mostradas por las estudiantes autoras del mismo. Es importante señalar que en todo este periodo han demostrado disponer de un alto nivel de sacrificio y entrega, cualidades que sin dudas denotan una personalidad responsable y laboriosa. La independencia y creatividad ha sido la principal distinción por parte de las estudiantes lo cual ha posibilitado la creación de un material de consulta importante para la Universidad de Ciencias Informáticas y en general para todas las entidades de este perfil del país. La tesis que elaboraron dispone de un alto valor científico - técnico, la cual se distingue por la creatividad y rigor profesional, al concentrar información relevante asociada al área de proceso clave en cuestión. Considero que el aporte y el cumplimiento de los objetivos trazados fueron llevados a cabo con alta expectativa, lográndose obtener de forma sencilla y oportuna beneficios reales para el control y seguimientos de todos los elementos de configuración en un ciclo o proyecto completo de desarrollo de software.

En resumen me siento satisfecho por el rigor y empeño puesto en este trabajo, el cual no es más que el propio fruto y esfuerzo de cada uno de los profesores que forman y educan en esta importante ciudad universitaria. Por todo lo anteriormente expresado considero que las estudiantes están aptas para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos con la posibilidad de realizar una publicación de los resultados obtenidos.

---

Firma

---

Fecha

*Agradecimientos compartidos.*

*No hay nada imposible, porque los sueños de ayer son las esperanzas de hoy y pueden convertirse en realidad mañana.*

*Ante todo dirigimos nuestros agradecimientos a Dios, por guiarnos por el camino correcto, dándonos la fe suficiente para sobreponernos a las dificultades. Fue del que siempre recibimos un sí como respuesta a nuestras oraciones.*

*A Wilfredo, quien durante todo el desarrollo de este trabajo más que un tutor fue un gran amigo. A él le agradecemos todo la amabilidad y las horas que nos dedicó, aún cuando las responsabilidades en su trabajo le exigían. Fue la persona a la que siempre pudimos recurrir y encontrar respuestas a nuestras incógnitas, permaneció a nuestro lado, atento a nuestras necesidades. Queremos agradecer en especial su incondicionalidad puesto que sin su mano amiga hubiera sido imposible presentar estos resultados. Realmente no encontramos palabras que describan los agradecimientos que lo caracterizan. Gracias por haber aceptado formar parte de este momento único.*

*A todo el personal de la empresa Casa de Software ETECSA-SIGTA, por todo el cariño y la atención brindada, en especial a Héctor, Ernesto y Seida, con los que pudimos contar, desde una respuesta de correo, hasta una consulta referente al tema de tesis.*

*A los compañeros Rosita, Yenlys y Lázaro de la empresa DESOFT, por la colaboración prestada proporcionándonos información vital para el desarrollo de este trabajo. A las colegas Rosalía Quest y Ana Silvia de la empresa SOFTEL por la ayuda prestada.*

*A los miembros del proyecto SCADA dentro de la universidad, por proporcionarnos la documentación referente a este tema de tesis. A la Revolución y a la Universidad de las Ciencias Informáticas por formarnos durante estos 5 años como profesionales, a todos los profesores, que con su enseñanza y su paciencia colaboraron para que llegáramos al triunfo de nuestros estudios.*

### *Yulien*

*A mi mamá por ser una madre tan especial, dedicada y entregada, mejor no la quiero, por ti y para ti he llegado a la cima.*

*A mis abuelas por estar siempre pendiente de mí, por quererme tanto.*

*A mis padres por darme la fuerza necesaria para seguir adelante, por todo el esfuerzo realizado con el fin de garantizar mi futuro, gracias por todo.*

*A mis tíos por todo el cariño demostrado, por apoyarme cuando lo he necesitado, por quererme siempre como a una hija.*

*A mi familia en general por haber creado todo lo que soy, por formarme como una mujer de bien, por ayudarme hacer realidad este sueño de todos.*

*A mi novio y a mis suegros, por toda la atención brindada y el amor entregado.*

*A todas mis tías y primas de La Habana por haber sido tan atentos conmigo, por acogerme como si fuera parte de la familia, y en especial a Aida por quiarme, por orientarme, por darme tantos consejos y tanto cariño siempre.*

*A mis amigas Irina, Dayamí, Diana, Liuby y a mi amigo Egar por confiar en mí, por brindarme su amistad y su ayuda en cada momento. Nunca los voy a olvidar.*

### *Diana*

*Mis padres por todo el apoyo, dedicación, porque siempre han estado a mi lado aun lejos, a ellos le debo todo lo que hoy soy.*

*A mi hermana Deli que más que hermana es mi amiga, mi cómplice, quien siempre ha estado dispuesta a escucharme.*

*A Isly por toda la alegría que me hizo llegar, por sus consejos, apoyo incondicional por ser especial.*

*A mis amigas Giselle y Gerxis, con quienes siempre pude contar tanto en las buenas como en las malas, por ser súper especiales. A Daya y Yuli por todo el cariño y apoyo, porque nunca hubo un no.*

*Dedicatoria.*

*A mi madre por ser la luz de mis días*

*A mis abuelas por ser la fuerza inspiradora*

*A mis padres, y a mis tíos por todo el apoyo y el amor mostrado.*

*Yulien*

*A mis padres que lo han dado todo por ver realizado  
este gran sueño.*

*Diana*

## RESUMEN

En la actualidad la Industria Cubana del Software se ha convertido en un sector de vital importancia para la economía de nuestro país. A través de los años se han realizado numerosos estudios sobre las causas del fracaso de los proyectos, muchos de estos, tienen su raíz en malas prácticas de ingeniería y gestión de software durante el proceso de desarrollo, pero gracias a la disciplina conocida como Gestión de Configuración muchos de estos problemas pueden solucionarse.

En el presente trabajo se realiza la propuesta de un procedimiento que sirve de guía a los proyectos de desarrollo de software en la Universidad de las Ciencias Informáticas (UCI), haciendo referencia a los conceptos fundamentales de esta disciplina. Como resultado de este estudio se analizan las herramientas que han existido y otras que han surgido en estos últimos años, para solucionar muchas de las situaciones que provocan el fracaso de los proyectos, con el objetivo de incrementar el grado de éxito en los proyectos de desarrollo de software.

De igual forma se hace referencia a las empresas que se dedican a la producción de software en nuestro país, como La Casa de Software SIGTA-ETECSA, DESOFT, SOFTEL y a nivel internacional AMCIS, para lograr unificar los procedimientos empleados por estas instituciones para un mayor control y calidad en esta área de proceso clave.

## ÍNDICE

|  |    |
|--|----|
| INTRODUCCIÓN .....   | 1  |
| CAPÍTULO 1 .....   | 6  |
| FUNDAMENTACIÓN TEÓRICA DE GESTIÓN DE CONFIGURACIÓN. ....   | 6  |
| 1.1 INTRODUCCIÓN. ....   | 6  |
| 1.2 GESTIÓN DE CONFIGURACIÓN. DEFINICIONES. ....   | 6  |
| 1.3 SITUACIÓN NACIONAL E INTERNACIONAL DE LA GESTIÓN DE CONFIGURACIÓN. ....                        | 7  |
| 1.4 GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE (GCS) EN LOS MODELOS DE CALIDAD. ....                    | 11 |
| 1.4.1 Modelo de Calidad. Concepto. ....  | 11 |
| 1.4.2 IEEE. ....   | 12 |
| 1.4.3 Integración del Modelo de Madurez de las Capacidades (CMMI). ....                            | 12 |
| 1.4.4 La Organización Internacional de Estándares (International Standard Organización, ISO). .... | 14 |
| 1.5 PROCESO UNIFICADO DE RATIONAL (RATIONAL UNIFIED PROCESS, RUP). ....                            | 15 |
| 1.6 GESTIÓN DE CONFIGURACIÓN. IMPORTANCIA. ....  | 16 |
| 1.7 ADMINISTRACIÓN DE LA CONFIGURACIÓN. ....   | 17 |
| 1.7.1 Requerimientos para el éxito. ....   | 17 |
| 1.7.2 Roles y responsabilidades. ....  | 18 |
| 1.7.3 Control de versiones. ....   | 19 |
| 1.7.4 El proceso de Administración de la Configuración. ....                                       | 21 |
| 1.5 LÍNEA BASE. ....   | 25 |
| 1.5.1 Alcance de la Línea Base para el control de código. ....                                     | 26 |
| 1.5.2 Establecimiento de líneas base. ....   | 26 |
| 1.6 ORGANIZACIÓN DEL AMBIENTE CONTROLADO EN LA GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE. ....         | 28 |
| 1.6.1 Librería de Software. Definición. ....   | 29 |
| 1.6.2 Tipos de librerías de software. ....   | 29 |
| 1.7 CONTROL DE CAMBIOS EN LA CONFIGURACIÓN. ....   | 31 |
| 1.7.1 Tipos de cambios. ....   | 31 |
| 1.7.2 Niveles de control de cambios. ....  | 32 |
| 1.7.3 Roles y responsabilidades. ....  | 32 |
| 1.7.4 El proceso de control de cambio. ....  | 33 |
| 1.8 CONCLUSIONES PARCIALES DEL CAPÍTULO 1. ....  | 34 |
| CAPÍTULO 2 .....   | 35 |
| EVALUACIÓN DE HERRAMIENTAS .....   | 35 |
| 2.1 INTRODUCCIÓN. ....   | 35 |
| 2.2 HERRAMIENTA RATIONAL CLEAR CASE. ....  | 35 |
| 2.2.1 Ventajas. ....   | 35 |
| 2.2.2 Integración. ....  | 36 |
| 2.2.3 Características principales. ....  | 36 |

|   |           |
|---|-----------|
| 2.2.4 Productos de Rational Clear Case. ....  | 36        |
| 2.3 HERRAMIENTA RATIONAL CLEAR QUEST. ....  | 37        |
| 2.3.1 Ventajas. ....  | 37        |
| 2.3.2 Características. ....   | 38        |
| 2.3.3 Producto de Rational Clear Quest. ....  | 38        |
| 2.4 HERRAMIENTA VISUAL SOURCE SAFE 2005.....  | 38        |
| 2.4.1 Ventajas. ....  | 38        |
| 2.4.2 Características. ....   | 39        |
| 2.5 HERRAMIENTA PLASTIC SCM. ....   | 39        |
| 2.5.1 Ventajas. ....  | 39        |
| 2.5.2 Características. ....   | 40        |
| 2.5.4 Integración.....  | 40        |
| 2.6 HERRAMIENTA VISUAL STUDIO TEAM SYSTEM. ....   | 40        |
| 2.6.1 Ventajas. ....  | 41        |
| 2.6.2 Características. ....   | 41        |
| 2.6.3 Productos de Microsoft Visual Studio 2005 Team Suite.....                                       | 41        |
| 2.6.4 Microsoft Visual Studio 2005 Team Foundation Server.....  | 41        |
| 2.7 HERRAMIENTA BORLAND STARTEAM VERSIÓN 6.0.....   | 43        |
| 2.7.1 StarTeam Server.....  | 44        |
| 2.7.2 StarTeam Client. ....   | 44        |
| 2.7.3 Ventajas. ....  | 44        |
| 2.7.4 Características principales.....  | 45        |
| 2.7.5 Productos de StarTeam. ....   | 45        |
| 2.8 HERRAMIENTA PVCS DIMENSIONS 7.0.....  | 46        |
| 2.8.1 Ventajas. ....  | 46        |
| 2.8.2 Características. ....   | 47        |
| 2.8.3 Integración.....  | 47        |
| 2.9. HERRAMIENTA SUBVERSION (SVN). ....   | 47        |
| 2.9.1 Ventajas. ....  | 48        |
| 2.9.2 Características principales.....  | 48        |
| 2.10. PRODUCTO GFORGE. ....   | 49        |
| 2.10.1 Ventajas Gforge. ....  | 49        |
| 2.10.2 Características de Gforge.....   | 49        |
| 2.11 HERRAMIENTA SUITE ALLFUSION™ CHANGE MANAGEMENT. ....   | 50        |
| 2.11.1 Herramienta AllFusion Harvest Change Manager. ....   | 50        |
| 2.11.2 AllFusion Endeavor Change Manager.....   | 51        |
| 2.11.3 AllFusion Change Manager Enterprise Workbench. ....  | 52        |
| 2.12 OTRAS HERRAMIENTAS EXISTENTES PARA LLEVAR A CABO EL CONTROL DE LA GESTIÓN DE CONFIGURACIÓN. .... | 54        |
| 2.13 CONCLUSIONES PARCIALES DEL CAPÍTULO 2. ....  | 54        |
| <b>CAPÍTULO 3</b> .....   | <b>55</b> |
| <b>CASOS DE ESTUDIO DE PROCEDIMIENTOS EXISTENTES</b> .....  | <b>55</b> |
| 3.1 INTRODUCCIÓN. ....  | 55        |
| 3.2 PLAN DE GESTIÓN DE CONFIGURACIÓN. ....  | 55        |

|            |   |    |
|------------|---|----|
| 3.2.1      | Introducción.....   | 55 |
| 3.2.2      | Gestión.....  | 56 |
| 3.2.3      | Actividades.....  | 58 |
| 3.2.4      | Planificación.....  | 60 |
| 3.2.5      | Herramientas, Técnicas y Metodologías.....  | 60 |
| 3.2.6      | Mantenimiento del Plan.....   | 61 |
| 3.2.7      | Captura y Retención de Registros.....   | 61 |
| 3.3        | MODELO DE PROCESOS PARA LA INDUSTRIA DE SOFTWARE (MOPROSOFT VERSIÓN 1.3).....                                 | 61 |
| 3.4        | PROCEDIMIENTO UTILIZADO POR LA CASA DE SOFTWARE ETECSA-SIGTA.....   | 62 |
| 3.4.1      | Objetivo.....   | 62 |
| 3.4.2      | Alcance.....  | 62 |
| 3.4.3      | Definiciones.....   | 62 |
| 3.4.4      | Referencias.....  | 63 |
| 3.4.5      | Responsabilidades.....  | 63 |
| 3.4.6      | Instrucciones.....  | 63 |
| 3.5        | PROCEDIMIENTO EN PROCESO DE IMPLANTACIÓN DE LA EMPRESA DESOFT.....  | 64 |
| 3.5.1      | Objetivo.....   | 64 |
| 3.5.2      | Alcance.....  | 64 |
| 3.5.3      | Definiciones y Términos.....  | 64 |
| 3.5.4      | Bibliotecas de trabajo utilizadas.....  | 65 |
| 3.5.5      | Gestión de Configuración.....   | 65 |
| 3.5.6      | Descripción de las Actividades.....   | 65 |
| 3.5.7      | Responsabilidades.....  | 66 |
| 3.5.8      | Registros.....  | 66 |
| 3.5.9      | Referencias bibliográficas.....   | 66 |
| 3.6        | PLAN DE GESTIÓN DE CONFIGURACIÓN UTILIZADO POR LA EMPRESA SOFTEL REFERENTE A LA GESTIÓN DE CONFIGURACIÓN..... | 66 |
| 3.6.1      | Introducción.....   | 66 |
| 3.6.2      | Organización, Responsabilidades e Interfaces.....   | 68 |
| 3.6.3      | Programa de Administración de la Configuración.....   | 68 |
| 3.6.4      | Nomenclatura de Elementos.....  | 69 |
| 3.6.5      | Líneas Base del Proyecto.....   | 69 |
| 3.6.6      | Control de Configuración y Cambio.....  | 70 |
| 3.6.7      | Estado de la Configuración.....   | 70 |
| 3.6.8      | Auditorías y reportes de la configuración.....  | 71 |
| 3.6.9      | Recursos y capacitación.....  | 71 |
| 3.7        | PLAN UTILIZADO EN EL PROYECTO SCADA EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS (UCI).....                 | 71 |
| 3.7.1      | Introducción.....   | 71 |
| 3.7.2      | Programa de Gestión de Configuración.....   | 72 |
| 3.7.3      | Estado de la configuración.....   | 74 |
| 3.7.4      | Entrenamiento y Recursos.....   | 74 |
| 3.7.5      | Proceso de Control Versiones y Gestión Cambios.....   | 74 |
| 3.8        | CONCLUSIONES PARCIALES DEL CAPÍTULO 3.....  | 75 |
| CAPÍTULO 4 | .....   | 76 |

|   |     |
|---|-----|
| PROPUESTA DE PROCEDIMIENTO.....   | 76  |
| 4.1 INTRODUCCIÓN .....  | 76  |
| 4.2 TÍTULO: PLAN DE ADMINISTRACIÓN DE CONFIGURACIÓN Y CAMBIO A NIVEL DE UNIVERSIDAD CENTRADO EN LAS FACULTADES..... | 76  |
| 4.2.1 Historial de Revisiones.....  | 76  |
| 4.2.2 Introducción.....   | 77  |
| 4.2.3 Gestión.....  | 78  |
| 4.2.4 Actividades.....  | 80  |
| 4.2.5 Planificación.....  | 87  |
| 4.2.6 Herramientas, Técnicas y Metodologías.....  | 87  |
| 4.2.7 Mantenimiento del Plan.....   | 88  |
| 4.2.8 Captura y Retención de Registros.....   | 88  |
| 4.3 PROPUESTA DEL PROCEDIMIENTO PARA LA ADMINISTRACIÓN DE CONFIGURACIÓN Y CAMBIO .....                              | 88  |
| 4.3.1 Propuesta de encabezado.....  | 89  |
| 4.3.2 Objetivo.....   | 89  |
| 4.3.3 Alcance.....  | 89  |
| 4.3.5 Referencias.....  | 89  |
| 4.3.6 Responsabilidades.....  | 90  |
| 4.4 INSTRUCCIONES.....  | 90  |
| 4.4.1 Control de versiones de Programas o Software.....   | 90  |
| 4.4.2 Control de versiones de Documentos.....   | 91  |
| 4.4.3 Bibliotecas de Trabajo.....   | 91  |
| 4.4.4 Descripción de las Actividades.....   | 93  |
| 4.4.5 Representación Gráfica de la descripción de las Actividades.....  | 98  |
| 4.7 CONCLUSIONES PARCIALES DEL CAPÍTULO 4.....  | 99  |
| CONCLUSIONES GENERALES.....   | 100 |
| RECOMENDACIONES.....  | 101 |
| REFERENCIA BIBLIOGRÁFICA.....   | 102 |
| BIBLIOGRAFÍA.....   | 103 |
| GLOSARIO DE TÉRMINOS.....   | 106 |

## INTRODUCCIÓN

A lo largo del tiempo el concepto de calidad ha adquirido un carácter multidimensional, debido a que los diferentes autores lo han enfocado desde puntos de vistas diferentes, para alcanzarla se ha pasado por varios estadios, comenzando con la verificación y terminando en la calidad total, pasando por el control y el aseguramiento de la misma. El hecho de que el software constituye un producto de conocimiento de difícil estandarización, hace necesaria la aplicación de estándares de calidad como el Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers, IEEE), la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI) y la Organización Internacional para la Estandarización ((International Standard Organización, ISO) que garanticen lograr un producto final de total eficacia.

Muchas empresas han buscado varias maneras de crear software para comercializar o para uso estratégico en el negocio, por lo que invierten grandes cantidades de dinero en proyectos, estos se enfrentan a largas vidas útiles, durante las cuales el entorno de la organización y las condiciones de los requisitos iniciales varían constantemente.

La promoción de la Industria Cubana del Software (INCUSOFT) en el ámbito internacional ha tenido que adoptar como línea estratégica la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. Nuestro país actualmente cuenta con un alto capital humano calificado que se ha estado preparando y superando a lo largo de estos últimos años para lograr insertar poco a poco a Cuba en el negocio del software, con el objetivo de aumentar los ingresos de la economía cubana.

La Administración de la Configuración del Software (Software Configuration Management, SCM) nace como una de las disciplinas para el control y la administración de la evolución del software, debido a que el programa de computador cambia constantemente, por lo que es necesario mantener la integridad y consistencia del producto. Debido a la aparición de la palabra cambio en el ciclo de vida del software, surge esta disciplina, con la idea de poder establecer mecanismos para tener el control sobre lo que se estaba creando en el ciclo de desarrollo de software, esto fue, porque se tenían sistemas complejos y se quería reducir el tiempo de respuesta al momento de buscar algún producto de trabajo en los proyectos.

Así fue como se creó la Administración de la Configuración de Software queriendo tener un método que pudiera mantener ese control sobre los productos de software.

Por este motivo, surgieron muchas herramientas las cuales se basan en mecanismos de bajo nivel que administran el producto como un agregado de archivos que evolucionan independientemente, obligando a los usuarios (desarrolladores) a manejar externamente los conceptos de configuración y línea base del software.

Hoy en día, ha aumentado la complejidad con la que se desarrollan sistemas de información para la industria, por lo que resulta difícil generar productos que cumplan cabalmente con las expectativas del cliente. Para responder a esta situación han surgido una serie de herramientas, técnicas y modelos que facilitan a las organizaciones generar productos que cumplan las expectativas del usuario e incluso las rebasen, herramientas que prometen ser la solución a los problemas de calidad, costo y tiempo de desarrollo.

Estos problemas implican a que se presente la siguiente **situación problémica**.

- La utilización de herramientas de Administración de la Configuración y Cambio puede ser por conocimiento previo o recomendación, esto es muy serio debido a que el software cambia constantemente en todo el ciclo de vida y cada cambio da lugar a una nueva versión.
- La no existencia de herramientas que controlen la Administración de la Configuración en la Universidad de las Ciencias Informáticas.
- La no existencia de una herramienta que controle todo el proceso de control de cambio en la Universidad de las Ciencias Informáticas, debido a que existe sólo la que controla las versiones por las que pasa el software, como Subversion.
- La ausencia de una política a nivel de universidad, que estandarice la utilización de un procedimiento a seguir.
- La falta de unificación en los equipos de trabajo dentro de los proyectos de desarrollo de software.
- La actualización simultánea de la misma versión/revisión de un ítem, que trae como consecuencia que los cambios de una persona cancelen los de otra.

- El control de cambio puede ser inexistente, poco efectivo, e ineficiente.
- El control de versiones puede ser poco riguroso principalmente en momentos que se necesite una versión anterior.

Toda esta situación antes mencionada nos lleva a definir el siguiente **problema de investigación**.

¿Cómo solucionar algunos de los problemas existentes en la Gestión de Configuración mediante el establecimiento de un eficaz procedimiento de control y aseguramiento de la calidad en la Administración de la Configuración y Cambio dentro de los proyectos de desarrollo de software?

Para responder al problema de investigación, se definió como **objetivo general**.

Proponer un procedimiento que garantice una mejor calidad y un mayor control en la Administración de la Configuración y Cambio.

A través de un análisis del objetivo general resultaron los siguientes **objetivos específicos**:

- Determinar y seleccionar los procedimientos existentes referentes a la Gestión de la Configuración de Software (GCS) en instituciones de desarrollo de software en nuestro país y otras a nivel mundial.
- Escoger y evidenciar que herramientas de software, de las existentes en el mercado pueden ser utilizadas para la Gestión de Configuración del Software (GCS) a partir de las especificaciones del procedimiento a utilizar.

Se plantean **tareas** como:

- Búsqueda bibliográfica referente a la Gestión de Configuración del Software (GCS) y dentro de ella la Administración de la Configuración y Cambio.
- Estudio técnico de las herramientas existentes en la Administración de la Configuración y Cambio.
- Estudio de instituciones que se dedican a la producción de Software como La Casa de Software SIGTA-ETECSA, DESOFT, SOFTEL y la Universidad de las Ciencias Informáticas (UCI) en lo referido a la utilización de un procedimiento para la Gestión de Configuración.

- Análisis y consolidación de toda la información recopilada para proponer un procedimiento a seguir en la Universidad de las Ciencias Informáticas (UCI) en lo relacionado con la Administración de la Configuración y el Cambio.

Entre los **Métodos de trabajo científico** utilizados se destacan los siguientes:

**Método teórico:** Dentro de este método se encuentra el **histórico-lógico**, el cual se utilizó con el objetivo de hacer un estudio detallado de los trabajos anteriores referente a la Gestión de Configuración del Software (GCS), como tesis, cursos en varios países en este caso Chile, y planteamientos del Instituto de Ingenieros de Software (Software Engineering Institute, SEI) referente a esta área de proceso clave, lo cual sirvió de guía para los resultados alcanzados.

