

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**DISEÑO E IMPLEMENTACIÓN DEL PROCESO DE GESTIÓN
DE LA CONFIGURACIÓN DE SOFTWARE EN LA EMPRESA
DE DESARROLLO VENTURE VENTI**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: WILSON SANTIAGO PAREDES GAIBOR

SANGOLQUÍ, Octubre de 2011

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. WILSON SANTIAGO PAREDES GAIBOR como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA

Fecha

ING. CECILIA HINOJOSA

DEDICATORIA

Este trabajo y el esfuerzo de estos años de estudio son dedicados a mi madre, ya que ella ha sido mi pilar y fortaleza, un excelente ejemplo a seguir y a la que le debo la persona que soy. Esta es una pequeña forma de agradecer todo el esfuerzo y sacrificio que ha hecho por mí en todos estos años.

Gracias por siempre estar a mi lado y poder contar contigo cuando lo necesito.

Wilson Santiago Paredes Gaibor

AGRADECIMIENTOS

Agradezco a todas personas que fueron parte de mi vida en esta etapa, a mis ingenieras Cecilia Hinojosa y Jenny Ruiz por su tiempo y haberme encaminado para la realización de esta tesis. Mi familia al apoyarme siempre de alguna forma, en especial a mi tía Ana Lucía que estuvo a mi lado en todo momento. Mis compañeros de aula, con quienes compartí muchos momentos, recibí y brindé, apoyo, cariño y aprecio.

Wilson Santiago Paredes Gaibor

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	II
DEDICATORIA	III
AGRADECIMIENTOS	IV
ÍNDICE DE CONTENIDOS	V
LISTADO DE TABLAS	VIII
LISTADO DE FIGURAS	IX
GLOSARIO	X
NOMENCLATURA UTILIZADA	XI
RESUMEN	1
CAPÍTULO I: GENERALIDADES	2
1.1. INTRODUCCIÓN	2
1.2. DESCRIPCIÓN DEL PROBLEMA	2
1.3. JUSTIFICACIÓN	3
1.4. OBJETIVOS	4
1.4.1. <i>Objetivo General</i>	4
1.4.2. <i>Objetivos Específicos</i>	4
1.5. ALCANCE	4
CAPÍTULO II: MARCO TEÓRICO	6
2.1. GESTIÓN DE LA CONFIGURACIÓN DE SOFTWARE	6
2.1.1. <i>GCS y la mejora de procesos.</i>	7
2.1.2. <i>Mediciones y Métricas.</i>	7
2.1.3. <i>Beneficios de la Gestión de la Configuración del Software</i>	8
2.1.4. <i>Actividades de GCS</i>	9
2.2. IDENTIFICACIÓN DE LA CONFIGURACIÓN	9
2.3. CONTROL DE VERSIONES	11
2.4. CONTROL DE CAMBIOS	12
2.5. PRUEBAS	15
2.5.1. <i>Pruebas de Caja Negra (Funcionales)</i>	15
2.5.2. <i>Pruebas de Caja Blanca (Estructurales)</i>	16
2.5.3. <i>Pruebas de Caja Gris (Estructural y Funcional)</i>	16
2.5.4. <i>Pruebas Manuales versus Automatizadas</i>	17
2.6. GENERACIÓN DE INFORMES SOBRE CAMBIOS DE CONFIGURACIÓN	17
2.7. AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE	18
2.7.1. <i>Auditoría de Configuración Funcional</i>	19
2.7.2. <i>Auditoría de Configuración Física</i>	19
2.8. PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE	19
2.8.1. <i>Métrica Versión 3</i>	19
2.8.2. <i>ISO/IEC 12207: Proceso del ciclo de Vida del Software</i>	24
2.8.3. <i>IEEE 828-1998 – Estándar para elaboración de planes de Gestión de la Configuración del Software</i>	27