**Método empírico:** Se utilizó dentro de este método la entrevista mediante la elaboración de preguntas, con el objetivo de obtener información, acerca del procedimiento o plantillas que emplean las instituciones de desarrollo de software y los proyectos dentro de la Universidad de las Ciencias Informáticas para efectuar el control de la Administración de la Configuración y Cambio, así como preguntas relacionadas con las herramientas utilizadas por estas empresas y estos proyectos para el control de las distintas versiones por la que pasa el software durante todo el ciclo de vida.

Para poder llevar a cabo las tareas enunciadas anteriormente se plantea el siguiente **objeto de investigación:**

Procedimientos y plantillas existentes para llevar a cabo el control de la Administración de la Configuración y Cambio en instituciones de desarrollo de software.

Teniendo en cuenta el objeto de estudio, se define como **campo de acción:**

Procedimientos y plantillas para llevar el control de la Administración de la Configuración y Cambio en la Universidad de las Ciencias Informáticas (UCI).

### **Hipótesis**

Es posible disponer de un control de los elementos de configuración de un software con la utilización de procedimientos y plantillas que reflejen el estado de los mismos mediante el uso de un plan de administración de configuración y cambio.

### **Posible Solución:**

Procedimiento de trabajo que muestre una guía para controlar la Gestión de Configuración de Software (GCS) dentro del ciclo completo de desarrollo de software, así como los requerimientos mínimos que deben tener las herramientas de soporte asociadas a este proceso, teniendo presente:

- Objetivo de este procedimiento.
- Alcance.
- Responsabilidades.
- Control de versiones de Programas o Software.
- Control de versiones de Documentos.
- Herramientas para el control de versiones.
- Bibliotecas de trabajo.
- Proceso de Administración de los Elementos de Configuración.

Con el procedimiento propuesto se garantiza la utilización de herramientas adecuadas en el área de proceso clave asociado a la gestión de la configuración e indica a los desarrolladores de software como llevar a cabo sus tareas.

En esta tesis el capítulo I aborda el tema referente a la Administración de la Configuración y Cambio en cuanto a definiciones. El capítulo II trata de la evaluación de las herramientas existentes respecto a esta área de proceso clave. En el capítulo III se toma como referencia empresas cubanas que se dedican a la producción de software en nuestro país y una a nivel internacional, además de proyectos existentes dentro de la Universidad de las Ciencias Informáticas (UCI) como el proyecto SCADA. En el capítulo IV se realiza una propuesta de un procedimiento a seguir para lograr una mayor calidad en la Administración de la Configuración y Cambio en la Universidad de las Ciencias Informáticas (UCI).

## CAPÍTULO 1

### Fundamentación Teórica de Gestión de Configuración.

#### 1.1 Introducción.

En este capítulo se hace referencia a la historia del arte a nivel internacional, nacional y de universidad, respecto a los procedimientos existentes para controlar la Administración de la Configuración y Cambio, teniendo presente las definiciones planteadas por varias personalidades que se dedicaron a estudiar este tema. También se hace referencia a la Gestión de Configuración del Software (GCS) en los modelos de calidad, la importancia que tiene en el proceso de desarrollo de Software y los aspectos más relevantes de la Administración de Configuración y Cambio.

#### 1.2 Gestión de Configuración. Definiciones.

A continuación mostramos una serie de conceptos fundamentales del tema.

Configuración: Es un arreglo de ítems de configuración. [1]

Configuración del Software: Es el conjunto de toda la información y productos utilizados o producidos en un proyecto como resultado del proceso de Ingeniería de Software (IS). [2]

Software: Probablemente la definición más formal de software es la atribuida por el Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers, IEEE) en su estándar 729: “Suma total de los programas de cómputo, procedimientos, reglas de documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo”. Bajo esta definición, el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente<sup>5</sup>, binario o ejecutable, además de su documentación: es decir, todo lo intangible. [3]

Ítem de Configuración: Es una agregación de componentes de hardware, software o de ambos, diseñada para la Gestión de Configuración y tratada como una sola entidad en dicho proceso de gestión. Puede ser un programa, un archivo de cualquier índole y sus metadatos asociados., un documento, un equipo, o un conjunto de ellos. [1]

De todas las definiciones revisadas para esta investigación, se ha considerado la más completa y a su vez concisa la de Ivar Jacobson, Martín Griss y Patrick Jonson en el libro “Software Reuse: Architecture, Process, and Organizartions for Busisness Success” en la que plantean:

“Gestión de Configuración es el proceso de soporte cuyo propósito es identificar, definir y almacenar en una línea base los elementos de software, controla los cambios, reporta y registra el estado de los elementos y de las solicitudes de cambio; asegura la completitud, consistencia y corrección de los elementos; controla, almacena, maneja y libera los elementos asociados al producto de software”. [6]

En el modelo que plantea la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI) se manifiesta la administración de la configuración como un área de proceso, donde su propósito es establecer y mantener la integridad de los productos de trabajo mediante la identificación, control, reporte y auditoría<sup>13</sup> de la configuración. Es donde se establecen las líneas bases, son las especificaciones o productos desde los cuales se partirá como base del ciclo de desarrollo, se rastrean y controlan los cambios y se establece la integridad de los productos de software.

**Control de Configuración:** Aquella parte de la Gestión de la Configuración de Software (GCS) que se preocupa de evaluar, coordinar, aprobar o rechazar, e implementar los cambios a los ítems de configuración. [1]

**La Administración de Control de Cambios:** Es el proceso de manejar el desarrollo de aplicaciones con seguridad y garantizar que el código fuente<sup>5</sup> sea guardado adecuadamente, manejo de cambios al código fuente<sup>5</sup>, promover los cambios realizados al código fuente a través de las diferentes etapas del desarrollo de la aplicación (AD por sus siglas en inglés), control de versiones, control de releases<sup>11</sup> y modificación de los diferentes niveles del código fuente y finalmente, manejar la etapa de la generación/compilación para crear aplicaciones ejecutables desde el código fuente.

### **1.3 Situación Nacional e Internacional de la Gestión de Configuración.**

A nivel mundial existen muchas empresas que cumplen con las áreas de procesos claves del nivel 2 de la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI), las

cuales utilizan diversos procedimientos de control asociados al tema en cuestión, los cuales se soportan en herramientas de control de versiones y cambio.

Ejemplo de esto es la Asociación Mexicana para la Calidad (AMCIS), los cuales desarrollaron un modelo de procesos equivalentes a la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI), Modelo de Procesos para la Industria de Software (MoProSoft) versión 1.3, el cual trata el tema de las solicitudes de cambio y el registro de cambio, esto fue creado con el objetivo de servir de base a la Norma Mexicana para la Industria de Desarrollo y Mantenimiento de Software. En estas empresas los procedimientos empleados son de información confidencial garantizando que la información interna de la empresa no sea divulgada.

En nuestro país existen entidades que cumplen con todas las áreas de procesos claves del nivel 2 de la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration ,CMM-CMMI), pero no están acreditadas oficialmente, empresas como **La Casa de Software SIGTA-ETECSA** tiene definido un procedimiento llamado Manual para la Gestión de Configuración del Software para llevar el control de la configuración, teniendo en cuenta parámetros como control de versiones de programas y documentos, responsabilidades, alcances, bibliotecas de trabajo entre otros aspectos con el objetivo de organizar el funcionamiento de la institución.

Para obtener información más concreta de cómo está la administración de la configuración y cambio en esta institución, se le realizó una entrevista al directivo a nivel de organización, el cual se encarga de llevar a cabo el control de esta área de proceso clave.

Primeramente decir que en la organización existe un procedimiento pero no está aún institucionalizado, esto trae como consecuencia que el Plan de Configuración no se encuentre aprobado según los procedimientos documentados, por lo que la identificación, control, auditoría y los reportes de estado de la configuración no se ejecutan según el Plan de Configuración para todos los proyectos por igual.

La empresa cuenta con las autoridades responsables del control de la línea base, las cuales generalmente están a cargo del jefe de proyecto. Algunas veces se entrenan a los ingenieros de software,

desarrolladores, etc. en la ejecución del Sistema de Gestión de Configuración. Al realizarse las solicitudes de cambios o reportes de problemas sobre los elementos de configuración no se lleva a cabo todo el proceso de control, es decir: iniciación, análisis, ejecución, almacenamiento, revisión, aprobación y chequeo de dichas solicitudes o informes de problemas según los procedimientos documentados.

Para llevar a cabo el control de la Administración de la configuración y cambio se utiliza como herramienta de control de versiones Visual Source Safe 2005, versión 6.0. A la hora de seleccionar esta herramienta de las tantas existentes en el mercado, se tomó como premisa aspectos importantes como sistema operativo, lenguaje de programación, modelo de repositorio, sencillez, facilidad de uso, integración con herramientas y ambientes de programación.

**DESOFT** es otra empresa dedicada a la producción de software en Cuba, esta institución tiene un procedimiento nombrado Gestión de Configuración, mediante el cual se utilizan plantillas como Lista de Copias Autorizadas, Modelo de Ingreso a Configuración, Nota de Cambio, Registro de Cambios del Proyecto, Registros de Cambios por Soporte y Requerimiento de Cambio, estas sirven de apoyo a la hora de utilizar el procedimiento para controlar la Administración de Configuración y Cambio.

En esta empresa se entrevistó al directivo a nivel de organización el cual se desempeña como desarrollador.

No se cuenta actualmente con los recursos necesarios para ejecutar todo el control de la configuración en los proyectos. No siempre el Plan de Configuración es aprobado según los procedimientos documentados, por lo que la identificación, control, auditoría, y los reportes de estado no siempre se ejecutan según el plan.

El sistema de control de configuración establecido en la empresa si está acorde con las políticas de la organización. Para llevar a cabo la Administración de la Configuración y Cambio se entrenan a los ingenieros de software, a los desarrolladores, es decir a todo el personal involucrado en la ejecución del sistema de Gestión de Configuración.

En la institución existen las autoridades responsables del control de la línea base, aunque dichas líneas bases no son auditadas. En la institución no se controla ni el mantenimiento ni la implementación del software. La correcta ejecución de los procedimientos definidos permite a la institución conocer en cualquier momento el estado de los elementos de configuración. No siempre el jefe de proyecto controla la ejecución del sistema de control de configuración, y al no realizar este control debidamente trae como consecuencia que el grupo de control de calidad no revise de forme periódica ni audite las actividades de dicho sistema.

La empresa se rige por el estándar ISO 9003 y CMMI, pues son los más conocidos y establecidos en la industria del software a nivel internacional. Actualmente la institución no utiliza ninguna herramienta para controlar la Administración de Configuración y Cambio, se está probando el CVS para ver los resultados que trae para la entidad. A la hora de probar dicha herramienta se tuvo en cuenta aspectos como sistema operativo, gestor de base de datos, lenguaje de programación, y muy importante el hecho que no ser propietaria.

**SOFTTEL** es también otra empresa dedicada al desarrollo de software, actualmente están elaborando el procedimiento referente a la Gestión de Configuración, utilizan plantillas como el Plan de Gestión de Configuración, Orden de Trabajo y Solicitud de Cambio.

En esta empresa se entrevistó a la persona que lleva a cabo el control de la configuración, la cual nos proporcionó información de cómo está el control de la administración de la configuración y cambio a nivel de institución.

La compañía se rige por el estándar CMMI, ya que es el más empleado en la actualidad en las empresas de desarrollo de software. Se configuró RUP a las necesidades de la empresa. Se tiene definido personas que ocupan roles específicos como el administrador de configuración, el comité de control de configuración, integrador y revisores. Se tienen establecido líneas bases, específicamente la de producción y operación. Los entregables se almacenan en carpetas, para que sean usados por los que lo requieran, según política de seguridad.

Se utiliza como herramienta de control de versiones Visual Source Safe y Subversion, y para el control de cambio se utiliza Rational Clear Quest porque posee herramientas que integran todo el proceso de desarrollo. A la hora de seleccionar estas herramientas se tuvo en cuenta que la Universidad de las Ciencias Informáticas imparte como proceso de desarrollo RUP en la asignatura de ingeniería de Software,

En la **Universidad de las Ciencias Informáticas (UCI)** actualmente no existe un procedimiento general para controlar la Gestión de Configuración del Software (GCS) a nivel de universidad, pero si existen procedimientos específicos definidos en algunos proyectos, como el proyecto SCADA. Allí se entrevistaron algunos estudiantes que llevan el control de las líneas de trabajo definidas en el proyecto.

El proyecto no se rige por ningún estándar de calidad, pero aún así se lleva a cabo todo el proceso de Gestión de Configuración. Las líneas bases dentro de este proyecto lo constituye el entregable estable de cada mes, sobre el cual se trabaja iterativamente. Se entrega lo de codificación pues se parte del análisis.

El manejo de las versiones de los desarrolladores en la rama unistable se realiza libremente, mientras que las versiones del producto estable la realiza el personal de gestión de configuración y el líder de la línea con coordinación y documentación respectiva. Existe una persona que se encarga de la gestión de configuración y un comité de control de cambio junto con el jefe de línea.

Se utiliza como herramienta de control de versiones Subversion, versión 1.4.2. Para seleccionarla se tomó aspectos fundamentales como sistema operativo, herramientas de integración, precio, modelo de repositorio, interface Web y licencia.

## **1.4 Gestión de Configuración del Software (GCS) en los Modelos de Calidad.**

### **1.4.1 Modelo de Calidad. Concepto.**

La Organización Internacional de Estándares (International Standard Organización, ISO), define la calidad como la ausencia de deficiencias: "Es la totalidad de aspectos y características de un producto o servicio que se refieren a su capacidad para satisfacer necesidades dadas en la adecuación de sus objetivos".

El Instituto de Ingenieros de Software (Software Engineering Institute, SEI) en su modelo de Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI) define la calidad como el grado en el cual un sistema, componente o proceso cumple con los requisitos especificados y el grado en el cual el sistema, componente o proceso cumple con las expectativas del cliente o usuario.

#### **1.4.2 IEEE.**

El Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers, IEEE) 730-1998 es considerado de vital importancia para el Plan de Aseguramiento de la Calidad de Software, puesto que aborda aspectos como la administración, la documentación, el control de código, todo lo referente a la asignación de las responsabilidades, la identificación de las actividades que se realizarán durante todo el proceso, la identificación de la configuración, el reconocimiento de los elementos de configuración, el control de la configuración, el acceso a las bibliotecas, la aprobación o desaprobación de un cambio y la implementación del cambio al ser aprobado (todas estas incluidas entre las tareas de la Gestión de configuración).

#### **1.4.3 Integración del Modelo de Madurez de las Capacidades (CMMI).**

El CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.

CMM para software o SW-CMM no es más que un grupo de lineamientos para el desarrollo de software. Es una descripción de los escenarios por los cuales las organizaciones de software evolucionan, cómo ellos definen, implementan, miden, controlan y desarrollan sus procesos de software.

En diciembre del 2001 surge la Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI) el cual es un enfoque para la mejora de procesos que brinda a las organizaciones los elementos básicos sobre procesos efectivos. Puede ser empleado para guiar la mejora

de procesos a lo largo de un proyecto, división u organización. CMMI ayuda a integrar funciones organizacionales tradicionalmente separadas, otorga metas y prioridades en la mejora de procesos, brinda además una guía para los procesos específicos de calidad y un punto de referencia para la valoración de los procesos actuales. Cuenta al igual que CMMI con cinco niveles, aunque algunos cambian de nombre.

### **Niveles de madurez.**

La Integración del Modelo de Madurez de las Capacidades (Capability Maturity Model Integration, CMMI) es un modelo escalonado que define 5 niveles posibles de madurez para las organizaciones que desarrollan y mantienen software como:

- Nivel1 (Inicial).
- Nivel2 (Repetible).
- Nivel3 (Definido).
- Nivel4 (Gestionado).
- Nivel5 (Optimizado).

En el presente trabajo se hace referencia al nivel 2 de CMMI porque incluye como área de proceso clave la Gestión de Configuración. CMMI tiene una representación continua y una escalona, la representación continua se analiza con el objetivo de obtener el nivel de capacidad con que cuentan las empresas en esta área de proceso clave.

### **Nivel 2: Repetible.**

Se considera un nivel 2 de madurez cuando se llevan a cabo prácticas básicas de gestión de proyectos, gestión de requisitos, control de versiones y de los trabajos realizados por subcontratistas. Para obtener Nivel 2 de CMMI (Gestionado) es necesario tener definidos los siguientes procesos.

Este nivel consta de 5 áreas de procesos claves como:

- Gestión de Requisitos.
- Planificación del proyecto de software.
- Seguimiento y Supervisión del proyecto.
- Gestión de subcontratos de software.

- Garantía de calidad de software.
- Gestión de la configuración del software.

Por lo que se puede decir que la Gestión de la Configuración del Software (GCS) es la administración y control de los ítems que conforman el proyecto, es el proceso más largo y solo culmina cuando el software es retirado de circulación, requiere una organización impecable de los componentes en desarrollo. El objetivo de la gestión de la configuración es establecer y mantener la integridad de los elementos de trabajo identificando, controlando y auditando dichos elementos.

Más concretamente mediante:

- La identificación de los elementos de trabajo que componen una línea base.
- Controlando los cambios de dichos elementos.
- Proporcionando formas de construir los elementos de trabajo a partir del sistema de control de la configuración.
- Manteniendo la integridad de las líneas base.
- Proporcionando información precisa de los datos de la configuración a desarrolladores y clientes.

En la versión 1.2 de CMMI que propone el Instituto de Ingenieros de Software (Software Engineering Institute, SEI) respecto a Gestión de Configuración se hace referencia a nuevos parámetros como Metas y Prácticas Específicas y Genéricas las cuales incluyen sub-prácticas que permiten llevar el control de la GCS.

#### **1.4.4 La Organización Internacional de Estándares (International Standard Organización, ISO).**

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios como de aplicaciones [ISO, 2005]. El objetivo del estándar es desarrollar un código mínimo que contenga prácticas de administración para garantizar el Aseguramiento y Administración de la Calidad, es decir, qué hacer para responder a los requerimientos de un mercado cada vez más competitivo y cómo deben responder los proveedores y compradores respecto a la calidad de los bienes o servicios intercambiados. Para ello establece una serie

de guías para la selección y uso del estándar deseado, así como aclara conceptos en cuanto a la calidad y las interrelaciones que se establecen.

Relacionado con los procesos de software ha sido implementado el modelo de evaluación y mejora del proceso de software ISO 9000, específicamente la guía ISO 9000-3.

Entre los aspectos que tiene en cuenta la Norma ISO 9000-3 se encuentran:

- Sistema de calidad.
- Especificación de los requisitos del comprador.
- Planificación del desarrollo.
- Planificación de la calidad.
- Pruebas y validaciones.
- Gestión de Configuración de Software.
- Control de documentos.
- Mediciones.

El hecho de que una norma de calidad tan reconocida a nivel mundial incluya dentro de sus requerimientos, el de efectuar las labores de Gestión de Configuración del Software (GCS), es plenamente consistente con la importancia que le otorgan los diversos autores y que tiene merecida.

La norma ISO 10007, define el objetivo principal de La Administración de la Configuración de Software (Software Configuration Management, SCM) como: Documentar y proveer visibilidad de los productos de software y del estado de progreso en la satisfacción de los requerimientos funcionales y físicos.

### **1.5 Proceso Unificado de Rational (Rational Unified Process, RUP).**

Proceso Unificado de Rational (Rational Unified Process, RUP), es un proceso o metodología definida dentro de Ingeniería de Software (IS) para la construcción completa del ciclo de ingeniería de software. Permite la productividad en equipo y la realización de mejores prácticas a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo del software.

Comprende 9 disciplinas dentro de las cuales 6 son flujos de trabajo y 3 son de apoyo, como Modelamiento del Negocio, Requerimientos, Análisis y diseño, Implementación, Prueba, Despliegue, **Administración de la Configuración y Cambio**, Administración de Proyecto y Ambiente. Dentro del RUP se llevan a cabo subprocesos referente al desarrollo de un software en equipo para lograr una mejor calidad en la liberación del producto como:

- Selección de la versión: El propósito de la buena 'selección de la versión' es asegurarse que las versiones correctas de los elementos de la configuración están seleccionadas para el cambio o implementación. La selección de la versión se basa en una fundamentación sólida de la 'identificación de la configuración'.
- Contabilidad de estado de la Configuración: Se utiliza para describir el estado del producto basado en tipo, número, precio y la gravedad de los defectos encontrados y fijados, durante el curso del desarrollo del producto.
- Administración de Cambio: Describe la estructura del producto e identifica sus elementos de configuración que son tratados como entidades versionables en el proceso de Administración de la Configuración. La Administración de Cambio se ocupa de definir configuraciones, construcción y etiquetado, además de mantener la capacidad de análisis y rastreo entre estas versiones.
- El seguimiento del cambio: describe qué se hace a los elementos por qué razón y cuándo. Sirve como historia y razón fundamental de cambios. Determina el impacto de los cambios propuestos según lo definido en el proyecto.

### **1.6 Gestión de Configuración. Importancia.**

Esta disciplina controla los cambios sobre los productos de trabajo que se están creando, permitiendo verificar versiones anteriores, realizar alguna consulta en específico, nos da orden sobre la manera en cómo organizamos los archivos, dónde guardamos cada cosa, cómo la nombramos para identificarla más rápidamente y cómo vamos versionándola, ayudando con esto a reducir el tiempo de re-trabajo en el desarrollo de software.

Además garantiza repositorio centralizado de entregables, acceso controlado y seguro, actualización simultánea prohibida o administrada, una sola copia física de cada entregable y manejo de versiones. Sin

la implantación de la Gestión de Configuración no se forjará el éxito de una institución ya que es la raíz de un buen proyecto y la solución satisfactoria a los problemas.

Valoraciones de diversos autores respecto a la importancia de la Gestión de Configuración dentro del desarrollo de software.

Primeramente hacer referencia a una frase escrita por Tom Milligan asesor de IBM Rational Software la cual expresa el papel significativo que tiene la GCS, planteando que: “La Gestión de Configuración de Software es el héroe no nombrado en el proceso de desarrollo de software”.

Watts Humphrey, en el libro, Administrando el Proceso de Software plantea que en el desarrollo de software, los problemas más frustrantes, son causa de un pobre proceso de Gestión de Configuración, y resultan frustrantes porque toman tiempo para solucionarlos, ocurren generalmente en momentos críticos y su ocurrencia es totalmente innecesaria. [8]

No debe dejarse de mencionar la Ley Fundamental de la Gestión de Configuración del Software, donde se define su papel en el proceso de desarrollo.

“La Gestión de Configuración del Software (GCS) es el fundamento de un proyecto software. Sin ella, no importa cuan talentoso sea el equipo, cuan grande sea el presupuesto, cuan robusto sean los procesos de desarrollo y prueba, o cuan superior sean las herramientas de desarrollo técnicamente, la disciplina del proyecto colapsará y se perderá la posibilidad del triunfo. Haz bien la Gestión de Configuración, u olvídate de avanzar en el proceso de desarrollo de Software”.

## **1.7 Administración de la Configuración.**

### **1.7.1 Requerimientos para el éxito.**

- Debe existir una planificación, un Plan de Gestión de Configuración: aprobado por todos los niveles de gestión, ampliamente difundido y claramente establecido.
- Todos los elementos constitutivos del proyecto (documentos, modelos, código fuente<sup>5</sup>, librerías,...) deberán estar almacenados en un mismo repositorio.

- Deberán hacerse copias de resguardo del repositorio periódicamente.
- Cada uno de los elementos mencionados deberá encontrarse bajo control de versiones.
- Para cada cambio en cualquiera de los elementos, deberá guardarse la fecha y hora del cambio y una breve descripción del mismo.
- Se deberán trazar líneas bases para marcar un conjunto de elementos en un determinado estado del proyecto (generalmente luego de finalizado un proceso durante el desarrollo del sistema).
- Se deberá poder retroceder a cualquier línea base recuperando todos los elementos del proyecto en el estado en el que estaban al momento de trazada la línea base.
- Se deberá poder efectuar una ramificación en los elementos del proyecto, con el fin de hacer desarrollo en paralelo de distintas alternativas (sobre todo durante el proceso de Construcción del Sistema).
- Se deberán poder obtener listados o reportes con información sobre los cambios en los elementos (elementos que sufrieron más cambios, cambios entre dos líneas base,...). [1]

### 1.7.2 Roles y responsabilidades.

#### Alta Gerencia:

- Establece (aprueba) la(s) política(s) de Administración de la Configuración a implantar, y controla continuamente su aplicación.
- Identifica y define los ítems de configuración en un sistema.
- Controla la liberación y cambio de esos ítems por todo el ciclo de vida del sistema.
- Registra y reporta el estado de los ítems de configuración y de los requisitos de cambio.
- Verifica la completitud y correctitud de los ítems de configuración.
- La alta gerencia cubre el desarrollo completo del producto, desde la definición del ámbito de trabajo, estimación y planificación hasta el monitoreo y control de los procesos productivos. [1]

#### Responsable de Configuración se dedica a:

- Desarrollar, documentar y distribuir procedimientos de Gestión de Configuración del Software (GCS).
- Establecer la línea base del sistema, incluyendo provisiones para respaldo

- Asegurar que:
  - No se realizan cambios no autorizados a la línea base.
  - Todos los cambios a la línea base se registran con detalle suficiente como para reproducirlos o volverlos atrás.
  - Todos los cambios a la línea base pasaron test de regresión<sup>1</sup>.
- Proporcionar información para resolución de problemas.
- Debe estar capacitado en herramientas de planeación para ejecutar sus actividades de forma eficiente y eficaz.
- Debe tener un control sobre lo que cada programa debe de tener, que productos va a desarrollar cada programador y que productos se van a ir desarrollando a lo largo de cada proyecto, que componentes estará necesitando cada desarrollador a lo largo del proyecto ya sean del proyecto mismo o de otros.

#### **Comité de Control de Configuración (CCC).**

- Puede ser formado por una sola persona (analista o gestor del proyecto) o por varias personas (gestor de calidad, el cliente y gestor del proyecto).
  - Valora las ventajas e inconvenientes de la introducción del cambio (beneficios coste de la introducción).
  - Evalúa el impacto de la modificación sobre los parámetros del proyecto (agenda costes riesgos).
  - Determina su prioridad cuando y como debe llevarse a cabo y como deberá comprobarse y verificarse su implementación.
  - Recibe, estudia y aprueba las solicitudes de cambio en el software que son presentadas ya sea por los usuarios o por los propios encargados del mantenimiento.

#### **1.7.3 Control de versiones.**

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. Una versión de un ítem de configuración indica que tiene un conjunto definido de

capacidades funcionales. A medida que se añaden, modifican o eliminan capacidades funcionales, es decir se tiene una nueva versión del ítem. [2]

Revisiones: Son las distintas versiones que aparecen en el tiempo, según se va avanzando en el desarrollo de un elemento. Una revisión se asocia con la noción de corrección de errores es decir realizar cambios a los ítems solo para corregir errores sin afectar las capacidades funcionales ya que ninguno de sus requerimientos ha cambiado. [2]

La manera más fácil de crear una nueva revisión de un elemento de configuración (ECS) 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. [2]

#### **1.7.3.1 Identificación de la versión y la revisión.**

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 grado de evolución o grafo de revisión. El esquema de identificación debe:

- Permitir identificar fácil y, ojalá, automáticamente las distintas versiones/revisiones de un mismo ítem.
- Facilitar la clara identificación de las versiones / revisiones de cada promoción y/o liberación de los ítems.

El esquema más utilizado es:

- Un dígito entero para la versión.
- Uno o dos dígitos decimales para las revisiones.

#### **1.7.3.2 Árbol de versiones.**

Un ítem de configuración puede no tener versiones, o tener más de una. La jerarquía de versiones de un ítem de configuración se puede ver como un árbol, el identificador de la versión esta relacionado con una versión padre, de esta manera se mantiene una jerarquía, por lo que definimos como árbol de revisión a:

Un documento que define la estructura de una versión/revisión particular de un producto ya liberado para su uso operacional. Identifica a cada uno de los ítems de configuración que lo componen y la versión / revisión de cada uno de ellos.

Se utiliza cuando las modificaciones a un producto dan origen a distintas versiones que deben coexistir:

- Debido a liberaciones incrementales de los productos.
- A la comercialización de las distintas versiones/revisiones.

Contenido:

- Inventario de los ítems de configuración que componen la versión/revisión del producto.
- Identificación (incluyendo versión/revisión) de cada ítem.
- Identificación y descripción de los cambios incorporados.
- La información de instalación y operación aplicable en forma exclusiva a la versión/revisión descrita. [1]

#### **1.7.4 El proceso de Administración de la Configuración.**

##### **Elementos de Configuración del Software. Definición.**

Elemento de Configuración del Software (ECS): Es cada uno de los componentes de la configuración del software. [2]

##### **Los Elementos de Configuración del Software son los siguientes:**

1. La especificación del sistema.
2. El plan del proyecto software.
3. La especificación de requisitos software.
4. Un prototipo, ejecutable o en papel.
5. El diseño preliminar.
6. El diseño detallado.

7. El código fuente.
8. Programas ejecutables.
9. El manual de usuario.
10. El manual de operación e instalación.
11. El plan de pruebas.
12. Los casos de prueba ejecutados y los resultados registrados.
13. Los estándares y procedimientos de ingeniería de software utilizados.
14. Los informes de problemas.
15. Las peticiones de mantenimiento.
16. Los productos hardware y software utilizados durante el desarrollo.
17. La documentación y manuales de los productos hardware y software utilizados durante el desarrollo.
18. Diseños de bases de datos.
19. Contenidos de bases de datos.

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

En cuanto al software propiamente dicho, dependiendo de su tamaño, complejidad y necesidad de control y visibilidad sobre el mismo, puede requerir de su descomposición en varios ECS, aunque el sistema en su conjunto será a su vez un ECS.

#### **1.7.4.1 Selección de los Elementos de Configuración del Software.**

La selección de un número adecuado de elementos de configuración del software (ECS) 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 ECS 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.

Sin embargo, el tener pocos elementos de configuración del software (ECS) puede hacer que no se tenga la suficiente visibilidad sobre el producto.

Al seleccionar los ECS hay que tratar de separar en ECS distintos las funcionalidades distintas, para tratar de minimizar el impacto de los cambios. Si un ECS incluye funciones muy diversas es probable que aumente el número de veces que será necesario repetir el proceso de reconstrucción del elemento y liberación de una nueva versión. [2]

#### **1.7.4.2 Identificación de la Configuración.**

Identificación de la Configuración: 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: Es una tarea crítica, difícil, pero importante y necesaria.

La tarea de Identificación engloba o puede englobar varias 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 bases.
- Definición y Establecimiento de bibliotecas de software.

#### 1.7.4.3 Definición de un Esquema de Identificación.

Es necesario decidir también cuál es el método que se va a utilizar para identificar de forma unívoca cada elemento de configuración del software (ECS). Dicho en otras palabras, es necesario establecer un esquema de identificación que permita etiquetar cada uno de los elementos de configuración del software.

Sea cual sea, el esquema de identificación utilizado debe proporcionar al menos la siguiente información:

- Número o código del Elemento de Configuración del Software.
- Nombre del ECS: Cadena de caracteres.
- Descripción del ECS: Tipo, ID del proyecto, versión, cambio.
- Autor/es del ECS.
- 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 subfase en la que se creó.
- Tipo de Elemento de Configuración (documento, programa, elemento físico).
- Localización.

Por otro lado, el esquema de identificación debe permitir diferenciar entre las diferentes versiones de un mismo elemento de configuración del software (ECS), puesto que los elementos de configuración del software van a evolucionar a lo largo del ciclo de vida. Para ello, se suelen utilizar los siguientes campos:

- Número de versión.
- Fecha de la versión. Toda esta información puede ir codificada y agrupada en un único código de identificación o puede ser referenciada como partes separadas. A la hora de definir un código, se puede elegir entre dos tipos de sistemas de codificación:
  - Significativos.
  - No significativos (por ejemplo, asignar números consecutivos).

Una codificación no significativa requiere del establecimiento y mantenimiento de un registro de asignación de códigos. Aunque es más sencilla la selección del código, tiene la desventaja de que el código no da ninguna información acerca del tipo de ECS de que se trata, o cualquier otra. Una

codificación significativa puede ser muy simple o muy compleja, dependiendo del tipo y cantidad de información que codifica acerca del ECS. La ventaja es que no es necesario mirar en el registro de asignación de códigos para saber de qué ECS se trata. [2]

El esquema de identificación:

Identifica los ítems de configuración:

- Lista de entregables por cada fase del proyecto (internos y para el cliente).
- Construida con un criterio técnico, de largo plazo y optimizando la relación costo / beneficio.
- Que refleje cabalmente la estructura del producto y del proceso de desarrollo.
- Revisada periódicamente.

Asigna nombres a los ítems de configuración:

- Unívocos e inequívocos.
- Prácticos, ojala nemotécnicos.
- Implementables (tiene que ser práctico).
- Trazables (identificar al ítem en todos los procesos):
  - Almacenamiento.
  - Recuperación.
  - Seguimiento.
  - Reproducción.
  - Distribución.

Documenta características para el control:

- Cuando ingresarán los ítems a control.
- En que estado lo harán.
- Que características físicas y funcionales deben cumplir. [1]

### **1.5 Línea Base.**

Línea base o baseline: Es una especificación, resultado, documento, acuerdo o producto, o conjunto de ellos, fijo en un momento determinado del desarrollo (fotografía), y cuyos cambios requieren justificación y aprobación. Son un mecanismo muy efectivo y eficiente para sincronizar a los diferentes equipos de

trabajo de un mismo proyecto. Se establecen con una complejidad creciente y permiten determinar el progreso del proyecto y su estado técnico. [2]

Según el Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers, IEEE) la Línea Base es 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.

### **1.5.1 Alcance de la Línea Base para el control de código.**

A continuación se muestra los distintos niveles de alcance de las líneas base.

- Nivel actual de cada módulo (fuente y objeto).
- Nivel actual de cada caso de prueba (fuente y objeto).
- Nivel actual de cada herramienta usada.
- Nivel actual de cada prueba especial o datos operacionales.
- Nivel actual de software auxiliar (bibliotecas, archivos).
- Nivel actual de procedimientos de instalación u operación.

### **1.5.2 Establecimiento de líneas base.**

Las líneas bases se establecen en hitos previamente especificados a lo largo del proceso de desarrollo. Es uno de los primeros pasos para poder efectuar la actividad de Identificación de la Configuración, por lo tanto, consiste en definir cuáles van a ser los hitos, dentro del proceso de desarrollo en los que se va a establecer una línea base. Cuando se crean líneas bases se logra que todos los desarrolladores estén sincronizados unos con otros. Lo más corriente es implantar líneas bases secundarias al final de cada iteración y líneas bases principales al final de cada fase. Se necesita hacer una distinción en el momento de crear una línea base teniendo:

- Una línea base con TODAS las versiones de los ficheros y directorios que han sido modificados en el subsistema o subsistemas o,
- Una línea base con una UNICA versión de todos los ficheros y directorios de todos los subsistemas.

Las mismas se establecen con dos objetivos:

- Identificar los resultados de las tareas realizadas durante la fase.
- Asegurar que se ha completado la fase.

Las líneas bases más comunes son:

- Línea Base de Requerimientos: Se establece al final de la fase de Especificación de Requerimientos. Contiene todos los requerimientos aprobados. Está a cargo del Responsable de la Configuración.
- Línea Base Funcional: Se establece al finalizar la fase de análisis y especificación de los requisitos del sistema, y comprende todos aquellos documentos en los que se define el problema a resolver, los costes del proyecto, el plan de tiempos, y los diferentes requisitos funcionales, de interoperabilidad y de interfaz del sistema.
- Línea Base de Diseño: Se establece al finalizar la fase de diseño detallado o de arquitectura. Comprende todos aquellos documentos que contienen el diseño detallado del software y el plan de implementación, y también la descripción de los casos de prueba. Está a cargo del Responsable de Configuración.
- Líneas Base de Desarrollo: Se establece en puntos específicos de la fase de Implementación y Pruebas. Lo más común es al inicio de la integración. Contienen el software y la documentación técnica asociada. Están a cargo de los desarrolladores. Se usan en desarrollos de gran envergadura.
- Línea Base de Producción: Se establece al finalizar la fase de pruebas de Aceptación. Comprende los programas creados y todos aquellos documentos que contienen la información relativa a las pruebas realizadas, contiene el código fuente<sup>5</sup>, el código ejecutable, y todos los componentes asociados que definen un ítem durante la producción, operación, mantención y soporte. Está a cargo del Comité de Control de Configuración.

- Línea Base de Operación: Se establece al finalizar la fase de implantación. Comprende los manuales de usuario, guías de operación y mantenimiento, manuales de formación, etc.

Hay autores que sólo consideran las líneas base Funcional, de Asignación, y de Producto, más una Línea Base de Desarrollo, que va evolucionando a lo largo del desarrollo, y que incluiría la Línea Base de Diseño Preliminar y la Línea Base de Diseño.

Una línea base se puede establecer de dos formas:

- Físicamente: Etiquetando cada elemento de configuración del software (ECS) 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.

La idea consiste en permitir cambios rápidos e informales sobre un elemento de configuración del software (ECS) antes de que se pase a formar parte de una línea base, pero en el momento en que se establece una línea base se debe aplicar un procedimiento formal para evaluar y verificar cada cambio. [2]

### **1.6 Organización del ambiente controlado en la Gestión de Configuración del Software.**

A continuación se hace referencia a la organización que debe existir a la hora de querer mantener un buen control de la Administración de la Configuración y Cambio.

- La plataforma de desarrollo.
- Las herramientas de automatización disponibles.
- La experiencia del equipo de desarrollo.
- La experiencia del equipo de gestión.
- La planificación de las actividades de Gestión de Configuración.

La estructura más utilizada es la de **librerías**, que permiten:

- Identificar y clasificar los ítems en control de configuración para una determinada línea base.
- Conocer y realizar el seguimiento de control al estado de los cambios a dichos ítems.

La cantidad y tipo de las librerías depende de:

- La envergadura y complejidad del desarrollo.

- Las variaciones en los derechos de acceso.
- Las necesidades de los usuarios.

Dependiendo de la automatización y de la plataforma, se puede:

- Almacenar, en una sola librería, ítems variables en su forma física.
- Tener librerías lógicas en sólo una librería física.

### 1.6.1 Librería de Software. Definición.

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 de Software (GCS), especialmente en cuanto al Control de Cambios y la Contabilidad de Estado. [1]

### 1.6.2 Tipos de librerías de software.

A continuación se mencionan las librerías existentes dentro de la Gestión de Configuración del Software (GCS).

- **Librería 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<sup>21</sup>. 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 al Proyecto. En esta librería el control de cambios es informal.
- **Librería de Integración:** Es de esta librería de donde se toman los ECS para su integración en ECS de nivel superior del sistema.
- **Librería de Soporte al Proyecto:** En esta librería se almacenan los ECS aprobados y transferidos desde la librería de Trabajo o desde la librería de Integración. Cuando un elemento pasa a esta librería, se encuentra sujeto a un control de cambios interno y semiformal. Los procesos de alta y baja de la librería de Proyecto implementan dos elementos importantes del Control de Cambios: el control de acceso y el control de sincronización.

El Control de Acceso se refiere a los derechos que tienen los diferentes miembros del equipo de desarrollo para acceder y modificar ECS concretos y el Control de Sincronización ayuda a asegurar que los cambios en paralelo, realizados por equipos o personas diferentes, no se sobrescriban. Así, cuando un ECS se da de baja de la librería de Soporte, el Control de Sincronización bloquea el objeto para que no se puedan hacer más actualizaciones sobre él hasta que se haya reemplazado con una nueva versión.

- **Librería Maestra:** Se usa para almacenar ECS liberados para su entrega al cliente o distribución en el mercado. Los elementos en la librería maestra se encuentran sujetos a un control de cambios formal y estricto. Y normalmente esta librería tiene fuertes restricciones de acceso para escritura, aunque normalmente no los tiene para lectura. En esta librería se almacenan los Releases<sup>11</sup> (Es una configuración del sistema que se va a comercializar o entregar al cliente) del sistema. Mantiene las unidades y componentes de un ítem de configuración ya desarrollados y probados, a la espera de su integración, es de acceso controlado, pero se pueden obtener libremente copias de su contenido, y es controlada por el Responsable de la Configuración.
- **Librería de Software o Repositorio:** Es la entidad en la que se archivan los ECS de un proyecto tras su cierre. Se transfieren a él desde la librería maestra. 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. Es central y común a todos los proyectos, mientras que la librería de Producción y la Maestra son individuales para cada proyecto. El repositorio permite ahorrar espacio de almacenamiento, evitando guardar por duplicado elementos comunes a varias versiones o configuraciones.

- **Librería dinámica o de programación:** Mantiene las entidades de **software** recientemente creadas y/o modificadas. Se usa para administrar las líneas bases, es de acceso libre a los desarrolladores y es controlada por los desarrolladores.
- **Librería estática o producción:** Mantiene las copias maestras y las autorizadas de los ítems de configuración liberados para uso operacional. Se usa para administrar las líneas base liberadas para uso general. Es de acceso restringido, y se pueden obtener copias solo con la debida autorización y es controlada por un Comité de Control de Configuración. [2]
- **Librería de calidad:** Su función principal consiste en almacenar los resultados de las revisiones técnicas formales y de las auditorías. Será de acceso controlado (Grupo de Aseguramiento de la Calidad y Responsable de Configuración) y se podrán realizar copias libres.

### 1.7 Control de Cambios en la Configuración.

El control de cambio es la actividad de Gestión de Configuración del Software (GCS) 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.

#### 1.7.1 Tipos de cambios.

- Corrección de un defecto: Los clientes tienden a clasificar todos los cambios en esta categoría.
- Mejora del sistema: Los programadores, sin embargo, los suelen clasificar aquí.
- Mejora de funcionalidades existentes.
- Mejora del desempeño.
- Desarrollo de nuevas funcionalidades.
- Adaptaciones de nuevas plataformas.
- Nuevos requerimientos del usuario.

### 1.7.2 Niveles de control de cambios

- Control de cambios informal: Control de cambios informal: Se realiza antes que el elemento de configuración del software (ECS) pase a formar parte de una línea base, es efectuado por aquel que haya desarrollado el elemento de configuración del software (ECS), por lo que podrá realizar cualquier cambio justificado sobre él.
- Control de cambios al nivel del proyecto o semi-formal: Una vez que el elemento de configuración del software (ECS) pasa a la revisión técnica formal y se convierte en una línea base, para que el encargado del desarrollo pueda realizar un cambio debe recibir la aprobación de:
  - El director del proyecto, si es un cambio local.
  - El Comité de Control de Cambios, si el cambio tiene algún impacto sobre otros elementos de configuración del software.

En un proceso semi-formal, puede suprimirse la necesidad de generar la solicitud de cambio, el informe de cambio y la orden de cambio, pero sí debe realizarse la evaluación del cambio y su seguimiento.

- Control de cambios formal: Se suele adoptar una vez que se empieza a comercializar el producto, cuando se transfieren los ECS a la librería maestra. Todo cambio deberá ser aprobado por el Comité de Control de Cambios.

En cualquier caso es necesario establecer de forma precisa, al comienzo de cada proyecto, cuál será el proceso de control de cambios que se va a utilizar.

### 1.7.3 Roles y responsabilidades.

#### **Comité de Control de Cambios:**

Es una persona o grupo encargado de tomar las decisiones finales acerca del estado y la prioridad de las peticiones de cambio.

- Su obligación es tener una visión general del producto para poder evaluar el impacto de cada cambio en un determinado elemento de configuración del software sobre otros elementos de configuración.

- Evalúa el impacto sobre la calidad del producto, su rendimiento, su fiabilidad.
- Evalúa la visión que el cliente tiene del producto, esta labor se suele delegar en el Comité de Revisión del Diseño.

Todos los miembros de un proyecto:

- Pueden solicitar cambios.
- Serán los encargados de realizar los cambios.
- Deben informar acerca del estado de los cambios que están realizando.

#### **El Responsable del Producto:**

- Se debe asegurar de que los procedimientos de Control de Cambios se siguen correctamente.
- Puede participar en la evaluación de los cambios, sobre todo en la determinación de impactos sobre otros elementos software, del coste del cambio y de su efecto sobre la programación del proyecto.
- Informa acerca del estado de los cambios.

#### **1.7.4 El proceso de control de cambio.**

No hay ningún estándar para el control informal o interno de los cambios, aunque sí hay algunas recomendaciones.

En cuanto al control de cambios formal, se puede estructurar de muchas formas. Se muestra el proceso que habría que seguir para hacer un cambio sobre una línea base:

1. Iniciación del Cambio.
2. Clasificación y registro de la solicitud de cambio.
3. Aprobación o rechazo inicial de la solicitud de cambio.
4. Evaluación de la solicitud de cambio.
5. Se presenta el Informe de Cambio al Comité de Control de Cambios.
6. Se realiza el cambio, entrando en un proceso de seguimiento y control.
7. Una vez finalizado el cambio, se certifica, mediante una revisión, que se ha efectuado correctamente el cambio.
8. Se notifica el resultado al originador del cambio.

### 1.8 Conclusiones Parciales del Capítulo 1.

- Puede observarse la importancia que le otorgan a la Gestión de Configuración del Software (GCS) los diferentes autores como Watts Humphrey y dentro de ella se analiza la identificación de la configuración por ser la que permite reconocer los elementos de configuración durante todo el desarrollo de un proyecto.
- Puede verse el papel que juega esta disciplina dentro de los estándares más utilizados a nivel mundial como IEEE, CMMI e ISO acentuando que sin la Gestión de Configuración del Software (GCS) no se puede llevar con éxito un buen proyecto.
- Se evidencia a la Gestión de Configuración del Software (GCS) como la disciplina que lleva a cabo la identificación de la configuración, el control del cambio y el control de versiones como una parte esencial para el buen funcionamiento de esta área de proceso clave.

## CAPÍTULO 2

### Evaluación de Herramientas

#### 2.1 Introducción.

En este capítulo se hace referencia a las herramientas existentes en esta área de proceso clave, teniendo como objetivo principal las ventajas, desventajas, características principales que estas proporcionan a los proyectos de desarrollo de software. Así como una comparación de las mismas que facilitará una panorámica de cuáles serán las propicias a utilizar en los proyectos, en dependencia a los requerimientos planteados.

#### 2.2 Herramienta Rational Clear Case.

Rational Clear Case es la herramienta líder en el Mercado para la solución de la Administración de la Configuración de Software (Software Configuration Management, SCM), se basa en el control de versiones para archivos, directorios y otros activos de desarrollo. Ofrece herramientas y procesos que se pueden implementar de inmediato y personalizar a medida que se evoluciona.

##### 2.2.1 Ventajas.

- Provee una gestión fiable, ampliable y flexible de los activos de software para equipos de desarrollo de gran tamaño y tamaño medio.
- Simplifica el proceso de cambio.
- Proporciona seguimiento a los cambios a cada archivo y directorio, manteniendo una historia completa y anotada de versiones de activos, incluyendo código fuente<sup>5</sup>, binarios, ejecutables, documentación, test scripts y librerías.
- Proporciona las funciones necesarias para crear, actualizar, ofrecer, reutilizar y mantener los activos más importantes del negocio mediante un control integrado de versiones, una gestión del espacio de trabajo automatizado, un soporte de desarrollo en paralelo incluyendo bifurcaciones<sup>2</sup> automáticas en distintas plataformas, una gestión de línea base y gestión de builds<sup>4</sup> y releases<sup>11</sup>.