2.8.4.	<i>Rational Unified Process(RUP)</i>	31
2.9.	VENTAJAS Y DESVENTAJAS DE ESTÁNDARES Y MODELOS RELACIONADAS AL PROCESO DE GCS	35
2.9.1.	<i>Métrica Versión 3</i>	35
2.9.2.	<i>ISO/IEC 12207</i>	36
2.9.3.	<i>IEEE 828-1998</i>	36
2.9.4.	<i>Rational Unified Process (RUP)</i>	37
CAPÍTULO III: EVALUACIÓN DE LA SITUACIÓN ACTUAL DEL PROCESO DE GESTIÓN DE LA COFIGURACIÓN DE SOFTWARE Y PRUEBAS EN LA EMPRESA		38
3.1.	INTRODUCCIÓN	38
3.2.	ASPECTOS TECNOLÓGICOS	39
3.2.1.	<i>Software Personal de Desarrollo</i>	39
3.2.2.	<i>Software Gerencia de Tecnología</i>	39
3.2.3.	<i>Software Servidor Principal</i>	40
3.3.	ASPECTOS PROCEDIMENTALES	40
3.3.1.	<i>Gestión de Cambios por requerimientos nuevos de un cliente</i>	40
3.3.2.	<i>Gestión de Cambios por corrección de error a un cliente</i>	42
3.3.3.	<i>Gestión de cambios por procesos nuevos del sistema</i>	44
3.3.4.	<i>Control de Versiones</i>	45
3.3.5.	<i>Pruebas</i>	46
3.3.6.	<i>Problemas actuales de la empresa</i>	47
CAPÍTULO IV: ANÁLISIS COMPARATIVO ENTRE METODOLOGÍAS Y ESTÁNDARES RELACIONADOS AL PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE.....		49
4.1.	INTRODUCCIÓN	49
4.2.	IDENTIFICADORES DE EVALUACIÓN PARA EL PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE	49
4.3.	PESOS PARA EVALUACIÓN	49
4.4.	EVALUACIÓN: IMPLEMENTACIÓN DEL PROCESO	50
4.5.	EVALUACIÓN: IDENTIFICACIÓN DE LA CONFIGURACIÓN	51
4.6.	EVALUACIÓN: MANTENIMIENTO Y CONTROL DE LA GESTIÓN DE LA CONFIGURACIÓN	52
4.7.	EVALUACIÓN INFORME DE ESTADO DE LA CONFIGURACIÓN	53
4.8.	EVALUACIÓN AUDITORÍA DE LA CONFIGURACIÓN	54
4.9.	CONCLUSIÓN DE LA EVALUACIÓN	54
4.10.	RESUMEN METODOLOGÍA PROPUESTA	55
CAPÍTULO V: GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE		56
5.1.	MODELO Y/O ESTÁNDAR IDENTIFICADO	56
5.2.	IMPLEMENTACIÓN DEL PROCESO	56
5.3.	IDENTIFICACIÓN DE LOS ECS(ELEMENTOS DE LA CONFIGURACIÓN DEL SOFTWARE)	56
5.3.1.	<i>Selección de los ECS</i>	56
5.3.2.	<i>Identificación Líneas Base</i>	58
5.3.3.	<i>Esquema de Identificación</i>	59
5.3.4.	<i>Definiciones y Establecimiento de Bibliotecas Software</i>	60
5.4.	PRUEBAS	62
5.4.1.	<i>Planeamiento de Pruebas</i>	62
5.4.2.	<i>Realización de Pruebas</i>	64
5.4.3.	<i>Entorno de Pruebas</i>	66

5.5. CONTROL DE VERSIONES	66
5.5.1. <i>Herramientas de Versionamiento</i>	66
5.6. CONTROL DE CAMBIOS.....	73
5.6.1. <i>Petición de cambio</i>	73
5.6.2. <i>Evaluación del cambio</i>	73
5.6.3. <i>Informe de Cambios</i>	74
5.6.4. <i>Generación orden de Cambio de Ingeniería</i>	74
5.6.5. <i>Objeto dado de baja</i>	75
5.6.6. <i>Realización del cambio</i>	75
5.6.7. <i>Revisión del cambio</i>	76
5.7. GENERACIÓN DE INFORMES SOBRE CAMBIOS DE CONFIGURACIÓN	77
5.8. AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE	77
5.8.1. <i>Introducción</i>	77
5.8.2. <i>Realizar Auditoría de la Configuración Física</i>	77
5.8.3. <i>Realizar Auditoría de la Configuración Funcional</i>	78
5.8.4. <i>Reporte de Novedades</i>	78
5.9. SEGUIMIENTO GCS EN LA EMPRESA.....	79
5.9.1. <i>Resultados obtenidos con la aplicación de la GCS</i>	79
5.9.2. <i>Análisis de los resultados obtenidos</i>	81
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	82
6.1. CONCLUSIONES.....	82
6.2. RECOMENDACIONES	83
BIOGRAFÍA.....	86
HOJA DE LEGALIZACIÓN DE FIRMAS	87