### 2.2.2 Integración.

- IDEs líderes (WebSphere Studio, Microsoft Visual Studio .NET y el framework<sup>3</sup> open source<sup>18</sup> de Eclipse).
- Rational Suite y Rational Team Unifying Platform.
- Herramientas para desarrollo Web.
- Control de versiones de todos los activos de desarrollo.
- Rational Clear Quest para obtener una solución completa de Gestión de la Configuración del Software (GCS).
- Soporta Oracle, Microsoft SQL Server, SQL Anywhere, y bases de datos Microsoft Access.

### 2.2.3 Características principales.

- Modelo de desarrollo Check in (Protección de un elemento en el control de versiones bajo el control del sistema)/check out (Operación que desprotege un elemento en un control de versiones para permitir su modificación).
- Versionado de directorios, subdirectorios y todos los objetos del sistema de archivo.
- Base de datos de objetos versionados seguros, que garantiza que solo usuarios acreditados. puedan acceder a los archivos y solo aquellos a los que los habilita su perfil.
- Soporte de integración para productividad individual y de equipo.

### 2.2.4 Productos de Rational Clear Case.

A continuación se muestran los siguientes productos que contiene Rational Clear Case para llevar a cabo la Gestión de Configuración del Software (GCS).

- **IBM Rational Clear Case Change Management Solution Enterprise:** La Administración de la Configuración de Software (Software Configuration Management, SCM) integrada para proyectos de desarrollo de porte mediano a grande, y/o equipos distribuidos geográficamente.
- **IBM Rational Clear Case Change Management:** Administración de la Configuración de Software (Software Configuration Management, SCM) integrada para proyectos de desarrollo de porte mediano a grande.
- **IBM Rational Clear Case LT:** Control de versión en el nivel de entrada para proyectos pequeños, locales.

- **IBM Rational Clear Case MultiSite<sup>23</sup>**: Una opción al Rational Clear Case para soportar administración de activos de software geográficamente distribuida, para proyectos de porte mediano a grande.

### 2.3 Herramienta Rational Clear Quest.

Rational Clear Quest es una herramienta clave como componente del Rational Suite, es un sistema flexible para detectar, asignar y rastrear defectos para cualquier tamaño de equipo, incluyendo soluciones como:

- Administración de cambios.
- Pruebas.
- Administración de requerimientos.
- Modelamiento visual.
- Guía de procesos.

Rastrea nuevas características, defectos y requisitos de cambios para los miembros del equipo y otras personas involucradas en un formato que es accesible para todos. La interface Web de Rational Clear Quest hace que ésta sea una herramienta ideal para ambientes de desarrollo de plataformas múltiples por lo se puede acceder fácilmente desde cualquier navegador web estándar.

Con Rational Clear Quest y Rational Clear Case, se puede usar Administración Unificado del Cambio, conocido en ingles como (Unified Change Management (UCM)) para:

- Administrar requisitos cambiantes.
- Diseñar modelos.
- Diseñar documentación.
- Diseñar componentes.
- Diseñar pruebas de casos.
- Diseñar código fuente<sup>5</sup>.

#### 2.3.1 Ventajas.

- Se implementa en cualquier lugar y para cualquier tamaño de equipo.

- Reduce los ciclos de desarrollo al unificar el trabajo de todos los miembros del equipo, directores de proyecto, ingenieros de prueba, desarrolladores y analistas.
- Administra los cambios en el desarrollo de software.
- Brinda herramientas basadas en componentes de Rational para administración unificada de cambios.

### 2.3.2 Características.

- Soporta Oracle, Microsoft SQL Server, SQL Anywhere, y bases de datos Microsoft Access.
- Integrado con productos de administración de la configuración incluyendo el Rational Clear Case y el Microsoft Visual SourceSafe.
- Integrado con herramientas de la Suite de Rational para pruebas, desarrollo y análisis como Rational Suite AnalystStudio para analistas, Rational Suite TestStudio para ingenieros de pruebas y Rational Suite DevelopmentStudio para arquitectos y desarrolladores.

### 2.3.3 Producto de Rational Clear Quest.

- **IBM Rational Clear Quest MultiSite<sup>23</sup>**: Una opción al Rational Clear Quest para soportar rastreo de defectos y cambios en proyectos geográficamente distribuidos.

## 2.4 Herramienta Visual Source Safe 2005.

Microsoft Visual SourceSafe es un sistema de control de versiones en el nivel de archivos, que permite a muchos tipos de organizaciones trabajar en distintas versiones de un proyecto al mismo tiempo. Esta funcionalidad es especialmente ventajosa en un entorno de desarrollo de software, donde se usa para mantener versiones de código paralelas. Sin embargo, el producto también se puede utilizar para mantener archivos en cualquier otro tipo de equipo.

### 2.4.1 Ventajas.

- Admite la bifurcación, el uso compartido, la combinación y la administración de versiones de archivos de forma rápida y eficaz.

- Modelo de desarrollo check in (Protección de un elemento en el control de versiones bajo el control del sistema)/check out (Operación que desprotege un elemento en un control de versiones para permitir su modificación).
- Admite el desarrollo paralelo y las técnicas de desarrollo multiplataforma.
- Los miembros de un equipo podrán ver la última versión de un archivo, realizar cambios en sus copias locales y guardar nuevas versiones en la base de datos.

#### **2.4.2 Características.**

- Rendimiento y estabilidad mejorados
- Capacidad para equipos distribuidos y proyectos mayores
- Capacidad de programación y personalización
- Mejoras en la combinación de archivos
- Apariencia actualizada de los programas cliente
- Cambios en los complementos de SourceSafe para Visual Studio

#### **2.5 Herramienta Plastic SCM.**

Es un sistema de control de versiones de alto nivel, mediante el cual es posible gestionar todas las tareas que se realizan en el proyecto y su relación con los documentos de diseño, con el código y con los ejecutables. La funcionalidad de este producto hace que pueda ser empleado únicamente como control de versiones o para la gestión completa del ciclo de vida del desarrollo. En este caso integra la gestión de tareas, entregas, pruebas, gestión de los resultados, así como la trazabilidad entre todos ellos, según las normas de las prácticas recomendadas por diversos métodos de evaluación de calidad en los desarrollos de software, como CMMI.

##### **2.5.1 Ventajas.**

- Permite introducir todo un proceso controlado de avance, dando soporte a la creación y mantenimiento de nuevas versiones del producto, gestionando rigurosamente actividades complejas que pasan de ser descontroladas a sistemáticas y bien documentadas.
- Permite el desarrollo en paralelo mediante la implantación de un ágil manejo de ramas.

- Incorpora una visualización 3D de la historia de los ficheros y directorios, algo que hasta ahora no se había visto en ningún otro producto similar.
- Proporciona una herramienta gráfica (GUI) fácil de usar, en la que todas las acciones pueden ejecutarse visualmente. Diferentes vistas permiten al usuario acceder a la información contenida por el sistema, adecuándola a diferentes escenarios y perfiles.

### 2.5.2 Características.

- Versionado completo de la estructura del proyecto.
- Excelente soporte de desarrollo paralelo.
- Soporte multiplataforma.
- Completo sistema de seguridad basado en Listas de Copias Autorizadas (ACLs).
- Nuevas fórmulas de visualización.

### 2.5.4 Integración.

- Con clientes gráficos Visual Studio y Microsoft Access, los cuales están soportados en Windows.
- Eclipse para entornos Java/J2EE.
- Visual Studio (6 /.NET y 2005) para entornos Windows.

### 2.6 Herramienta Visual Studio Team System.

Visual Studio 2005 Team System es un conjunto de entornos, cada uno de ellos desarrollados para tareas específicas, proporcionando un entorno para el desarrollo de aplicaciones Web, el desarrollo de Servicios Web, otro para Arquitectos y otro para Testers<sup>17</sup>.

Visual Studio Team System, nombre en clave Burton, aporta un conjunto de herramientas dirigidas a desarrolladores, arquitectos, jefes de proyectos, testers<sup>17</sup>, en fin al equipo de desarrollo de una aplicación, Visual Studio Team System no ha sido desarrollado para aplicaciones de 64 bits, aunque si puede ser utilizado sobre estas y es complementario con Microsoft Project Server.

### 2.6.1 Ventajas.

- No solo aporta herramientas que soluciona de forma vertical algunas deficiencias sino que reduce la complejidad que supone la distribución de arquitecturas basadas en servicios.
- Facilita la colaboración y comunicación de todos los miembros del equipo, incluso si están geográficamente separados, desde el director ejecutivo hasta los programadores (pueden llegar a trabajar de manera simultánea con 500 personas).
- Incrementa la productividad y consistencia de las aplicaciones o proyectos integrándose junto a la interfaz de usuario, con los datos y con los procesos.
- Proporciona una mayor adaptabilidad a los negocios y la colaboración con los responsables de los departamentos de informática, a través de un desarrollo de software que responde dentro del paradigma familiar de las herramientas de Microsoft.

### 2.6.2 Características.

- Según sus requisitos personales, es extensible y no solo en cuanto a sus herramientas internas sino que además se extiende con cerca de 450 productos de 190 patrones distintos.
- Incluye un servidor para control de versiones, seguimiento de ficheros de proyecto, construcción automatizada, y productos para generar pruebas de carga adicionales.
- Soporta a SQL Server como gestor de base de datos.
- Se puede acceder a los archivos fuentes usando FTP.

### 2.6.3 Productos de Microsoft Visual Studio 2005 Team Suite.

- Microsoft Visual Studio 2005 Team Edition para Arquitectos.
- Microsoft Visual Studio 2005 Team Edition para Desarrolladores.
- Microsoft Visual Studio 2005 Team Edition para Testers<sup>17</sup>.
- Un servidor, Microsoft Visual Studio 2005 Team Foundation Server.
- Microsoft Visual Studio 2005 Team Test Load Agent (herramienta de generación de pruebas).

### 2.6.4 Microsoft Visual Studio 2005 Team Foundation Server.

Microsoft Visual Studio 2005 Team Foundation Server es la plataforma para Team System. Es un sistema de administración de cambios. En líneas generales posee la capacidad de reunir el trabajo colaborativo de

los VSTS 2005, centraliza el repositorio para cada equipo, está diseñado para soportar los diferentes roles en el proceso de desarrollo de software. Se basa en una arquitectura orientada a servicio, y está abierto para ser extendido e integrado a otras herramientas.

#### **2.6.4.1 Ventajas.**

- Proporciona control de código fuente<sup>5</sup> integrado.
- Provee seguimiento de problemas como son la integración completa entre los productos cliente y el servidor para facilitar la colaboración.
- Proporciona control de versiones.
- Provee gestión de ficheros de proyecto.

El control de versiones de Team Foundation (TFVC) se presenta como parte de Team Foundation Server y ofrece un control de código fuente<sup>5</sup> integrado para Visual Studio 2005. TFVC no comparte partes heredadas con Visual SourceSafe, se escribió únicamente para resolver las limitaciones del uso de VSS en grandes proyectos de desarrollo. En lugar de basarse en un sistema de archivos como el repositorio, aprovecha Microsoft SQL Server 2005 como mecanismo de almacenamiento sólido, escalable y de alto rendimiento.

#### **2.6.4.2 Ventajas de TFVC.**

- Escalabilidad.
- Fiabilidad.
- La posibilidad del desarrollo de forma remota, Mediante SOAP/http(s).
- Desarrollo en paralelo. Múltiples liberaciones, ramas, múltiples check out de un mismo archivo.
- Experiencia de Integración con check in. La combinación de cambios de fuente, comentarios, elementos de trabajo, políticas y notificaciones, se contribuyen integrando y ajustándose al proceso de la organización.

Los elementos de Visual Studio Team Foundation Server pueden ser utilizados sobre una red de área extensa ya que utiliza una comunicación basada en Servicios Web con el servidor. Además, diversos

patrones de Microsoft proporcionan integración a otros IDEs más allá de Visual Studio. Una nueva característica SCM disponible con (Team Foundation Server) es la denominada “shelving”, la cual permite al desarrollador verificar código incompleto y no incorporar ese cambio en la versión.

Es fundamentalmente una rama privada que almacena el trabajo en progreso que puede ser completado más adelante y fusionado en la rama de la versión. Visual Studio Team Foundation Server (VSTS) habilita el marco de trabajo para el resto de VSTS. Entre los componentes se incluyen.

- El seguimiento de los elementos de trabajo.
- Un portal de proyectos.
- El control de código fuente<sup>5</sup>.
- La automatización de versiones.
- Recomendaciones sobre procesos MSF™.
- Servicios de integración.

El control de versiones de Team Foundation ofrece toda la funcionalidad básica de los demás mecanismos de control de código fuente<sup>5</sup> y la mayoría de las mejoras de Visual Source Safe 2005. Asimismo, proporciona nuevas características avanzadas como:

- Aplazamiento de cambios.
- Directivas de protección.
- Integración con el nuevo sistema de seguimiento de elementos de trabajo.

### **2.7 Herramienta Borland StarTeam versión 6.0.**

StarTeam es una herramienta profesional para el control de versiones y desarrollo en equipo. Es lo suficientemente simple como para ser utilizada en proyectos pequeños y lo suficientemente robusta como para ser utilizada en proyectos grandes. Es un sistema completo de control de defectos que le permite identificar problemas y realizar un seguimiento desde el momento en que un problema es detectado hasta el momento en que es solucionado.

### 2.7.1 StarTeam Server.

Provee un repositorio con un proceso unificado para todos los puntos del desarrollo de aplicaciones, debe ser instalado en un servidor central al que todos los miembros del equipo tengan acceso. Para poder acceder, cada uno de los miembros del equipo debe instalar en su ordenador StarTeam Client. Utiliza configuraciones, cada una de las cuales puede contener uno o más proyectos que, a su vez, contienen ítems. Star Team Server provee dos herramientas para configurar y administrar el servidor.

- StarTeam Server: Es el punto de partida.
- Server Administration: Es una herramienta para la administración de servidores y provee una interface visual mediante la cual es posible, entre otras cosas, administrar los usuarios de un servidor.

### 2.7.2 StarTeam Client.

StarTeam Client es la herramienta que permite a los miembros del equipo acceder a las diferentes configuraciones y a los proyectos que contienen. Se trata de una herramienta visual intuitiva y fácil de utilizar que permite trabajar con varias configuraciones y proyectos al mismo tiempo. Es un sistema de alto rendimiento para la gestión de la configuración y del cambio para equipos de todos los tamaños. Proporciona la funcionalidad, seguridad y escalabilidad necesaria para cualquier equipo de desarrollo independientemente de su tamaño y distribución.

### 2.7.3 Ventajas.

- Acceso flexible, aunque seguro: los miembros del equipo pueden acceder desde y como quieran, ya sea desde Web, IDE, escritorio o línea de comandos.
- Alta seguridad y protección de los proyectos: StarTeam asegura que todos los elementos se almacenan de forma segura, controlando el acceso y las modificaciones en función de reglas establecidas.
- Control centralizado de los proyectos y tareas. Al proporcionar un repositorio común, StarTeam mantiene actualizado los elementos del proyecto y mejora la productividad y comunicación del equipo.
- Permite optimizar la gestión de las actividades de desarrollo de software con equipos dispersos geográficamente.

#### 2.7.4 Características principales.

- StarTeam Server está disponible para Windows, UNIX, Linux y Mac OS X.
- Los miembros del equipo pueden acceder a StarTeam Server utilizando:
  - StarTeam Client en Windows.
  - Star Disk Client, integrado en el Explorador de Windows.
  - StarTeam Web Edition, permite acceder desde cualquier navegador.
- Notificación automática de cambios utilizando correo electrónico.
- Está diseñado para mantener todas y cada una de las versiones permitiendo un acceso inmediato y transparente tanto de la última versión como de cualquiera de las versiones anteriores.
- Cada configuración es almacenada en una base de datos, que en la versión actual puede estar contenida en MSAccess, MSSQL Server, Oracle o IBM DB2.

#### Integración.

- Merant PVCS Version Manager.
- Microsoft Visual Source Safe.
- Borland CaliberRM para lograr un mejor funcionamiento en SCM.
- Microsoft Project.

#### 2.7.5 Productos de StarTeam.

A continuación se muestra los siguientes productos de StarTeam para un mejor funcionamiento.

- **StarTeam Enterprise Advantage:** Diseñado para cubrir las necesidades de gestión de cambios y configuración para equipos numerosos y muy dispersos que trabajen en proyectos de nivel de empresa. Integra gestión de requisitos y de cambios, seguimiento de defectos, gestión de versiones de archivos, gestión de proyectos y tareas.
- **StarTeam Enterprise:** Para equipos de desarrollos medianos que necesitan mejorar su eficacia y colaboración. De muy fácil utilización e integrando archivos, petición de cambio, tareas, y componentes de discusión en una simple interface.

- **StarTeam Standard:** Herramienta de iniciación de SCM de fácil uso, con componentes integrados para la gestión de versiones de archivos y seguimiento de defectos. Ayuda a los grupos de trabajo y equipos distribuidos poco numerosos a colaborar y gestionar con eficacia los cambios dentro del ciclo de vida del desarrollo de las aplicaciones.
- **StarTeam Datamart:** Es un sistema de apoyo para la toma de decisiones, ofrece funciones exhaustivas de consulta y análisis para SCM y entornos de gestión de cambios.

### 2.8 Herramienta PVCS Dimensions 7.0.

PVCS Dimensions 7.0 es una solución para el control completo de grandes procesos a través de Work flows personalizados, hasta control de versiones, gerenciamiento de líneas base y de construcción. Este paquete, ayuda ahorrar tiempo y dinero en la producción de sistemas.

La palabra clave de PVCS es disciplina. Con el paquete de Merant los equipos pueden trabajar organizadamente, ya sea en paralelo o concurrente, sin importar si los programadores están en un mismo lugar o distribuidos por el mundo. Esta herramienta ofrece:

- Recursos para administración del código fuente<sup>5</sup>.
- Software personalizado para cada cliente, a partir de un núcleo común.

Es un producto para Administración de Configuración de Software (SCM) apto para administrar sistemas distribuidos en cualquier plataforma, con una única infraestructura para interoperar en Windows, Mainframe, Unix y Web. Es una solución Merant de ayuda para la administración de cambios de aplicaciones en plataformas heterogéneas, que protege los datos de la compañía y disminuye el tiempo de desarrollo.

#### 2.8.1 Ventajas.

- Simplifica el proceso de cambio de administración para crear trabajos en medianas y grandes empresas.
- Controla la línea base.
- Permite organizar, administrar y proteger el software, creando una infraestructura eficiente mediante el control de cambios en las revisiones del software.

- Proporciona flexibilidad para adaptarse a pequeños proyectos así como funciones de expansión, controlando las versiones de los productos en dichos proyectos.

### 2.8.2 Características.

- Los desarrolladores tienen la libertad de elección entre una gran variedad de plataformas, diversos ambientes de desarrollo y la posibilidad de crear aplicaciones innovadoras y eficientes.
- Cada programa puede ser utilizado varias veces, sin poner en riesgo la calidad de los sistemas.
- La línea PVCS está conformada por Version Manager, Configuration Builder y Tracker.
  - **Version Manager:** organiza, gestiona y protege el inventario de software y crea una infraestructura eficiente, mediante el control de cambios en las innumerables revisiones a que es sometido el software.
  - **Configuration Builder:** automatiza el proceso de reconstrucción del sistema y asegura que sea precisamente reconstruido en cualquier momento.
  - **Tracker:** Permite la captura, gestión y comunicación de los problemas críticos de orden técnico o del negocio, en los más variados ambientes operaciones del mercado.

### 2.8.3 Integración.

- Websphere<sup>9</sup> Studio.
- Visual Studio. NET.

## 2.9. Herramienta Subversion (SVN).

Subversion es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es software libre bajo una licencia de tipo Apache/BSD. Se lo conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos y es usado por la mayoría de los proyectos de software open source<sup>18</sup> más grandes como KDE, GCC, Apache.

### 2.9.1 Ventajas.

- Versionado de directorios: Subversion implementa un sistema virtual de ficheros versionados que sigue los cambios por todo el árbol de directorios sobre el tiempo, permitiendo que todos los ficheros y directorios sean versionados.
- Historial real de versiones: Con Subversion, usted puede añadir, borrar, copiar, y renombrar ficheros y directorios. Y cada fichero nuevo añadido comienza con un historial nuevo, limpio y enteramente suyo.
- Envíos atómicos: Esto permite a los desarrolladores construir y enviar los cambios como trozos lógicos, y previene problemas que pueden ocurrir cuando solo una porción de un conjunto de cambios son enviados al repositorio exitosamente.
- Metadatos versionados: Subversion tiene una noción abstracta del acceso al repositorio, siendo fácil para las personas el implementar nuevos mecanismos de red. Subversion puede conectarse al Servidor de HTTP Apache como un módulo de extensión.
- Ramificación y etiquetado eficientes: Subversion crea ramas y etiquetas simplemente copiando el proyecto, usando un mecanismo similar al enlace duro. De este modo estas operaciones toman solamente una cantidad de tiempo pequeña y constante.
- Hackability<sup>24</sup>: Subversion no tiene un equipaje histórico; está implementado como una colección de bibliotecas compartidas en C con APIs<sup>22</sup> bien definidas. Esto hace a Subversion extremadamente fácil de mantener y reutilizable por otras aplicaciones y lenguajes.

### 2.9.2 Características principales.

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Puede ser servido, mediante Apache, sobre WebDAV/DeltaV.
- Maneja eficientemente archivos.
  - Posee clientes gráficos como TortoiseSVN para (Windows), RapidSVN para (Linux y Windows) y JSVN creado en Java y por lo tanto multiplataforma.
  - Un repositorio central.
  - Copias de trabajo en los clientes.
  - Arreglos de árboles de revisiones del repositorio.
  - Utiliza el método “copiar-modificar-mezclar”.

### **2.10. Producto Gforge.**

Gforge es un producto que integra un conjunto de herramientas muy aptas para el desarrollo software y la gestión de proyectos de software libre. Entre dichas herramientas se encuentran la de control de versiones como Subversión, CVS, Rational Clear Case, entre otras. Gforge es, por tanto, un producto que proporciona una infraestructura básica sobre la que apoyar la construcción de comunidades de software libre. Es un producto que parte de la última versión liberada (con licencia libre) del producto Sourceforge.

Es una plataforma para el desarrollo de aplicaciones orientada a grupos de trabajo; es el mayor sitio Web de desarrollo de software. Esta plataforma proporciona:

- Un entorno configurable con control de versiones.
- Herramientas para comunicación entre desarrolladores y servidor Web por proyecto.
- Desarrollo de una base de conocimiento compartida del proyecto.

#### **2.10.1 Ventajas Gforge.**

- Permite centralizar y homogeneizar la gestión de proyectos.
- Es una ventanilla única.
- Se consigue aumento de productividad.
- Se tienen herramientas comunes a toda la empresa o departamento.
- Centralización de los recursos técnicos en un servidor (en vez de tener que soportar múltiples máquinas por proyecto).

#### **2.10.2 Características de Gforge.**

- Un servidor Web por proyecto (Sitio Web).
- Control de versiones.
- Servidor Web para la coordinación del equipo de desarrollo.
- Peticiones de mejora, distribución de parches.
- Compartición de la documentación.
- Descargas de ficheros.

- Gestión de la planificación de tareas..

### **2.11 Herramienta Suite AllFusion™ Change Management.**

Computer Associates ofrece soluciones, líderes del mercado y reconocidas por la industria, para la Gestión de Cambios y Configuración del Software que soporta toda la empresa. Estas soluciones funcionan en una gran variedad de entornos operativos desde el mainframe al PC y UNIX.

Soportan variados entornos de desarrollo homogéneos, cliente/servidor o distribuidos a través de plataformas y automatizan las tareas rutinarias del desarrollo de aplicaciones, como las notificaciones, las aprobaciones y las migraciones de cambio de una fase a otra, aumentando así la calidad del software y reduciendo sus defectos. Al poder realizar una sincronización y seguimiento eficiente durante todo el ciclo de vida de la aplicación, los equipos de desarrollo pueden garantizar que los proyectos se entreguen siempre a tiempo, dentro del presupuesto y de acuerdo con las especificaciones.

Todas las soluciones AllFusion se integran transparentemente entre sí con el fin de tratar todas las fases del ciclo de vida de la aplicación de negocio:

- Modelado de Datos, Componentes y Procesos de Negocio.
- Gestión de Cambios y Configuración.
- Gestión de Proyectos y Procesos.

#### **2.11.1 Herramienta AllFusion Harvest Change Manager.**

Solución completa, integrada y basada en un repositorio para Gestión de Cambios y Configuración del Software que puede ayudar a las organizaciones a gestionar eficazmente complejas actividades de desarrollo empresarial en todo el ciclo de vida de desarrollo de aplicaciones. Completamente escalable, es perfecta para equipos que estén trabajando tanto en grandes como pequeños sistemas empresariales distribuidos. Posee una interfaz Web completa e intuitiva y ofrece escalabilidad y seguridad en toda la empresa. Además de estar diseñada para entornos distribuidos y heterogéneos.

Proporciona control y gestión de cambios del desarrollo de la aplicación a lo largo del ciclo de vida, desde el desarrollo, comprobación y distribución hasta la producción. Aquí se almacenan únicamente copias aprobadas del software en uso, así como las fuentes de estas copias.

#### **2.11.1.1 Ventajas.**

- CCC/Harvest 5.0 ofrece una interfaz gráfica simplificada para automatizar los procesos de gestión de cambios en el software en entornos distribuidos multiplataforma.
- Simplifica el desarrollo de aplicaciones Websphere<sup>9</sup> porque permite implementar cambios sabiendo que han sido probados a conciencia y aprobados antes de que se pasen al entorno de producción.
- Aporta un control a los desarrolladores con Websphere5.0 Studio.
- Permite automatizar los procesos repetitivos y controlar los diversos estadios del proceso de desarrollo del software.

#### **2.11.1.2 Características.**

- Se integra en la plataforma de código abierto Eclipse 2.0. Esta interoperabilidad hace de AllFusion Harvest CM una solución de gestión de cambios ideal para los entornos multiplataforma de las empresas de hoy.
- Soporta base de datos como Oracle, DB2, Microsoft SQL Server, Informix, ODBC 3.0.

#### **2.11.2 AllFusion Endeavor Change Manager.**

AllFusion Endeavor se encarga de automatizar la gestión de todo el proceso de desarrollo con el objeto de ofrecer una mayor eficiencia y un control total, desde la fase de diseño inicial a la fase de distribución.

##### **2.11.2.1 Ventajas.**

- Permite una gestión eficaz de las tareas asociadas con el entorno de desarrollo mainframe a través de funcionalidades automatizadas de transformación, gestión de las relaciones de módulos, gestión paralela del desarrollo y capacidades de automatización de versiones.
- Elimina los pasos manuales que atrasan el proceso de desarrollo de software y mejora la eficiencia al reducir los errores.

- Al actuar en conjunto con AllFusion™ Change Manager Enterprise Workbench, proporcionan un punto de control único e integrado para dinamizar y coordinar los procesos de cambio de software durante todo el ciclo de vida de la aplicación.

#### **2.11.2.2 Características.**

- Permite a los usuarios de Endeavor acceder a las funciones de gestión de cambios en el mainframe desde cualquier lugar vía un navegador Web.
- Dispone de una función de notificación por correo electrónico.

#### **2.11.3 AllFusion Change Manager Enterprise Workbench.**

Diseñado para racionalizar y automatizar complejos procesos de desarrollo de software a través de múltiples plataformas, esta solución ofrece niveles sin precedentes de soporte a la gestión de cambios y configuración a través de la empresa. Es uno de los pocos productos que pueden dinamizar y automatizar procesos complejos de desarrollo de software a través de múltiples plataformas.

##### **2.11.3.1 Ventajas.**

- Gestiona y define fácilmente los procesos de cambio.
- Simplifica el seguimiento de problemas y defectos.
- Gestiona las configuraciones en distintas plataformas.
- Ayuda a mejorar la capacidad de respuesta a los cambios y evita los errores que suelen darse como resultado de
- procesos de cambios deficientes.

##### **2.11.3.2 Características.**

- Al incorporar la tan necesaria disciplina al proceso completo de desarrollo, Enterprise Change Management Solution mejora significativamente la calidad de las aplicaciones o webs resultantes.
- Lleva a cabo un control efectivo sobre los desarrollos y versiones.

| Herramientas /Atributos               | Clear Case V7.0                                     | Clear Quest   | MS SourceSafe 2005 | Visual Studio Team System 2005 | PVCS Dimensions V 7.0 | Subversion V1.3.1                      | AllFusion               | StarTeam V6.0                | Plastic SCM                  |
|---------------------------------------|---|---|--------------------|--------------------------------|-----------------------|--|-------------------------|------------------------------|------------------------------|
| <b>1. Sistemas Operativos</b>         |   |   |                    |                                |                       |  |                         |                              |                              |
| MS Windows                            | ✓   | ✓   | ✓                  | ✓                              | ✓                     | ✓                                      | ✓                       | ✓                            | ✓                            |
| POSIX / UNIX                          | ✓   | ✓   | ✓                  |                                | ✓                     | ✓                                      | ✓                       | ✓                            | ✓                            |
| APPLE / Macintosh                     |   | ✓   | ✓                  |                                | ✓                     |  |                         |                              |                              |
| IBM                                   | ✓   | ✓   |                    |                                | ✓                     |  |                         |                              |                              |
| <b>2. Herramientas de Integración</b> | IBM, Eclipse, Microsoft, Sun                        | IBM, Eclipse, Microsoft                             | IBM, Microsoft     | Microsoft Eclipse              | IBM, Borland          | Eclipse (subclipse)                    | Eclipse, Microsoft, Sun | Microsoft, IBM, Borland, Sun | Eclipse, Sun, Visual Studio, |
| <b>3. Gestor de DB</b>                | SQL, Oracle, IBM DB2                                | SQL, Oracle, IBM DB2                                | SQL, Visual FoxPro | SQL, Oracle IBM DB2            | Oracle, SQL, IBM DB2  | SQLite, MYSQL, PostgreSQL, Berkeley DB | Oracle, IBM DB2, SQL    | SQL, Oracle, IBM DB2         | Postgresql, SQL Server       |
| <b>4. Fabricante</b>                  | IBM   | IBM   | Microsoft          | Microsoft                      | Merant                | CollabNet                              | Computer Associates     | Star Base Corporation        | Codice Software              |
| <b>5. Precio (USD)</b>                | 4,335.00  | 4,763.40  | 600.00             | 2,300.00                       | 2,000.00              | Open Source                            | 1,100.00                | 2665.00                      | 719.54                       |
| <b>6. Navegador</b>                   |   |   |                    |                                |                       |  |                         |                              |                              |
| MS Internet Explorer                  | ✓   | ✓   |                    | ✓                              | ✓                     | ✓                                      | ✓                       | ✓                            | ✓                            |
| Mozilla                               | ✓   | ✓   |                    |                                | ✓                     |  |                         | ✓                            |                              |
| Netscape                              | ✓   | ✓   |                    |                                | ✓                     |  |                         | ✓                            |                              |
| Apple Safari                          |   | ✓   |                    |                                | ✓                     |  |                         | ✓                            |                              |
| <b>7. Modelo de Repositorio</b>       | Cliente-Server                                      | Cliente-Server                                      | Cliente-Server     | Cliente-Server                 | Cliente-Server        | Cliente-Server                         | Cliente-Server          | Cliente-Server               | Cliente-Server               |
| <b>8. Licencia</b>                    | Propietaria   | Propietaria   | Propietaria        | Propietaria                    | Propietaria           | Apache/BSD                             | Propietaria             | Propietaria                  | Propietaria                  |
| <b>9. Lenguaje de Programación</b>    | ASP, C, C++, Java, Delphi, Java Script, PHP, Pascal | ASP, C, C++, Java, Delphi, Java Script, PHP, Pascal | C                  | C++                            | Visual Basic, C++     | C                                      | C, java                 | Delphi                       | C, C++                       |
| <b>10. Interfaces Web</b>             | ✓   | ✓   | -                  | ✓                              | ✓                     | ✓                                      | ✓                       | ✓                            | ✓                            |

**2.12 Otras herramientas existentes para llevar a cabo el control de la Gestión de Configuración.**

Durante el transcurso del capítulo 2 se hizo referencia a una muestra de herramientas que permiten llevar un control sobre la gestión de configuración. Se conocen que existen otras las cuales no fueron detalladas porque se tomó como punto principal para este estudio el grado de conocimiento con que cuentan las instituciones de software en nuestro país. A continuación se muestran algunas de estas herramientas.

Tabla (1) .Herramientas para la Gestión de Configuración del Software.

| <b>Herramientas para la Gestión de Configuración</b> |                                 |
|--|---------------------------------|
| <b>AccuRev</b>                                       | <b>TeamCoherence</b>            |
| <b>Aldon</b>   | <b>Vault</b>                    |
| <b>Bazaar</b>  | <b>LibreSource Synchronizer</b> |
| <b>SVK</b>   | <b>Aegis</b>                    |
| <b>CVS</b>   | <b>Arch</b>                     |
| <b>CVSNT</b>   | <b>BitKeeper</b>                |
| <b>Darcs</b>   | <b>CMSynergy</b>                |
| <b>Git</b>   | <b>OpenCM</b>                   |
| <b>GNU arch</b>                                      | <b>PureCM</b>                   |
| <b>Mercurial</b>                                     | <b>Superversión</b>             |
| <b>Monotone</b>                                      | <b>Vesta</b>                    |
| <b>Perforce</b>                                      | <b>SourceHaven</b>              |

**2.13 Conclusiones Parciales del capítulo 2.**

- Se hace referencia al papel protagónico que juegan las herramientas de control de Administración de la Configuración y Cambio en el desarrollo de software, enfatizando en su funcionamiento, ventajas, desventajas, características particulares que las diferencian entre sí.
- Se establecen comparaciones entre las herramientas con el objeto de brindar información mucho más completa y factible a la hora de elegir una en particular.

## CAPÍTULO 3

### Casos de Estudio de procedimientos existentes

#### 3.1 Introducción.

En este capítulo se hace referencia a empresas que se dedican a la producción de software en Cuba como La Casa de Software ETECSA-SIGTA, DESOFT, SOFTEL, SCADA y a nivel internacional AMCIS, teniendo en cuenta los procedimientos definidos para controlar la Administración de la Configuración y Cambio en estas instituciones.

Primeramente se hace un análisis de aspectos importantes los cuales se debe tener en cuenta a la hora de desarrollar un procedimiento.

#### 3.2 Plan de Gestión de Configuración.

##### 3.2.1 Introducción.

El Plan de Gestión de Configuración es un documento que se debe producir al comienzo de cada proyecto y define las políticas, estándares y procedimientos que se van a utilizar tanto por el resto de los integrantes del grupo de desarrollo como por el Responsable del Producto, para así gestionar la configuración en el transcurso de dicho proyecto. Debe contener los siguientes apartados explicados a continuación teniendo en cuenta el estándar del Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers, IEEE), aunque por lo general los estándares se adaptan a las necesidades de la organización.

##### 3.2.1.1 Propósito del Plan.

Aquí se explica cuál es el propósito del Plan, y a quién va dirigido. Normalmente el propósito del Plan de Gestión de Configuración es establecer y documentar los requisitos, políticas, estándares y procedimientos para la gestión de la configuración de los elementos software (ECS).

### **3.2.1.2 Alcance.**

Se especifica cuál es el alcance del proceso de Gestión de Configuración que se define en el plan. Se detalla:

- A qué proyectos o programas se aplica.
- Cuáles serán los elementos de configuración que se van a gestionar, tanto código como documentación.
- Relación de las actividades de GCS con otras actividades de gestión de configuración del hardware o del sistema.
- Limitaciones (de tiempo, recursos, etc.).

Supuestos que podrían tener un impacto sobre las actividades de GCS (coste, programación, capacidad de ser ejecutadas,...)

### **3.2.1.3 Documentación Complementaria.**

A otros documentos, como por ejemplo los estándares de IEEE, otros planes del proyecto, directivas, procedimientos, etc.

### **3.2.1.4 Glosario.**

Incluye las definiciones de los términos necesarios para entender el Plan de Gestión de Configuración que ayudan a la comunicación entre los miembros del proyecto, es decir términos, definiciones y acrónimos propios de este documento.

Ejemplo: Línea base, Elemento de configuración, Comité de Control de Cambios, ECS, CCC.

Normalmente hay un glosario común a todos los proyectos, y aquí sólo se especifican términos y acrónimos específicos de un proyecto.

### **3.2.2 Gestión.**

En esta sección se identifican las tareas de coordinación y gestión que serán necesarias para llevar a cabo las actividades de GCS. Principales interacciones entre las personas y las actividades de GCS, indicando los flujos de información y los repositorios que se utilizarán. Esta es una visión estática, sin tener en cuenta la secuencia, duración o localización temporal de estas actividades. Se puede usar, por ejemplo, un Diagrama de Flujo de Datos.

### **3.2.2.1 Organización.**

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

### **3.2.2.2 Responsabilidades.**

Se identificarán las responsabilidades asignadas para las diferentes actividades de gestión de configuración del software a ciertas personas u organizaciones en el proyecto.

Así, por ejemplo, se definirá:

- El Comité de Control de Cambios: Composición del CCC, y funciones.

### **3.2.2.3 Políticas, directivas y procedimientos aplicables.**

Aquí se identifica cualquier restricción externa a la GCS que provenga de las políticas, directivas y procedimientos impuestos sobre ella, especificando su impacto y efectos.

Ejemplos de cosas que pueden estar cubiertas por políticas, directivas y procedimientos preexistentes son:

- Designación de las versiones.
- Métodos de identificación de los productos software.
- Procesamiento de informes de incidencias, solicitudes de cambio y órdenes de cambio.
- Estructura y forma de operación de los comités de control de cambios.
- Métodos de manejo de los sistemas de librerías software, incluyendo la preparación, almacenamiento y actualización de módulos.
- Auditoría<sup>13</sup> de las actividades de Gestión de Configuración del Software (GCS).
- Requisitos para los informes de incidencias, solicitudes de cambios y órdenes de cambios.
- Nivel de pruebas necesario antes de que el software entre a gestión de configuración.

A nivel de Proyectos:

- Separado o formando parte de otro plan de gestión.
- Conteniendo la planificación de todas las actividades de AC.
- Con un formato estándar para todos los proyectos.
- Elaborado por el Responsable de la Configuración.

Establece:

- Roles y responsabilidades.
- Identificación de los ítems.
- Líneas base.
- Estructura y organización del almacenamiento.
- Procesos de ingreso y control de cambios.
- Herramientas, métodos y técnicas.
- Cronograma.

### **3.2.3 Actividades.**

#### **3.2.3.1 Identificación de la Configuración.**

Se describe el esquema de configuración que refleje la estructura de los productos generados a lo largo del proyecto.

##### **3.2.3.1.1 Identificación de los ítems de configuración.**

Se registran los ítems de configuración que serán controlados, se describe además las líneas base que existirán en el proyecto y su identificación mediante etiquetas, incluyendo métodos y procedimientos para asignarlos.

##### **3.2.3.2 Control de la Configuración.**

Se describe como será manejado el proceso de control de configuración. Las modificaciones requieren un proceso de aprobación por lo que en esta sección se identifican los procedimientos que se utilizarán para procesar solicitudes de cambio a las líneas base, responsabilidades y aprobaciones.

##### **3.2.3.2.1 Solicitud de Cambios.**

Se indican los procedimientos que serán seguidos para realizar cambios en las líneas bases, desde la solicitud del cambio hasta su aprobación, describiendo los documentos que serán generados en las distintas instancias del procedimiento de cambios y adjuntando el formato que tendrán dichos documentos.

#### **3.2.3.2.2 Evaluación de Cambios.**

Se indican los procedimientos para hacer la evaluación de un cambio solicitado, una vez recibida una solicitud de cambio se debe considerar el impacto que este producirá en el proyecto.

#### **3.2.3.2.3 Aprobación y desaprobación de Cambios.**

Se indican las responsabilidades asignadas en el proceso de control de cambios, quien o quienes estudiarán y aprobarán las solicitudes de cambio, en general las responsabilidades están asociadas a los productos afectados.

#### **3.2.3.2.4 Implementación de los Cambios.**

Se describe como se implementará un cambio aprobado, incluyendo la información de la solicitud del cambio, los nombres y versiones de los ítems de configuración afectados, fecha de instalación e identificador de la nueva versión generada.

#### **3.2.3.3 Informes del estado de la Configuración.**

Se describen los reportes de configuración que serán realizados, el tipo, frecuencia, información que contendrán y control de acceso. Es la tercera de las cuatro tareas básicas de Gestión de Configuración. El objetivo es mantener a los usuarios, a los gestores y a los desarrolladores al tanto del estado de la configuración y su evolución.

Esto implica, por tanto, la realización de tres actividades básicas:

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

Para definirlos y elaborarlos se debe tener en cuenta:

- El tipo de información a reportar.
- La audiencia del informe (Director, Gerente, J' Proyecto).
- El grado de control requerido por el destinatario.
- La formalidad requerida para solicitar el informe.

#### **3.2.3.4 Auditorías.**

Se describen las auditorías que serán realizadas sobre los ítems de configuración para determinar que los mismos son consistentes. Es una verificación independiente de un trabajo o del resultado de un trabajo o grupo de trabajos para evaluar su conformidad respecto a especificaciones, estándares, acuerdos contractuales u otros criterios.

Requiere de personal experimentado, y con un conocimiento del proceso de desarrollo. Debe ser realizada por personal ajeno al equipo de desarrollo técnico para mantener la objetividad de la auditoría.

#### **3.2.3.5 Control de Interfaces.**

Se describe las actividades de control de Interfaces para controlar los cambios a los elementos de configuración del proyecto, que modifican las interfaces con elementos fuera del alcance del plan (HW, SW de soporte).

Interfaz de software: acuerdos compartidos entre el programa y las entidades.

Interfaz de hardware: acuerdos compartidos entre el programa y las características de cualquier Hardware del entorno.

#### **3.2.3.6 Control de Subcontratistas/Vendedores.**

Gestión del software adquirido tanto terminado como en producción, para asegurar que el software comprado o subcontratado cumple los requisitos técnicos.

#### **3.2.4 Planificación.**

Se describe el tiempo de duración de las actividades de Administración de Configuración del Software que serán realizadas.

#### **3.2.5 Herramientas, Técnicas y Metodologías.**

Se indican las herramientas especiales de software, técnicas y metodologías que apoyarán el control de la gestión de configuración por los responsables de esta área de proceso clave.

### **3.2.6 Mantenición del Plan.**

Identificar las actividades y responsabilidades necesarias para asegurar la planificación continua de la gestión de configuración durante el ciclo de vida del proyecto.

### **3.2.7 Captura y Retención de Registros.**

Se describen los tipos de registros que serán generados, mantenidos y almacenados en la base de datos del proyecto, así como el objetivo de los mismos, adjuntando el formato que tendrán dichos documentos.

Como resumen de lo anteriormente explicado referente a los aspectos que incluye el Plan de Gestión de Configuración, a continuación se muestra la figura (7) que recoge los aspectos fundamentales que se debe tener en cuenta a la hora de elaborar un Plan de Gestión de Configuración.

## **3.3 Modelo de Procesos para la Industria de Software (MoProSoft versión 1.3)**

El Modelo de Procesos para la Industria de Software (MoProsoft), tiene por objetivo proporcionar a la industria mexicana, y a las áreas internas dedicadas al desarrollo y mantenimiento de software, un conjunto integrado de las mejores prácticas basadas en los modelos y estándares reconocidos internacionalmente, tales como ISO 9000:2000y CMMI-SW niveles 2 y 3. MoProsoft está dirigido a la micro y pequeña industria. Define 9 procesos agrupados en 3 categorías que corresponden a la estructura organizacional de las empresas de software, con el fin de evaluar y/o certificar los procesos de las organizaciones.

Las categorías de procesos son: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización. La categoría de Alta Dirección contiene el proceso de Gestión de Negocio.

La categoría de Gerencia está integrada por:

- Gestión de Procesos.
- Gestión de Proyectos.
- Gestión de Recursos.
  - Recursos Humanos y Ambiente de Trabajo.
  - Bienes.

- Servicios e Infraestructura.
- Conocimiento de la Organización.

La categoría de Operación está integrada por:

- Administración de Proyectos Específicos y de Desarrollo.
- Mantenimiento de Software.

En cada proceso están definidos los roles y responsables por la ejecución de las prácticas. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y capacitación para desempeñarlos. En MoProSoft se clasifican los roles en Grupo Directivo, Responsable de Proceso y otros roles involucrados. Además se considera al Cliente y al Usuario como roles externo a la organización.

### **3.4 Procedimiento utilizado por la Casa de Software ETECSA-SIGTA**

En el Manual para la Gestión de la Configuración elaborado por la empresa Casa de Software ETECSA-SIGTA se utiliza parámetros como código, revisión, página y fecha con el objetivo de identificar este manual a la hora de institucionalizarlo. En este manual se tratan aspectos como objetivo, alcance, definiciones, referencias, responsabilidades, y otros como el control de versiones de programas y documentos, herramientas para el control de versiones y las bibliotecas de trabajo en las cuales se almacenan los elementos de configuración.

#### **3.4.1 Objetivo.**

Establecer la política de gestión de configuración en la Gerencia de Casa de Software SIGTA.

#### **3.4.2 Alcance.**

Se aplica a todas los procesos dentro de la Gerencia Casa de Software SIGTA

#### **3.4.3 Definiciones.**

Se hace referencia a las definiciones utilizadas respecto a Configuración, Elemento de Configuración, Gestión de Configuración, Control de la Configuración o Control de Cambios.

#### **3.4.4 Referencias.**

Se hace referencia a los documentos utilizados para la elaboración de este manual.

#### **3.4.5 Responsabilidades.**

Se especifica quién estará a cargo de las acciones de aprobar el documento, implantar el documento, ejecutar las acciones y supervisar el documento.

#### **3.4.6 Instrucciones.**

##### **3.4.6.1 Control de versiones de Programas o Software.**

En el caso del control de versiones de programas o software el código de revisión

El código de revisión se incrementará cada vez que se realice un cambio para la corrección de algún problema o falla detectada. Ejemplo: El software debe permitir que el usuario pueda editar el valor de un campo, sin embargo el campo está definido como de solo lectura.

El código de versión se incrementará cada vez que se adicione una nueva funcionalidad al software. Ejemplo: El software solamente exportaba a ASCII y se decide agregarle la funcionalidad de exportar a EXCEL.

##### **3.4.6.2 Control de versiones de Documentos.**

En el caso de los documentos (manuales de usuarios, ayuda en líneas, entregables, etc.)

El código de revisión se incrementará cuando los cambios realizados al documento sean menores al 50% de su contenido.

El código de versión se incrementará cuando los cambios realizados al documento sean mayores e igual al 50% de su contenido.

#### **3.4.6.3 Herramientas para el Control de Versiones.**

Se empleará la herramienta Visual Source Safe de Microsoft para el control de los diferentes elementos de configuración.

#### **3.4.6.4 Bibliotecas de Trabajo.**

Existirán, al menos, tres Bases de Datos o Bibliotecas de trabajo independientes. Una llamada Biblioteca Dinámica o de Programación, otra llamada Biblioteca Controlada o de Calidad, y otra llamada Biblioteca Estática o de Producción.

#### **3.4.6.5 Proceso de Administración de los Elementos de Configuración.**

Para administrar los elementos de Configuración la empresa Casa de Software SIGTA propone utilizar plantillas que lleven el control de los cambios de los elementos de configuración de una librería a otra.

### **3.5 Procedimiento en proceso de implantación de la empresa DESOFT**

En el procedimiento utilizado en la empresa Desoft se analizan parámetros como objetivo, alcance, definiciones y bibliotecas.

#### **3.5.1 Objetivo.**

Este Procedimiento tiene como objetivo establecer la metodología para administrar los elementos de configuración relacionados con los diferentes productos (software y documentación asociada) confeccionados durante el proceso de desarrollo de software. Describe además los roles que ejecutan cada una de las fases y referencia del modelo utilizado como patrón para la formación del estándar de DESOFT S.A.: Rational Unified Process (RUP).

#### **3.5.2 Alcance.**

Áreas de Ingeniería de Software, Desarrollo, Implementación y Soporte, Soporte a los Procesos Informáticos y Calidad.

#### **3.5.3 Definiciones y Términos.**

Se hace referencia a las definiciones utilizadas respecto a:

**IC:** Ingreso a Configuración.

**LCA:** Lista de Copias Autorizadas.

**RC:** Requerimiento de Cambio.

**NC:** Nota de Cambio.

**RCP:** Registro de Cambios del Proyecto.

**RCS:** Registro de Cambios por Soporte.

**CCC:** Comité de Control de Configuración.

#### **3.5.4 Bibliotecas de trabajo utilizadas.**

##### **Biblioteca Estática o Final.**

Mantiene las copias maestras y las autorizadas de los elementos de configuración liberados para uso operacional. La liberación de los elementos de configuración desde la Biblioteca Controlada o de Calidad hacia la Biblioteca Estática será autorizada por el Especialista de Calidad.

##### **Biblioteca Controlada o de Calidad.**

Es la que mantiene los elementos de configuración ya desarrollados y en fase de validación, verificación o pruebas por parte de los especialistas de Calidad.

##### **Biblioteca Dinámica.**

Será usada para mantener las líneas base de desarrollo. El acceso a la misma será libre para todo el equipo y será controlada por el Jefe de Proyecto. Se mantendrá en una PC del equipo de trabajo que funcione como servidor de control de configuración, y será designada por el propio Jefe de Proyecto.

#### **3.6.5 Gestión de Configuración.**

Tarea de definir y mantener las configuraciones y versiones de los artefactos. Esto incluye la definición de líneas base, control de versiones, control de estado y control de almacenamiento de los artefactos.

#### **3.5.6 Descripción de las Actividades.**

Se describe como se va a llevar a cabo el proceso de control de cambio durante el desarrollo del software, teniendo en cuenta la utilización de plantillas que permitirán llevar un control de los cambios realizados en el proyecto, en estas se encuentran:

### **3.5.7 Responsabilidades.**

Las personas asignadas para la ejecución de esta disciplina dentro del proceso de desarrollo de software deben cumplir con lo establecido en este procedimiento.

### **3.5.8 Registros.**

Los documentos como el registro de ingreso a configuración, lista de copias autorizadas, requerimientos de cambios asociados, nota de cambio, registro de cambios del proyecto y registro de cambios por soporte deberán adjuntarse al Expediente del Proyecto, los cuales se guardarán en copia digital en la Biblioteca Estática o Final durante un período de 5 años.

### **3.5.9 Referencias bibliográficas**

Se especifica las fuentes que sirvieron de referencia para la elaboración de este procedimiento.

## **3.6 Plan de Gestión de Configuración utilizado por la empresa SOFTEL referente a la Gestión de Configuración**

### **3.6.1 Introducción.**

Esta sección brinda una visión global del “Documento Plan de Gestión de la Configuración del Software” y contiene las siguientes subsecciones:

- Propósito.
- Alcance.
- Definiciones, Acrónimos y Abreviaciones.
- Referencias.
- Visión General.

#### **3.6.1.1 Propósito.**

Breve descripción del propósito del presente documento.

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán

bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

#### **3.6.1.2 Alcance.**

Una breve descripción del alcance del plan; que modelo está asociada a él y cualquier otra cosa que esté influenciada o afectada por este documento. El Plan de configuración está basado en algunos supuestos que se detallan:

Se deben incluir en control de configuración la mayor cantidad de productos posibles, tomando en cuenta siempre las restricciones dadas por la duración del proyecto y por la capacidad organizativa del grupo. La elección de los elementos de configuración se realizará en base a los documentos entregables definidos en el proceso de desarrollo.

#### **3.6.1.3 Definiciones, Acrónimos, y Abreviaturas.**

Describir todas las definiciones, acrónimos y abreviaturas que son empleadas en el documento.

#### **3.6.1.4 Referencias.**

Esta sección debe:

- Brindar una lista completa de todos los documentos referenciados en el Documento Plan de Administración de la Configuración del Software.
- Identificar cada documento por título, número del informe (si se aplica), fecha y organización que lo publica.
- Debe especificar las fuentes de las cuales las referencias fueron obtenidas.

Esta información puede estar contenida en un anexo o en un documento adicional.

#### **3.6.1.5 Vista General.**

Esta subsección describe lo que contiene este documento explicando como está organizado el mismo.

#### **3.6.1.6 Gestión de la configuración del Software.**

Se describen las responsabilidades y responsables para la realización de las actividades de administración de configuración dentro del proyecto según se describe en el proceso de administración de la configuración.

### **3.6.2 Organización, Responsabilidades e Interfaces.**

#### **3.6.2.1 Organización.**

Se deben especificar las estructuras organizacionales tanto técnicas como de gestión de proyecto, las cuales participarán en la implementación de las actividades de Administración de la Configuración del Software.

Se debe identificar:

- Todas las líneas de trabajo que participen o sean responsables de actividades de Administración de la configuración.
- El cometido de estas líneas de trabajo dentro del proyecto.
- Relaciones entre estas líneas de trabajo.

#### **3.6.2.2 Responsabilidades.**

El responsable de la administración del software debe proveer la infraestructura y el entorno de configuración para el proyecto. Debe seguir la línea base, controlando las versiones y cambios de ella. Debe definir y construir el Ambiente Controlado e informar al resto del equipo sobre la manera de usarlo. El responsable de la administración de la configuración es un apoyo importante para las decisiones que debe tomar el Comité de Control de Cambio, debiendo formar parte de éste si lo cree necesario.

#### **3.6.2.3 Interfaces.**

Las actividades de Control de Interfaces controlan los cambios a los elementos de configuración del proyecto, que modifican las interfaces con elementos fuera del alcance del Plan.

#### **3.6.2.4 Herramientas, Ambiente e Infraestructura.**

Se describe el ambiente de computadoras y herramientas de software que se emplearán para cumplir con las funciones de la Administración de la Configuración durante todo el ciclo de vida del proyecto.

Describir las herramientas y procedimientos necesarios para el control de versión de los elementos de configuración durante todo el ciclo de desarrollo.

### **3.6.3 Programa de Administración de la Configuración.**

#### **3.6.3.1 Identificación de la Configuración.**

### **Métodos de Identificación**

Para este proyecto los elementos de configuración se corresponderán con los entregables definidos en el Modelo de Proceso, aunque no necesariamente todos los entregables deben ser elementos de configuración.

La decisión de cuales de los entregables serán elementos de configuración será tomada por el responsable de la administración de la configuración, quién deberá tomar en cuenta qué productos serán necesarios cuando se quiera recuperar una versión completa del sistema.

#### **3.6.4 Nomenclatura de Elementos.**

En esta sección se especifican la identificación y descripción única de cada elemento de configuración y las diferentes versiones de cada elemento.

Nomenclatura para cada elemento:

- X es un número de 1 dígito que identifica al grupo.
- Y indica la versión del elemento de configuración o entregable.
- Extensión indica la extensión del elemento de configuración o entregable.

Ejemplo: RQALSG1v2.doc, es como se deberá llamar el entregable "Alcance del Sistema" correspondiente al grupo 1 y cuya versión del documento es la 2.

#### **3.6.5 Líneas Base del Proyecto.**

Las líneas base brindan un estándar oficial sobre el cual se continúa desarrollando el trabajo y solo podrá ser modificado si se autoriza el cambio mediante las autoridades competentes.

En este punto se describe donde serán definidas las líneas base del proyecto, lo más común es establecerlas al final de cada una de las fases.

Se debe generar una línea base por iteración en cada Fase, de acuerdo a lo siguiente:

- Los eventos que dan origen a la línea base.

- Los elementos que serán controlados en la línea base.
- Los procedimientos usados para establecer y cambiar la línea base.
- La autorización requerida para aprobar cambios a los documentos de la línea base.
- Elementos de la Línea Base del Proyecto.

### **3.6.6 Control de Configuración y Cambio.**

Se entiende por cambio al sistema, las modificaciones que afecten a la línea base del sistema, como pueden ser:

- Cambios en los Requerimientos.
- Cambios en el Diseño.
- Cambios en la Arquitectura.
- Cambios en las herramientas de desarrollo.
- Cambios en la documentación del proyecto (agregar nuevos documentos o modificar la estructura de los existentes).

Proceso de solicitud de cambio y aprobación.

- Para atender las solicitudes de cambio se seguirá el proceso descrito en el documento Procedimiento de Solicitud de Cambio.
- Cuando se realiza la solicitud de un cambio, se actualiza el documento de “Solicitud de cambio” para registrar esta solicitud.
- Se debe ingresar toda la información necesaria, detallada en el documento.

Comité de Control de Cambio.

- Describir los miembros y los procedimientos para procesar las solicitudes de cambio.

### **3.6.7 Estado de la Configuración.**

Las actividades de control de estado son para reunir información y reportar el estado de los elementos de configuración.

Se debe especificar lo siguiente:

- Qué elementos serán revisados de la línea base y por cambios a realizarse.
- Qué tipos de reportes de estado serán generados y con qué frecuencia.
- Como la información será obtenida, guardada, procesada, y reportada.

- Como será controlado el acceso a los datos de estado.

Si se utiliza una herramienta automática deberá ser especificada su funcionalidad y modo de uso explícitamente o por referencia.

En los reportes de estado de los elementos de configuración se debe incluir como mínimo la siguiente información:

- Su primera versión aprobada.
- El estado de los cambios solicitados.
- El estado de implementación de los cambios aprobados.

### **3.6.8 Auditorías y reportes de la configuración.**

Se realizarán auditorías de la línea base antes de una liberación de ésta o de una actualización de la versión de un componente prioritario de ésta.

### **3.6.9 Recursos y capacitación.**

Describir las herramientas de software, personal y entrenamiento requerido para implementar las actividades específicas de administración de la configuración.

## **3.7 Plan utilizado en el proyecto SCADA en la Universidad de las Ciencias Informáticas (UCI)**

Después de una investigación de los proyectos existentes dentro de la Universidad de las Ciencias Informáticas, se tomó como referencia el proyecto SCADA, de la facultad 5, dentro del cual se está desarrollando un procedimiento para controlar la Gestión de Configuración y organizar el trabajo de los desarrolladores a la hora de enfrentar sus tareas.

### **Plan de Gestión de Configuración Versión 1.0.**

#### **Tabla de Contenidos.**

#### **3.7.1 Introducción.**

Incluye un resumen del Plan.

#### **3.7.1.1 Propósito.**

Define el propósito del Plan.

#### **3.7.1.2 Alcance.**

Proyectos con los que se involucra el Plan.

#### **3.7.1.3 Definiciones, Acrónimos y Abreviaturas.**

Se especifica los conceptos utilizados referentes a la Gestión de Configuración del Software.

#### **3.7.1.4 Referencias.**

Lista de documentos a los que se hace referencia en el Plan de Aseguramiento de la Calidad.

#### **3.7.1.5 Resumen.**

Resumen de los aspectos del plan.

#### **3.7.1.6 Organización, Responsabilidades e interfaz.**

Este proyecto está organizado por líneas de trabajo en las cuales existe un responsable por cada línea:

- Línea Gráficos: Responsable X.
- Línea Drivers: Responsable Y.
- Línea Reportes: Responsable Z.
- Línea OPC<sup>14</sup>: Responsable W.

### **3.7.2 Programa de Gestión de Configuración.**

#### **3.7.2.1 Identificación de la Configuración.**

##### **3.7.2.1.1 Métodos de Identificación.**

Describe como serán nombrados, marcados y numerados los Elementos de Configuración del proyecto, estos pueden comprender código fuente<sup>5</sup>, ejecutables, documentación, modelos, hardware.

##### **3.7.2.1.2 Línea base del proyecto.**

La línea base provee un estándar oficial en el que se basan los siguientes trabajos y en el que solo se realizan los cambios autorizados. Describe en que punto del ciclo de vida son establecidas las líneas

bases. Las líneas bases más comunes son las que se generan al final de cada fase. Describe quien autoriza la línea base y que se hace con ella.

#### **3.7.2.1.3 Definición de Bibliotecas.**

Describe los tipos de bibliotecas que se usarán para el desarrollo de las aplicaciones, así como los niveles de acceso y la dirección física de las mismas.

#### **3.7.2.1.4 Control de versiones.**

Hay 4 ramas:

- Trunk: rama principal donde se encuentran los últimos cambios que se han efectuado en todo el proyecto.
- Branches: es la rama en la que cada miembro del proyecto realiza sus tareas correspondientes a su línea de trabajo.
- Tags: las actividades que tiene que realizar cada persona que se encuentra en el proyecto.
- Unestable: es la rama donde cada usuario puede publicar los cambios realizados al Branches de otro usuario para luego saber si va a ser admitida o no la modificación en el Trunk.

Describe los procedimientos a seguir para realizar el control de versiones al proyecto. Esto incluye la descripción de los procesos de tráfico de los ECS entre repositorios, las direcciones físicas de los mismos y las políticas que va a tener el proyecto para realizar dicha actividad.

#### **3.7.2.1.5 Control de configuración y de cambios.**

##### **Proceso de solicitud y aceptación de cambios.**

Describe el proceso mediante el cual los problemas y los cambios son propuestos, revisados y aprobados o no.

##### **Comité de Control de Cambios (CCB).**

Los que integrarán este comité serán el jefe de proyecto o línea.

##### **Responsable de la Gestión de la Configuración.**

El jefe de cada línea será responsable de la configuración.

### **3.7.3 Estado de la configuración.**

#### **3.7.3.1 Almacén del proyecto y procedimiento de liberación.**

Describe políticas de almacenamiento y planes de resguardo y recolección. En el proceso de liberación se describe que es lo que se libera, quién y como tratar los problemas además de instrucciones de instalación.

#### **3.7.3.2 Reportes y auditorías.**

Describe el contenido, el formato y el propósito de los reportes de solicitudes de cambio. Los reportes basados en los defectos o en las solicitudes de cambio pueden aportar indicadores de calidad útiles y además alertar a los gestores y a los desarrolladores sobre algunas áreas particularmente críticas del desarrollo. Los defectos deben ser calificados por lo crítico que pueden ser en el desarrollo del proyecto (alta, media y baja) y deben ser reportados sobre las bases siguientes:

- Antigüedad: (basado tiempo).
- Que tiempo se mantiene abierto un defecto.
- Momentos del ciclo de vida en el que son detectados defectos contra el momento en el que fueron resueltos.
- Combinación de tiempo y conteo.
- Número de defectos acumulados y resueltos en un tiempo.
- Razón de defectos descubiertos contra los resueltos.
- Brecha de calidad, defectos abiertos contra defectos cerrados.

#### **3.7.4 Entrenamiento y Recursos.**

Describe herramientas de software, personal y entrenamientos que son necesarios para implementarlas actividades específicas de Gestión de Configuración. Puede hacerse referencia al Plan de Capacitación.

#### **3.7.5 Proceso de Control Versiones y Gestión Cambios.**

El proceso de Control de Versiones dentro de la línea se realiza en su totalidad mediante la herramienta el "Subversion", herramienta la cual actúa como depósito de toda la información que se genera y se guarda. Para el proceso de Gestión de Cambios se utiliza como interfaz la herramienta denominada "trac", la cual

le da un ambiente más amigable para trabajar. Este implementa mediante los "ticket" un método para la organización, seguimiento y cumplimiento de las tareas asignadas a cada estudiante. Los que tienen potestad para gestionar los cambios son el Jefe de Línea y el Jefe de la Gestión de Configuración. Todos los artefactos que se generen deben estar bajo el control de versiones.

### **3.8 Conclusiones Parciales del capítulo 3**

- Durante el desarrollo de este capítulo se ha observado los diferentes planes y procedimientos utilizados por instituciones de software para llevar un control sobre la Gestión de Configuración.
- Toda la investigación y análisis de estos manuales se hizo con el objetivo de evidenciar los aspectos fundamentales que se debe tener en cuenta a la hora de elaborar un Plan de Gestión de Configuración, el cual sirve de base para la estandarización de un procedimiento determinado, en dependencia de las características de la entidad.

## CAPÍTULO 4

### Propuesta de Procedimiento.

#### 4.1 Introducción

En este capítulo se realiza la propuesta del procedimiento para lograr un mayor control en la Administración de la Configuración y Cambio dentro de los proyectos de desarrollo de software de la Universidad de las Ciencias Informáticas, tomando como premisa la existencia de un Plan de Gestión de Configuración, el cual es el primer paso a cumplir a la hora de medir la calidad de un producto en cuanto a Gestión de Configuración.

**4.2 Titulo:** Plan de Administración de Configuración y Cambio a nivel de Universidad centrado en las facultades.

#### 4.2.1 Historial de Revisiones.

Primeramente se propone utilizar un Historial de Revisiones para el Plan de Gestión de Configuración, con el siguiente formato, el cual contará con parámetros como Fecha, Versión, Descripción y Autor, con el objetivo de ir almacenando las actualizaciones y revisiones que pueda adquirir este Plan propuesto en dependencia a las necesidades de los proyectos existentes en la Universidad de las Ciencias Informáticas (UCI).

Tabla (10). Historial de Revisiones

| Fecha        | Versión | Descripción                       | Autor   |
|--------------|---------|-----------------------------------|---------|
| <DD/MM/AAAA> | <X.Y>   | <Texto descriptivo de la versión> | <Autor> |
|              |         |                                   |         |
|              |         |                                   |         |

A continuación se enuncian los parámetros a tener en cuenta a la hora de utilizar un Plan de Gestión de Configuración en un proyecto determinado:

#### **4.2.2 Introducción.**

Provee un resumen de las actividades de Administración de Configuración del Software para aquellos que interactúen con esta área de proceso clave puedan entender dicho Plan.

##### **4.2.2.1 Propósito del Plan.**

Este documento describe las actividades de gestión de configuración de software que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Aquí se definen tanto los productos que se pondrán bajo control de configuración como los procedimientos que deben ser seguidos por los integrantes del equipo de trabajo.

##### **4.2.2.2 Alcance.**

El Plan de Configuración está basado en algunos supuestos que se detallan a continuación:

- En dependencia del tiempo de duración del proyecto, se busca una respuesta a los cambios, tratando que este procedimiento sea lo menos burocrático posible.
- Resulta importante tener control sobre cada una de las iteraciones y fases, de los productos generados en estas y de los cambios surgidos, evaluados y aprobados.
- Se deben incluir en control de configuración la mayor cantidad de productos posibles, tomando en cuenta siempre las restricciones dadas por la duración del proyecto y por la capacidad organizativa del grupo.
- La elección de los elementos de configuración se realizará en base a los entregables, siendo ésta responsabilidad del Responsable de Configuración, apoyado por los integrantes del proyecto.

##### **4.2.2.3 Definiciones y acrónimos (Glosario).**

Se sugiere describir todas las abreviaturas que son empleadas en los documentos que se van creando en los proyectos como:

- ECS: Elemento de Configuración del Software
- CCC: Comité de Control de Configuración.

- CCB: Comité de Control de Cambio.
- RC: Responsable de Configuración.
- RHCP: Registro Histórico de Cambios del proyecto.
- GAC: Grupo de Aseguramiento de la Calidad.
- RP: Responsable del Producto.
- GP: Gerente de Proyecto.

#### **4.2.2.4 Documentación Complementaria (Referencias).**

Se sugiere:

Plan de Gestión de Configuración.

Esta información puede estar contenida en un anexo o en un documento adicional.

#### **4.2.3 Gestión.**

Se propone en esta sección identificar la vinculación y gestión de las tareas necesarias para desarrollar las actividades de GCS.

##### **4.2.3.1 Organización.**

Se propone especificar la estructura organizativa del proyecto, así como las técnicas necesarias para el desarrollo del mismo, las cuales permitirán la ejecución de las actividades de Administración de Configuración y Cambio.

Se debe identificar:

- Todas las líneas de trabajo que participen o sean responsables de las actividades de Administración de la Configuración y Cambio.
- La función de realizará cada integrante del proyecto.
- Relaciones que existirá entre cada tarea desempeñada por los desarrolladores.

##### **4.2.3.2 Responsabilidades.**

Se propone asignar responsabilidades para las diferentes actividades de GCS a ciertas personas u organizaciones.

Se definirá:

| No. | Tareas a llevar a cabo  | Responsable                                       |
|-----|---|---|
| 5.1 | <ul style="list-style-type: none"> <li>▪ Actualizar el estado de la Solicitud de Cambio.</li> <li>▪ Distribuir la información a los responsables de su implementación.</li> <li>▪ Controlar y entregar las copias de los ítems de configuración a modificar.</li> </ul>   | <b>Responsable de Configuración (RC).</b>         |
| 5.2 | <ul style="list-style-type: none"> <li>▪ Identificar los elementos de configuración.</li> <li>▪ Controlar las versiones y cambios</li> <li>▪ Mantener actualiza la base de datos del proyecto.</li> <li>▪ Mantener actualizado el Registro Histórico de Cambios.</li> </ul>   | <b>Comité de Control de Configuración (CCC).</b>  |
| 5.3 | <ul style="list-style-type: none"> <li>▪ Analizar y evalúa el impacto del Cambio en un determinado Elemento de Configuración</li> <li>▪ Adoptar la decisión final acerca de la aceptación o no de las solicitudes de cambios.</li> <li>▪ Controlar a través de una sección conocida como Registro Histórico de Cambios del Proyecto (RHCP) el estado de las peticiones de cambio pendientes.</li> </ul> | <b>Comité de Control de Cambio (CCB)</b>          |
|     | <ul style="list-style-type: none"> <li>▪ Participar en la evaluación de los cambios.</li> <li>▪ Informar acerca del estado de los cambios.</li> </ul>   | <b>Responsable del Producto (RP).</b>             |
| 5.4 | <ul style="list-style-type: none"> <li>▪ Definir y realizar las pruebas al producto.</li> <li>▪ Verificar los cambios realizados.</li> <li>▪ Detectar defectos en los ECS.</li> </ul>   | <b>Grupo de Aseguramiento de la Calidad (GAC)</b> |
| 5.5 | <ul style="list-style-type: none"> <li>▪ Controlar el acceso al repositorio.</li> <li>▪ Controlar el personal autorizado a realizar copias.</li> </ul>  | <b>Gerente de Proyecto (GP).</b>                  |

#### **4.2.3.3 Políticas, directivas y procedimientos aplicables.**

Se propone identificar en esta sección cualquier restricción que provenga de las políticas, directivas y procedimientos impuestos en el proyecto, especificando impacto y efectos.

A nivel de proyecto este Plan deberá establecer:

- Roles y responsabilidades.
- Identificación de los ítems.
- Líneas base.
- Estructura y organización del almacenamiento.
- Procesos de ingreso y control de cambios.
- Herramientas, métodos y técnicas.
- Separado o formando parte de otro plan de gestión.
- Conteniendo la planificación de todas las actividades de AC.
- Con un formato estándar para todos los proyectos.
- Elaborado por el Responsable de la Configuración.

#### **4.2.4 Actividades.**

##### **4.2.4.1 Identificación de la Configuración.**

###### **4.2.4.1.1 Elementos de Configuración a controlar.**

La Gestión de Configuración comienza desde el inicio del software hasta su mantenimiento, con el objetivo de lograr un producto de buena calidad, por lo que se propone que los elementos de configuración a controlar se definan teniendo en cuenta los distintos flujos de trabajo por los que el software transcurre durante su evolución. La decisión de cuales de los elementos de configuración serán tomados como entregables será tomada por el Responsable de Configuración.

Se propone que los elementos de configuración se definan de la siguiente manera:

##### **Ingeniería del Sistema:**

- Plan del Proyecto.
- Modelo de la situación actual.

- Informe preliminar de necesidades.
- Especificación de requisitos del Sistema Global (HW y SW).
- Especificación de la interfaz del Sistema.
- Especificación funcional del Sistema.
- Aceptación formal de los requisitos del Sistema Global por parte del cliente.
- Los estándares y procedimientos de ingeniería de software utilizados.

#### **Requerimientos:**

- Especificación formal de Requisitos de Software, que debe contener:
  - Descripción de Datos que consume y produce el Sistema.
  - Funciones que debe realizar el Software del Sistema.
  - Rendimiento requerido al Sistema.
  - Descripción del flujo de información.
  - Definición de interfaces.
  - Restricciones a aplicar.
- Plan de Pruebas.
- Aceptación formal por parte del cliente de los Requisitos de Software.

#### **Análisis y diseño:**

- Descripción de diseño del Software.
- Descripción de la arquitectura.
- Diseños de bases de datos.
- Contenidos de bases de datos.
- Descripción de interfaces.
- Descripción de algoritmos.
- Referencias cruzadas con los requisitos.
- Previsiones de pruebas.
- Manual del Usuario preliminar.
- El diseño preliminar.
- El diseño detallado.

### **Implementación:**

- Listado y archivo sobre soporte del código fuente<sup>5</sup> y librerías asociadas.
- Listado y archivo sobre soporte de las bases de datos generadas.
- Archivo objeto generado.
- Documentación del Sistema.
- Los productos hardware y software utilizados durante el desarrollo.
- La documentación y manuales de los productos hardware y software utilizados durante el desarrollo.
- Datos para la fase de pruebas.
- Documentación del usuario (Manual del Usuario, Manual de Operación, etc.).
- Plan de Formación.
- Plan de Integración.
- Sistema software integrado.
- El código fuente<sup>5</sup>.
- Programas ejecutables.
- Informes de Problemas
- Un prototipo, ejecutable o en papel.

### **Prueba (Testeo):**

- Especificación de las pruebas.
- Los casos de prueba ejecutados y los resultados registrados.

### **Ambiente (Mantenimiento):**

- Histórico de pedidos de mantenimiento.
- Las peticiones de mantenimiento.
- Documentación relacionada con la Gestión de Configuración.
- Recomendaciones de Mantenimiento.
- Plan de Retiro.
- El manual de operación e instalación.

Se debe detallar además los elementos de configuración que pertenezcan a las líneas bases del proyecto, especificándolos por fase del proyecto y por iteraciones dentro de cada fase. Se sugiere que cuando se cree una línea base, todos los elementos de configuración sean etiquetados de forma tal que puedan ser identificados de forma unívoca. Se propone crear una línea base si más del 30% de los elementos en un subsistema han sido modificados.

| FASE: [Fase]                           |   |                                       |
|--|---|---------------------------------------|
| ITERACIÓN: [Iteración]                 |   |                                       |
| Elemento                               | Descripción                                 | Línea de trabajo                      |
| [Nombre del elemento de la Línea Base] | [Descripción del elemento de la Línea Base] | [Línea de trabajo a la que pertenece] |

#### 4.2.4.1.2 Esquema de Identificación de los Elementos de Configuración.

Se propone describir los elementos de configuración que serán nombrados, marcados y numerados. Se sugiere que el esquema de identificación cubra todos los elementos de configuración manejados en el proyecto, por ejemplo: planes, modelos, componentes, prueba de software, datos y resultados, ejecutables, etc.)

El formato para la identificación rápida y almacenamiento de los elementos de Configuración se propone que se realice con 5 elementos de identificación separados por puntos:

**Ejemplo: <C>.<N>.<P>.<T>.<V>**

##### Nomenclatura:

**C:** Código del Elemento de Configuración, el cual se identificará con tres dígitos, se comenzará a enumerar por 001.

**N:** Nombre del Elemento de Configuración (cadena de caracteres).

**P:** Proyecto al que pertenece (cadena de caracteres).

**T:** Tipo de Elemento de Configuración (documento, programa, elemento físico,...).

**V:** Número de versión.

Ejemplo Práctico: **<001>.<Plan de Pruebas>.<Calidad>.<Documento>.<1.0>**

Esta nomenclatura será definida internamente por cada proyecto, aunque es factible estandarizarla a nivel de Facultad.

#### **4.2.4.1.3 Línea Base.**

Con el propósito de controlar los cambios se utiliza el concepto de Líneas Base, las cuales pueden ser vistas como hitos en el proceso de desarrollo del software pues se constituyen en función de la aprobación de uno o varios ECS mediante la ejecución de revisiones técnicas formales. Por este motivo se propone que las líneas bases se establezcan al culminar cada fase de desarrollo del ciclo de vida del software, y en caso que el producto sea de gran envergadura se sugiere además que se definan otras dentro de los flujos de trabajo en dependencia a las necesidades del proyecto.

Se propone contar con las siguientes Líneas Bases al finalizar las fases definidas por RUP:

- Inicio: Línea Base de Requerimientos, Línea Base Funcional.
- Elaboración: Línea Base de Diseño.
- Construcción: Línea Base de Desarrollo, Línea Base de Producción.
- Transición: Línea Base de Operación.

#### **4.2.4.2 Control de la Configuración.**

En esta sección se detallan las actividades de solicitud, evaluación, aprobación e implementación de cambios a los elementos de la línea base. Los cambios apuntan tanto a la corrección como al mejoramiento.

El procedimiento que se describe a continuación es el que se utilizará cada vez que se precise introducir un cambio al sistema.

##### **4.2.4.2.1 Solicitud de Cambios.**

Cuando se realiza la solicitud de un cambio, se actualiza la plantilla de “Solicitud de Cambio” para registrar esta solicitud. Se debe ingresar toda la información necesaria, detallada en el documento.

##### **4.2.4.2.2 Evaluación de Cambios.**

La evaluación del cambio involucra determinar qué es necesario hacer para implementar el cambio y la estimación de sus costos y plazos.

Se realiza en 2 pasos:

1. Planificación de la evaluación del cambio que involucra:

- Revisar la solicitud de cambio para entender su alcance. (Si es necesario se discute con el originador para aclarar el alcance de lo propuesto y los motivos de la solicitud).
- Determinar las personas del proyecto que deben realizar el análisis de evaluación del cambio e involucrarlas.
- Desarrollar un Plan para la evaluación del cambio.
- Si el cambio involucra al cliente, obtener el acuerdo de éste con el Plan.

2. Evaluar el cambio:

La evaluación del cambio puede ser realizado por el Responsable de Configuración o ser delegado a otras personas del proyecto.

Se debe determinar el impacto en:

- Los productos técnicos.
- Los Planes de proyecto.
- Los acuerdos con el Cliente.
- Los Riesgos del proyecto.

#### **4.2.4.2.3 Aprobación y desaprobación de Cambios.**

Se debe formar el “Comité de Control de Cambio” y determinar su autoridad para la aprobación de cambios.

La composición de este comité puede variar según el tipo de cambio y las líneas de trabajo involucradas.

Se sugieren como posibles integrantes:

- Responsable de Configuración (obligatorio).
- Arquitecto (opcional).
- Analista (opcional).
- Desarrollador (opcional).
- Cliente (opcional).

#### **4.2.4.2.4 Implementación de los Cambios.**

Una vez realizada la evaluación del cambio, se decide en qué momento implementarlo. Esta etapa involucra los procesos necesarios para implementar la solicitud y monitorear el progreso del trabajo. Además se especificará el momento de liberación del cambio; así como también los responsables de las actividades que involucra el cambio.

Recordando que se basa en un proceso de desarrollo incremental e iterativo, donde en cada iteración se realizan tareas de Análisis de requerimientos, Diseño, Implementación y Verificación; se debe introducir el cambio en el área que lo originó y continuar con las actividades del ciclo (Requerimientos, Análisis, Diseño, Implementación, Verificación) que impactarán los elementos de la línea base correspondientes a cada actividad.

#### **4.2.4.3 Informes del Estado de la Configuración.**

Se sugiere realizar informes de estado de Configuración, mediante el cual los usuarios, los gestores y los desarrolladores estarán al tanto del estado de la configuración y evolución.

Se debe generar un Informe de estado de Configuración:

- Cada vez que se asigne una nueva identificación a un ECS.
- Cada vez que se lleve a cabo una auditoría<sup>13</sup> de configuración.

#### **4.2.4.4 Auditorías.**

Se realizarán auditorías de la línea base antes de una liberación de ésta o de una actualización de la versión de un componente prioritario de ésta, se propone que se realice como mínimo una auditoría física y una funcional.

Debido a que las auditorías se complementan por la revisión técnica formal, estas deberán incluir:

- Objetivo: Verificar que en un momento dado la línea base se compone de una colección consistente y bien definida de productos.
- Elementos de configuración bajo auditoría<sup>13</sup>: se elegirán uno o más elementos de configuración que formen parte de una línea base teniendo en cuenta la prioridad asignada.
- Agenda de auditorías: antes de la liberación o actualización.
- Participantes: Responsable de Configuración y el Grupo de Aseguramiento de de la Calidad (GAC) tanto interno como a nivel de facultad.

- Documentos Requeridos: Solicitud de Cambio y Reportes de Estado de Configuración.
- Criterio de Aprobación: lo determina el Responsable de Configuración y el Jefe de Proyecto.
- Reportes de Deficiencias y Acciones Correctivas: Determinadas por los participantes.

#### 4.2.4.5 Control de Interfaces.

Las actividades de Control de Interfaces controlan los cambios a los elementos de configuración del proyecto, que modifican las interfaces con elementos fuera del alcance del Plan. Este control será llevado por el Responsable de Configuración como parte del control de la configuración.

#### 4.2.4.6 Control de Subcontratistas/Vendedores.

No procede.

#### 4.2.5 Planificación.

Las Actividades de Administración de Configuración definidas son:

| Actividad  | Entregable Asociado                      | Tiempo           |
|--|--|------------------|
| Elaboración del Plan de Administración de Configuración. | Plan de Administración de Configuración. | 1 semana         |
| Definición de ambientes controlados.                     | Plan de Administración de Configuración. | 1 semana         |
| Mantenimiento de la línea base del Software.             | Informe de la auditoría.                 | diario o semanal |

#### 4.2.6 Herramientas, Técnicas y Metodologías.

La herramienta que se utilice para el control de cambios y versiones deberá cumplir con los requisitos mínimos que se mencionan a continuación:

- Operaciones atómicas (todo o nada).
- Versionamiento de directorios (tener control de que cambio de hizo).
- El modo copiar/modificar/unir tanto de archivos, subdirectorios y directorios (pueden trabajar varios desarrolladores a la vez).
- Posibilidad de definir varias librerías con múltiples estructuras de directorios y subdirectorios.

- Control de acceso a nivel de archivos, subdirectorios, directorios y librerías.
- Modelo de Repositorio cliente-servidor.
- Manejo de versiones de todo tipo de archivos y directorios
- Comparación entre versiones de un producto en específico.
- Recuperar viejas versiones y examinar el historial de cambios.
- Bloqueo de ficheros antes de realizar un cambio: un solo acceso de escritura y múltiples de lectura.
- Identificar claramente los elementos de configuración que corresponden a una versión.
- Permitir el acceso a versiones previas.

#### **4.2.7 Mantención del Plan.**

Este Plan deberá ser revisado al inicio de cada fase, modificado de acuerdo a lo necesario, y claramente establecido, aprobado, y ampliamente conocido por el equipo del proyecto.

#### **4.2.8 Captura y Retención de Registros.**

A continuación se propone que documentos conservar después de haber culminado el desarrollo del software junto con el tiempo de conservación.

Plantilla Ingreso a Configuración: conservar durante 6 años.

Plantilla Lista de Copias Autorizadas: conservar durante 6 años.

Plantilla Solicitud de Cambio: conservar durante 6 años.

Plantilla Nota de Cambio: conservar durante 6 años.

Plantilla de Auditoría: conservar durante 6 años.

### **4.3 Propuesta del Procedimiento para la Administración de Configuración y Cambio**

Como propósito de este trabajo y teniendo en cuenta toda la teoría recopilada de este tema, y la propuesta que se hace del Plan de Administración de Configuración y Cambio, así como las aplicaciones prácticas llevadas a cabo por las empresas de software cubanas citadas, se muestra la propuesta de procedimiento. De igual forma se tiene presente las necesidades de la Universidad de las Ciencias Informáticas (UCI) para realizar esta propuesta.

#### **Procedimiento.**

#### 4.3.1 Propuesta de encabezado.

Tabla (15). Propuesta de encabezado

|   |                                   |            |
|---|-----------------------------------|------------|
|  | <b>No. Revisión</b>               | < X.Y >    |
|   | <b>Fecha de aprobación</b>        | DD/MM/AAAA |
|   | <b>Fecha de<br/>ación</b>         | DD/MM/AAAA |
| <b>Título: Procedimiento para la Administración de la Configuración y Cambio</b>  |                                   |            |
| <b>Aprobado por: nombre de la persona</b>   | <b>Cargo: Gerente de Proyecto</b> |            |

#### 4.3.2 Objetivo.

Establecer el procedimiento para la Administración de Configuración y Cambio en la Universidad de las Ciencias Informáticas (UCI).

#### 4.3.3 Alcance.

Se aplica a todos los proyectos a nivel de facultad dentro de la Universidad de las Ciencias Informáticas (UCI).

#### 4.3.4 Definiciones.

| Palabra, símbolo o abreviatura | Definición                                  |
|--------------------------------|---|
| <b>IC</b>                      | Ingreso a Configuración.                    |
| <b>LCA</b>                     | Lista de Copias Autorizadas.                |
| <b>NC</b>                      | Nota de Cambio.                             |
| <b>SC</b>                      | Solicitud de Cambio.                        |
| <b>RHCP</b>                    | Registro Histórico de Cambios por Proyecto. |
| <b>CCC</b>                     | Comité de Control de Configuración.         |
| <b>CCB</b>                     | Comité de Control de Cambio.                |
| <b>GAC</b>                     | Grupo de Aseguramiento de la Calidad.       |

#### 4.3.5 Referencias.

Aquí se brinda una lista de todos los documentos referenciados que se tomaron como base para la elaboración del procedimiento.

Plan de Administración de Configuración y Cambio: Universidad de las Ciencias Informáticas (Facultad 2).

#### 4.3.6 Responsabilidades..

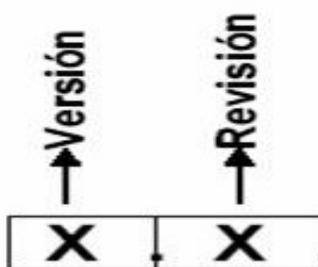
| Por la Acción de:                  | Responde:                        |
|------------------------------------|----------------------------------|
| <b>Aprobar el documento:</b>       | Gerente de Proyecto:             |
| <b>Implantar el documento:</b>     | Responsable del Producto:        |
| <b>Ejecutar las acciones:</b>      | Todos los miembros del proyecto: |
| <b>Supervisar el cumplimiento:</b> | Responsable de Configuración:    |

#### 4.4 Instrucciones.

##### 4.4.1 Control de versiones de Programas o Software.

Se propone que el control de versiones del software (aplicaciones, instaladores,...) se realice con 2 cifras separadas entre ellas por un punto. Se plantea que la primera cifra se refiera a la versión mayor del Software (Versión), la segunda a la versión menor del Software (Revisión), por lo que la primera versión de cada aplicación se enumerará 1.0.

Nota: En caso que un producto haya adquirido varias versiones debido a nuevas funcionalidades agregadas, y la versión con la cual se almacene este producto requiera de un número mayor que dos cifras, se propone agregarle una cifra a la nueva versión del producto.



Se sugiere que el **código de versión** se incremente cada vez que se adicione una nueva funcionalidad al software. Si empieza con un cero significa que el software aún no está listo para su liberación o no cumple con los requerimientos mínimos. Cada cambio en esta cifra denotará una incompatibilidad con versiones mayores anteriores.

Se propone que el **código de revisión** se incremente cada vez que se realice un cambio en el Software pero que no implique una nueva funcionalidad en este, ya sea para la corrección de algún problema o falla detectada.

#### **4.4.2 Control de versiones de Documentos.**

Se propone en el caso de los documentos que el control de versiones se realice con 2 cifras de números enteros decimales separados por un punto. La primera cifra se referirá al código de la versión y la segunda al código de la revisión. La primera versión de cada documento se numerará 1.0.

Se sugiere que el código de versión se incremente cuando los cambios realizados al documento sean mayores e igual al 50% de su contenido.

Se sugiere además que el código de revisión se incremente cuando los cambios realizados al documento sean menores al 50% de su contenido.

#### **4.4.3 Bibliotecas de Trabajo.**

Se propone establecer un sistema bibliotecario integrado por 3 librerías de trabajo mediante las cuales se controlarán todos los elementos de configuración de software (ECS) que se manejen en los proyectos dentro de una facultad determinada, estas son:

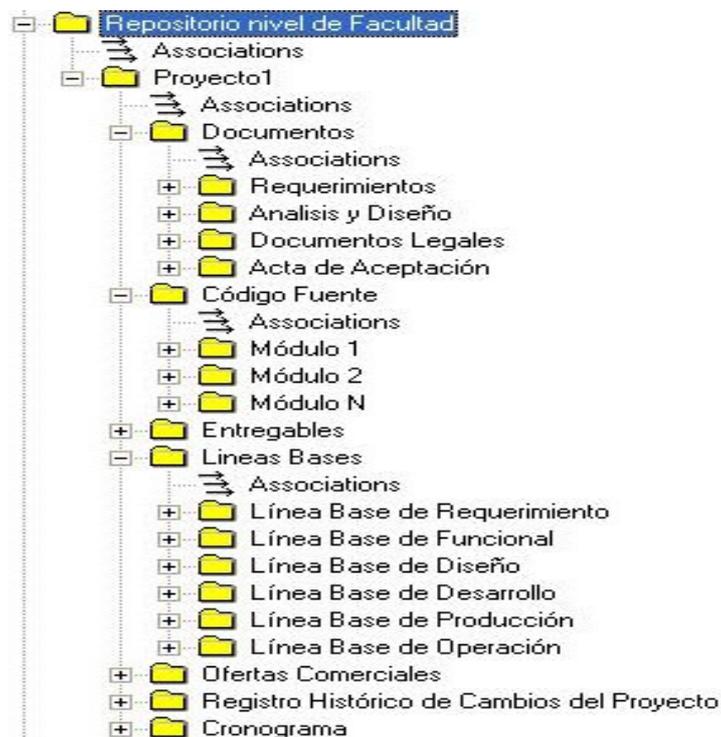
- Biblioteca Dinámica o de Programación.
- Biblioteca Controlada o de Calidad.
- Biblioteca Estática o de Producción.

##### **4.4.3.1 Librería Dinámica o de Programación.**

Se sugiere en esta librería conservar las entidades de software recientemente creadas y/o modificadas dentro de un proyecto de una facultad. Se usará para administrar las líneas base de desarrollado, por lo que se propone que sea de acceso libre y controlado por los desarrolladores del proyecto.

Se propone establecer un Repositorio a nivel de facultad, el cual estará constituido por los distintos proyectos que se lleven a cabo dentro de la facultad. Se propone que dichos proyectos tengan una estructura organizativa de la siguiente forma:

- Entregables: Se sugiere que se almacene todos los elementos de configuración que serán necesarios cuando se quiera recuperar una versión completa del sistema.
- Documentos: Aquí se sugiere que se almacene toda la información generada en el proyecto.
- Líneas Bases: Se propone que se tenga un control mediante esta opción sobre las líneas base que se definan en el proyecto para llevar una panorámica de que elementos de configuración están listos para el posterior desarrollo del proyecto.
- Ofertas Comerciales: Se propone que se almacenen todas las propuestas para un nuevo desarrollo de software, o todo lo que implique nuevas ofertas de trabajo.
- Registro Histórico de Cambios del Proyecto: Se propone que se almacene todas las solicitudes de cambios que hayan sido aprobadas.
- Código Fuente<sup>5</sup>: Se propone que se almacene todo el código fuente<sup>5</sup> utilizado.
- Cronograma: Se sugiere que se registre la planificación de cada tarea en el proyecto para llevar un control de los objetivos a cumplir.



#### **4.4.3.2 Librería Controlada o de Calidad.**

Se sugiere que esta librería contenga todos los elementos de configuración ya desarrollados y probados, es decir, a los que se le realizaron las pruebas unitarias<sup>21</sup> exitosamente. Se propone que solo tengan acceso a esta librería el Grupo de Aseguramiento de la Calidad y el Responsable de Configuración, los cuales podrán obtener copias libres de los elementos de configuración.

#### **4.4.3.3 Librería Estática o de Producción.**

Se sugiere que en esta librería se almacene todos los elementos de producción ya listos para su liberación, además de llevar el control de todo el proceso de soporte al producto. Se propone que sea controlada por el Responsable de Configuración y por el Comité de Control de Configuración. Se sugiere que esta librería contenga las copias maestras y de seguridad de los elementos de configuración, se propone que sea de acceso restringido y que solo se puedan obtener copias autorizadas.

#### **4.4.4. Descripción de las Actividades.**

**1)** Todo Elemento de Configuración (ECS) (documento o programa) que se encuentre en la librería dinámica o de programación, se promoverá a la librería controlada o calidad cuando los desarrolladores hayan efectuado las pruebas unitarias<sup>21</sup> y sus resultados hayan sido exitosos.

**2)** Una vez que los ECS se encuentren almacenados en la librería de Calidad, el GAC realizará las revisiones técnicas formales correspondientes. En caso que se detecte un defecto en el software o código del producto ocurrirá un proceso de remoción hacia la librería de programación, hasta que el defecto sea corregido por parte de los desarrolladores y las pruebas unitarias<sup>21</sup> hayan sido exitosas. Luego el/los ECS se liberará de nuevo a la librería de calidad, en la cual se le harán las pruebas correspondientes (GAC).

- 3) En caso que las pruebas realizadas a el/los ECS fueran exitosas, estos se liberarán a la librería de producción, iniciándose con esto el ingreso del ECS al proceso de Control de Configuración. Seguidamente Responsable de Configuración le comunicará al Comité de Control de Configuración que hubo un ingreso a configuración.
- 4) Al iniciarse este proceso, existirá la plantilla de **Ingreso a Configuración (IC)** y la de **Lista de Copias Autorizadas (LCA)**, la cuales serán actualizadas y controladas por el Comité de Control de Configuración.
- 5) Una vez que el producto se encuentren en la librería de producción, estará listo para el proceso de instalación por parte del cliente. En caso que el cliente después de haber probado el producto detecte algún error en el sistema o desee agregarle una nueva funcionalidad, esto implicará un cambio en el producto, por lo que deberá realizar una solicitud de cambio formal, la que se documentará a través de una plantilla de **Solicitud de Cambio (SC)**.

**Los integrantes del grupo de desarrollo de un proyecto de una facultad, podrán realizar “Cambios Informales” a los ECS que ellos mismos hayan desarrollado antes que formen parte de una Línea Base y se hayan introducido a la base de datos del proyecto. Una vez que los desarrolladores de un proyecto realicen una solicitud de cambio a otro proyecto, porque estén trabajando en conjunto, esta se documentará a través de una plantilla de Solicitud de Cambio (SC).**

- 6) El cliente le entregará esta plantilla al Gerente de Proyecto a nivel de facultad, el cual estará encargado de entregar esta solicitud de cambio al Responsable del Producto sobre el cual se solicitó dicho cambio.

**El desarrollador o grupo de desarrolladores del proyecto que realizaron la solicitud de cambio le entregarán esta plantilla al Responsable del Producto sobre el cual se haya realizado la solicitud de cambio.**

- 7) El Responsable del Producto verificará el contenido de dicha plantilla y en caso de una no conformidad se la devolverá a su originador para su corrección. Una vez correcto todos los datos le entregará dicha

solicitud al Responsable de Configuración, la cual registrará, asignará un número y un estado para su seguimiento y control.

8) El Responsable de Configuración remitirá la solicitud de cambio al Comité de Control de Cambios, el cual analizará y evaluará la solicitud de cambio, aprobando, rechazando o postergando la implementación de los cambios teniendo en cuenta:

- Envergadura del cambio.
- Complejidad del cambio, en relación a los sistemas involucrados.
- Fecha en que se necesita implementarlo.
- Impacto en el trabajo actual y en el futuro.
- Criticidad del área involucrada.
- Cambios aprobados, actualmente en desarrollo.
- Requerimientos de pruebas para validar el cambio.
- Políticas.
- Líneas base afectadas.
- Las prioridades.
- La urgencia del cambio.
- La gravedad de la falla detectada.

9) En caso de ser postergada o rechazada la solicitud de cambio, el Comité de Control de Cambio la devolverá al Responsable del Producto, explicándole la razón de la decisión tomada, las que se harán llegar a la persona que realizó la petición del cambio mediante el Gerente de Proyecto.

**En caso de ser postergada o rechazada la solicitud de cambio, el Comité de Control de Cambio del proyecto sobre el cual se hizo la solicitud de cambio la devolverá al Responsable de Configuración, explicándole la razón de la decisión tomada, las que se harán llegar a los desarrolladores del proyecto que realizaron la solicitud de cambio mediante el Responsable del Producto.**

- 10)** En caso de ser aprobada la solicitud de cambio por parte del Comité de Control de Cambio, esta será evaluada nuevamente por el Comité de Control de Configuración para estimar el esfuerzo técnico que demandará su implementación, los costos asociados y los posibles efectos secundarios e impacto sobre otras funciones del sistema, el cual se le informará al originador de la solicitud de cambio, si está de acuerdo con los costos impuestos y el tiempo de desarrollo del cambio se procederá al proceso de implementación del cambio.
- 11)** Posteriormente las copias de seguridad de los ECS serán removidos desde la librería de producción hacia la librería de programación para el desarrollo del cambio. Seguidamente el Comité de Control de Configuración registrará en la plantilla de Listas de Copias Autorizadas (LCA) los desarrolladores que tienen autorización para efectuar copias de trabajo de el/los elementos de configuración a modificar.
- 12)** Una vez actualizada dicha plantilla, el Responsable de Configuración informará a todos los incluidos en esta lista que tienen derecho a realizar cambios sobre el/los elementos de configuración registrados.
- 13)** El Responsable de Configuración además informará a los desarrolladores que la autorización a realizar copias de trabajo de el/los ECS podrán quedar obsoletas cuando el/los ECS se encuentren en estado cerrado. Le informará también que ya pueden comenzar con la implementación de los cambios solicitados. A partir de este momento se inmovilizará el/los ECS sobre el que se realizará los cambios y los ECS relacionados, con el objeto de evitar superposición de cambios.
- 14)** El Comité de Control de Cambio será el responsable del seguimiento y control de la implementación de los cambios, en cuanto a cumplimiento de objetivos, metas y plazos; verificación y validación de las modificaciones.
- 15)** Los desarrolladores luego de haber finalizado la implementación de los cambios y haber efectuado las pruebas unitarias<sup>21</sup> correspondientes, si fueron exitosas, el/los ECS se promoverán a la librería de calidad, para que el GAC del proyecto realice las revisiones técnicas formales con el objetivo de certificar que el problema se ha corregido o se han satisfecho los nuevos requisitos. Si las pruebas fueron exitosas los desarrolladores actualizarán la plantilla Nota de Cambio, para certificar que el cambio fue realizado satisfactoriamente.

16) Dicha plantilla se le hará llegar al Comité de Control de Configuración, el cual verificará su contenido y en caso de una no conformidad se la devolverá a su originador para su corrección.

17) Una vez correcto los datos, el Comité de Control de Configuración le enviará dicha plantilla al Responsable de Configuración, el cual registrará, asignará un número y un estado. Posteriormente entregará dicha plantilla al Responsable del Producto para que tenga constancia que el cambio fue efectuado.

18) Luego de haber informado al Responsable del Producto, este verificará el/los ECS modificados, y si está de acuerdo con los cambios implementados, los ECS serán promovidos hacia la biblioteca de producción, donde se almacenará la nueva versión del producto o el producto con los errores corregidos. Seguidamente el Grupo de Aseguramiento de la Calidad a nivel de Facultad realizará las auditorías para verificar que todo el proceso de control de cambio se realizó de acuerdo al procedimiento establecido.

19) Si la auditoría<sup>13</sup> no tuvo un resultado satisfactorio el Grupo de Aseguramiento de la Calidad a nivel de Facultad se lo comunicará al Responsable del Producto, el cual se lo devolverá al Responsable de Configuración, el cual tomará las medidas pertinentes para la corrección correspondiente.

20) Si la auditoría fue exitosa, el Grupo de Aseguramiento de la Calidad de la Facultad se lo comunicará al Responsable del Producto, el cual le hará llegar el producto al Gerente de Proyecto y almacenará en el Registro Histórico de Cambios de Proyectos (RHCP) la plantilla Nota de Cambio, para tener un registro de todos los cambios realizados en el proyecto.

**Si la auditoría<sup>13</sup> fue exitosa el Responsable del Producto les comunicará a los integrantes del proyecto (desarrolladores) que realizaron la solicitud, que el cambio fue efectuado de forma exitosa y almacenará en el Registro Histórico de Cambios de Proyectos (RHCP) la plantilla Nota de Cambio, para tener un registro de todos los cambios realizados en el proyecto.**

21) Una vez que el Gerente de Proyecto esté de acuerdo con el producto entregado, el Responsable de Configuración retirará de circulación y destruirá todas las copias autorizadas, y ahora obsoletas, del ítem modificado, asignándole el estado de cerrado.

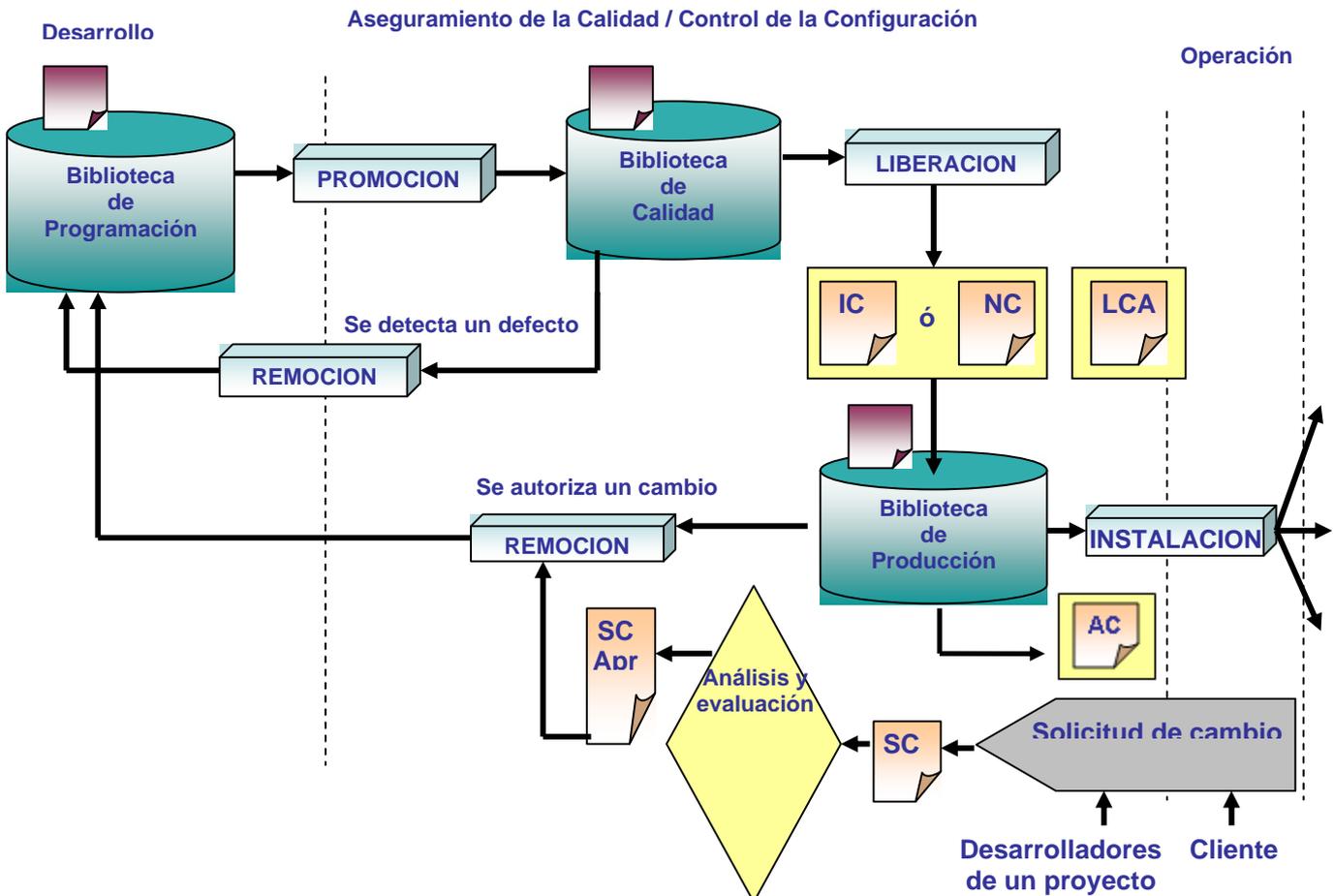
Responsable de Configuración del proyecto sobre el cual se realizó la solicitud de cambio retirará de circulación y destruirá todas las copias autorizadas, y ahora obsoletas, del ítem modificado, y la reemplazará la versión anterior del producto, por la nueva versión/revisión.

22) Como resultado final el Gerente de Proyecto entregará el producto con los cambios efectuados al originador de la solicitud de cambio.

23) Como resultado final el Responsable del Producto entregará el producto con los cambios efectuados a los desarrolladores del otro proyecto.

#### 4.4.5 Representación Gráfica de la descripción de las Actividades.

### Administración de la Configuración



#### **4.7 Conclusiones Parciales del capítulo 4**

- En este capítulo se realizó la propuesta de un Plan de Administración de Configuración y Cambio, así como la propuesta de un procedimiento que sirve de guía a los proyectos de la Universidad de las Ciencias Informáticas (UCI), a la hora de llevar el control de la Gestión de Configuración.
- Se tomó como referencia toda la teoría recopilada durante la investigación de este trabajo, así como los procedimientos empleados por las empresas cubanas de software como La Casa de Software SIGTA, DESOFT, SOFTEL Y MoProsoft

## CONCLUSIONES GENERALES

Después del estudio realizado referente a la Administración de la Configuración y Cambio se demostró el papel esencial que ocupa esta disciplina en el proceso de desarrollo de software, puesto que examina aspectos tan importantes como el control de versiones y control de cambio, el proceso de administración de los elementos de configuración, las líneas bases y librerías existentes.

Después del estudio realizado a cerca de las herramientas de Administración de la Configuración y Cambio, se ha visto que existen muchas en el mercado internacional, unas son licenciadas y otras son de código abierto, tienen similitudes en cuanto a funcionalidades respecto a la utilización de librerías y a la estructura del modelo del repositorio. Muchas se acomodan a las necesidades de los proyectos de la universidad con el objetivo de controlar los cambios y las versiones por las que transcurre el software durante su desarrollo.

Se obtuvieron procedimientos de referencia utilizados por instituciones que se dedican a la producción de software en nuestro país para tener conocimiento de los parámetros a tener en cuenta a la hora de proponer una política a nivel de universidad, que estandarice la utilización de un procedimiento a seguir.

Con la propuesta tanto del procedimiento como del plan de administración de configuración y cambio se logra establecer una disciplina de trabajo en las tareas asociadas al control de cambios teniendo en cuenta la descripción de las actividades planteadas. Se logra además detallar una propuesta referente al control de versiones de programas y documentos. Se identifica los ECS en dependencia a las necesidades de los proyectos en la Universidad de las Ciencias Informáticas.

Por lo anteriormente planteado se llega a la conclusión que se cumplieron todos los objetivos y tareas trazados al inicio de esta investigación.

## RECOMENDACIONES

- 1) Aplicar el procedimiento para la administración de la configuración y cambio en los proyectos de la Universidad de las Ciencias Informáticas y evaluar sus resultados.
- 2) Enriquecer el estudio de las herramientas analizadas para la administración de la configuración y cambio teniendo en cuenta otras mencionadas en el capítulo 2, para tener un conocimiento más profundo de cuales de las herramientas existentes en el mercado mundial serían las propicias para solucionar muchos de los problemas en esta área de proceso clave.

## REFERENCIA BIBLIOGRÁFICA

- [1] Angélica de Antonio. La Gestión de la Configuración del Software. Volumen (59): 1-59.
- [2] Addison Wesley Longman Inc. El Proceso Unificado de Desarrollo de Software, 2000.
- [3] Bamford, R. and W. J. Deibler, "Configuration Management and ISO 9001." SSQC, 1995.
- [4] Ernesto Quiñones A. Modelos de Calidad de Software y Software Libre  
[Disponible en: <http://www.apesol.org.pe>]
- [5] Empresa de Telecomunicaciones de Cuba, Gerencia Casa de Software SIGTA.
- [6] Febles Ailyn Dra. Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software. CUJAE. La Habana, Cuba. 2004.
- [9] Watts Humphrey. Administrando el Proceso de Software. 1989.
- [10] Wikipedia, Software, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Software>]

## BIBLIOGRAFÍA

Angélica de Antonio. La Gestión de la Configuración del Software. Volumen (59): 1-59.

Addison Wesley Longman Inc. El Proceso Unificado de Desarrollo de Software, 2000.

Ayuda del Rational.2007

Artículo de Plastic SCM .Codice Software .2007.[Disponible en: <http://www.mkm-pi.com/spip.php?article130>]

Artículos de información técnica Visual Studio team System. [Disponible en: <http://www.danysoft.com/bol/060504.htm>]

Alejandro Martorell. Visual Studio Team System. 2006.[Disponible en: [http://athenea.ort.edu.uy/publicaciones/ingsoft/investigacion/ayudantias/VSTS\\_GestionConfiguracion.pdf](http://athenea.ort.edu.uy/publicaciones/ingsoft/investigacion/ayudantias/VSTS_GestionConfiguracion.pdf)]

AMCIS. Modelo de calidad de desarrollo de software. Moprosoft .2007.México.

Bamford, R. and W. J. Deibler, "Configuration Management and ISO 9001." SSQC, 1995.

Brenda Bastida Martínez Lic. Fundamentos de Administración de la Configuración. [Disponible en: [http://www.avantare.com/articulos/novedades\\_articulodelmes.html](http://www.avantare.com/articulos/novedades_articulodelmes.html).]

Borland StarTeam® (Manage) [Disponible en: <http://www.samsistemas.com.ar/borland/starteam.htm> ]

Borland StarTeam. [Disponible en: <http://www.quinaria.cl/Borland%20Starteam.htm>]

Ben Collins-Sussman, Brian W. Fitzpatrick, y C. Michael Pilato. Control de versiones con Subversion: Revision 2805. [Disponible en: <http://svnbook.red-bean.com/nightly/es/svn-book.pdf>]

CMM. Carnegie Mellon Software Engineering Institute. The Capability Maturity Model. USA.1993

CMM. Carnegie Mellon Software Engineering Institute. The Capability Maturity Model. USA.2005  
[Disponible en: <http://www.SEI.com>]

Código software. Arbol de versiones, 2007.[Disponible en : <http://www.codicesoftware.es/esproducts/esclient/esvtree.aspx>]

Codice Software.Ramas. [Disponible en: <http://www.codicesoftware.es/esproducts/esserver/esbranch.aspx>]

Departamento de Control de Calidad y Auditoría Informática. Control de calidad en los Sistemas.2002. [Disponible en:

<http://sistemas.dgsca.unam.mx/publica/pdf/Control%20de%20Calidad.PDF>]

Descripción de las mejoras de Gforge. [Disponible en: [http://www.morfeo-project.org/index.php?option=com\\_content&task=view&id=24&Itemid=42morfeo](http://www.morfeo-project.org/index.php?option=com_content&task=view&id=24&Itemid=42morfeo)]

Empresa de Telecomunicaciones de Cuba, Gerencia Casa de Software SIGTA, Manual para la Gestión de la Configuración.

Ernesto Quiñones A. Modelos de Calidad de Software y Software Libre

[Disponible en: <http://www.apesol.org.pe>]

Empresa de desarrollo de software. DESOFT. Empresa de desarrollo de software. SOFTEL. 2007. La Habana. Cuba.

Empresa de desarrollo de software. SOFTEL. Documento Plan de Gestión de la Configuración del Software.2007. La Habana. Cuba.

Febles Ailyn Dra. Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software. CUJAE. La Habana, Cuba. 2004.

Fundamentos del Rational ClearCase.[Disponible en:

[http://www.indudata.com/cursos/5rational\\_clear\\_case.htm](http://www.indudata.com/cursos/5rational_clear_case.htm) ]

Gartner.Inc. Describing the Capability Maturity Model. 2001

Gestión de configuración. 2007. [Disponible en:

[http://www.lcc.uma.es/~guzman/gp/docs/29\\_MAY\\_MAN\\_1.pdf](http://www.lcc.uma.es/~guzman/gp/docs/29_MAY_MAN_1.pdf)]

Gestión de configuración. 2007 [Disponible en:

<http://www.fing.edu.uy/inco/cursos/gestsoft/ppts/gs13.ppt#445,18>]

Gestión de Configuración. [Disponible en:

<http://kybele.escet.urjc.es/documentos/ISG/Segundo/2006-2007/GestionConfiguracion.pdf>]

Herramientas y Soluciones IBM. [Disponible en:

<http://www.rational.com.ar/herramientas/clearcase.html>]

IEEE Standard for Software Configuration Management ", IEEE Computer Society, 1990.

Junior A. Altamiranda P .Ing. Requerimientos y servicios de Gforge, 2006.

[Disponible en: <http://sistemas.fsl.funmrd.gov.ve/docman/view.php/16/22/GFORGE.pdf>]

Moisés Maciá. Eclipse, EL entorno de desarrollo, 2006. [Disponible en:

<http://quarkblog.org/2006/09/12/eclipse-el-entorno-de-desarrollo/>]

Plastic SCM Platform free for open source. Codice Software S.L. [Disponible en:

[www.codicesoftware.com](http://www.codicesoftware.com)]

Plastic SMC Platform [Disponible en: <http://www.danyshop.com>]

Plastic SCM platform.[Disponible en: <http://www.codicesoftware.com/content/catalogo.pdf>]

Plastic SCM [Disponible en: <http://www.codicesoftware.com/whitepapers/overview.pdf>].

Rational Unified Process, IBM, 2003.

Rational ClearQuest. [Disponible en: [http://www.indudata.com/1rational\\_clear\\_quest.htm](http://www.indudata.com/1rational_clear_quest.htm)]

Sowre Consulting España. Valor añadido de SOWRE en IBM Rational.2007. [Disponible en:

<http://www.sowre.es>]

Sitio Oficial de PVCS. [Disponible en: [www.Merant.com](http://www.Merant.com)]

Sitio Oficial. [Disponible en: [www.ca.com/allfusion](http://www.ca.com/allfusion)]

SCADA.2007. proyecto de la Universidad de las Ciencias Informáticas (UCI). Plan de Gestión de Configuración. La Habana. Cuba

Visual Studio Team System2005. [Disponible en:

<http://www.danyshop.com/shop/prod/catalog%5Fname,Software/product%5Fid,1%2D00002/VisualStudio.NET2005.htm>]

Visual SourceSafe.2007.[Disponible en : [http://msdn2.microsoft.com/es-](http://msdn2.microsoft.com/es-es/library/th1ac915(VS.80).aspx)

[es-](http://msdn2.microsoft.com/es-es/library/th1ac915(VS.80).aspx)  
[library/th1ac915\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/th1ac915(VS.80).aspx)]

Wikipedia, Software, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Software>]

Watts Humphrey. Administrando el Proceso de Software.1989.

## GLOSARIO DE TÉRMINOS

Test de regresión<sup>1</sup>: es cualquier tipo de pruebas de software las cuales buscan descubrir bugs de regresión.

Bifurcaciones<sup>2</sup>: Una bifurcación o fork en el ámbito de desarrollo de software es la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código del proyecto ya existente. Comúnmente se utiliza el término inglés.

Framework<sup>3</sup>: Marco de trabajo.

Builds<sup>4</sup>: Construir una versión de un producto.

Código Fuente<sup>5</sup>: el programa en su forma original, tal y como fue escrito por el programador, el cual debe convertirse en lenguaje de maquina mediante compiladores, ensambladores o intérpretes.

Unicode<sup>6</sup>: Es un estándar industrial cuyo objetivo es proporcionar el medio por el cual un texto en cualquier forma e idioma pueda ser codificado para el uso informático.

WorkSpace<sup>7</sup>: un ambiente unificado para el desarrollo de aplicaciones, es el primero en unir la brecha entre la visión de una arquitectura orientada a servicios (SOA) y la realidad de las herramientas de desarrollo tradicionales.

Merge<sup>8</sup>: mezclar, fusionar. La función Merge realiza operaciones de combinación.

WebSphere<sup>9</sup> : es actualmente conocida como una familia de productos de software propietario de IBM, aunque el término se refiere de manera popular a uno de sus productos específicos: WebSphere Application Server (WAS) está diseñado para configurar, operar e integrar aplicaciones de e-business a través de varias plataformas de red usando las tecnologías del Web.

Línea Middleware<sup>10</sup>: es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

Releases<sup>11</sup>: liberaciones de un producto de software.

IVSS<sup>12</sup>: es una colección de interfaces de automatización basadas en COM del espacio de nombres Microsoft.VisualStudio.SourceSafe.Interop que Visual SourceSafe expone a los usuarios. Proporciona funcionalidad tanto para las interacciones de automatización con la base de datos de Visual SourceSafe como para las respuestas a los eventos de la base de datos.

Auditoría<sup>13</sup>: es una evaluación cuyo único fin es detectar errores y señalar fallas

Línea OPC<sup>14</sup>: el bus OPC (OLE para Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

Service Pack<sup>15</sup>: grupo de parches que actualizan, corrigen y mejoran aplicaciones y sistemas operativos.

Servicio LAN Booster<sup>16</sup>: es un servicio de NT para el acceso de red de área local (LAN) a la base de datos de Visual SourceSafe. Puede configurar este servicio para mejorar el rendimiento del control de código fuente del complemento de LAN de SourceSafe para Visual Studio. Si opta por no utilizar el servicio, el complemento puede seguir obteniendo acceso a la base de datos siguiendo métodos de red estándar.

Testers<sup>17</sup>: es un usuario de programas cuyos ejecutables están pendientes de terminar su fase de desarrollo, o alcanzar un alto nivel de funcionamiento, pero que aún no son completamente estables. Los testers usan sus conocimientos informáticos y su tiempo, para detectar errores y así poder informar de éstos para que los desarrolladores los corrijan, o corregirlos ellos mismos.

Open source<sup>18</sup>: código abierto (del inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. Fue a utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).

Pruebas de manual<sup>19</sup>: sirven para realizar tareas paso a paso, normalmente para pruebas que son difíciles o imposibles de automatizar, como una prueba en la que la conexión entre el equipo y su fuente de alimentación se interrumpe.

Pruebas load<sup>20</sup> (carga): se utilizan para encapsular pruebas no manuales, es decir, unitarias, Web, genéricas, por orden, y luego ejecutarlas simultáneamente a través de usuarios virtuales. La ejecución de estas pruebas con carga genera resultados, incluidos contadores de rendimiento y de otros tipos, en tablas y en gráficos.

Pruebas Unitarias<sup>21</sup>: son comprobaciones que se hacen a las unidades lógicas de un programa. Se verifica que una unidad funciona correctamente por sí misma, sin tener en cuenta las relaciones que pueda tener con otras partes del sistema.

APIs<sup>22</sup>: Application Programming Interface - Interfaz de Programación de Aplicaciones es el conjunto de funciones y procedimientos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción. Una API representa un interfaz de comunicación entre componentes software.

MultiSite<sup>23</sup>: es una opción que permite una mejor visión, capacidad de predicción y control del proceso de desarrollo de software entre entornos geográficamente distribuidos.

Hackability<sup>24</sup>: se refiere a que la herramienta Subversion es extremadamente manipulable por distintas aplicaciones y lenguajes de programación.

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.