LISTADO DE TABLAS

TABLA 2.1. TAREAS DEL REGISTRO DEL CAMBIO EN EL SISTEMA DE GESTIÓN DE LA CONFIGURACIÓN MÉTRICA V3	23
TABLA 2.2. ESTRUCTURA ESTÁNDAR IEEE 828-1998	27
TABLA 3.1. SOFTWARE PERSONAL DE DESARROLLO VENTURE VENTI	39
TABLA 3.2. SOFTWARE GERENCIA DE TECNOLOGÍA VENTURE VENTI.....	39
TABLA 3.3. SOFTWARE SERVIDOR PRINCIPAL VENTURE VENTI.....	40
TABLA 3.4. PROBLEMAS, CANTIDADES Y PORCENTAJES RELACIONADOS AL GCS VENTURE VENTI (ABR-DIC 2010)...	48
TABLA 4.1. PESOS PARA LA EVALUACIÓN GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE.....	50
TABLA 4.2. EVALUACIÓN IMPLEMENTACIÓN DEL PROCESO.....	50
TABLA 4.3. EVALUACIÓN IDENTIFICACIÓN DE LA CONFIGURACIÓN.....	51
TABLA 4.4. EVALUACIÓN DEL MANTENIMIENTO Y CONTROL DE LA GESTIÓN DE LA CONFIGURACIÓN.....	52
TABLA 4.5. EVALUACIÓN DEL INFORME DE ESTADO DE LA CONFIGURACIÓN	53
TABLA 4.6. EVALUACIÓN AUDITORÍA DE LA CONFIGURACIÓN	54
TABLA 4.7. CONCLUSIÓN DE EVALUACIONES	55
TABLA 4.8. RESUMEN DE METODOLOGÍA PROPUESTA	55
TABLA 5.1. ELEMENTOS CONFIGURACIÓN DEL SOFTWARE VENTURE VENTI.....	57
TABLA 5.2. NOMENCLATURA LÍNEAS BASE	58
TABLA 5.3. HERRAMIENTAS VERSIONAMIENTO APLICACIONES WEB.....	67
TABLA 5.4. COMANDOS PARA VERSIONAMIENTO NETBEANS	70
TABLA 5.5. TABLA DE TÓPICOS EN REPORTE DE NOVEDADES	79
TABLA 5.6. PROBLEMAS, CANTIDADES Y PORCENTAJES RELACIONADOS AL GCS VENTURE VENTI (ENE-SEP 2011)...	79
TABLA 5.7. MEJORAS RELACIONADAS AL PROCESO DE GCS VENTURE VENTI (ENE-SEP 2011).....	80

LISTADO DE FIGURAS

FIGURA 2.1 VERSIONAMIENTO DE UN OBJETO	11
FIGURA 2.2. PROCESO DE UN CAMBIO GENÉRICO	13
FIGURA 2.3. PREGUNTAS PARA GENERAR INFORMES DE CAMBIOS.....	17
FIGURA 2.4. PREGUNTAS A RESPONDER DURANTE UNA AUDITORÍA.....	18
FIGURA 2.5. ESTRUCTURA MÉTRICA V3	20
FIGURA 2.6. RELACIÓN DE LA GC EN EL EVS	21
FIGURA 2.7. RELACIÓN ENTRE PROCESOS Y ACTIVIDADES DE GC EN MÉTRICA V3.....	22
FIGURA 2.8. GRUPOS DE PROCESOS DEL CICLO DE VIDA DEL SOFTWARE	24
FIGURA 2.9. PROCESOS DE IDENTIFICACIÓN DE LA CONFIGURACIÓN.....	28
FIGURA 2.10. DISCIPLINAS, FASES, ITERACIONES DE RUP.....	31
FIGURA 2.11. FLUJO DE TRABAJO DE LA GESTIÓN Y CONFIGURACIÓN DE CAMBIOS.....	34
FIGURA 3.1. ESTRUCTURA ORGANIZACIONAL DE LA EMPRESA VENTURE VENTI	38
FIGURA 3.2. PROCESO GESTIÓN DE CAMBIOS POR NUEVO REQUERIMIENTO DE UN CLIENTE VENTURE VENTI	41
FIGURA 3.3. PROCESO GESTIÓN DE CAMBIOS POR CORRECCIÓN DE ERRORES VENTURE VENTI	42
FIGURA 3.4. PROCESO GESTIÓN DE CAMBIOS POR IMPLEMENTACIÓN DE NUEVOS PROCESOS VENTURE VENTI.....	44
FIGURA 3.5. DEFICIENCIAS DE PERSONAL EN GCS VENTURE VENTI (ABR-DIC 2010)	48
FIGURA 5.1. ESTRUCTURA BIBLIOTECA DE SOPORTE VENTURE VENTI	60
FIGURA 5.2. ESTRUCTURA BIBLIOTECA DE SOPORTE AL PROYECTO VENTURE VENTI.....	61
FIGURA 5.3. ESTRUCTURA BIBLIOTECA MAESTRA VENTURE VENTI	61
FIGURA 5.4. MANEJO DE DEFECTOS DEL DESARROLLADOR	64
FIGURA 5.5. FLUJO DE TRABAJO PRUEBAS GESTOR DE LA CONFIGURACIÓN Y DESARROLLADOR DEL SOFTWARE....	65
FIGURA 5.6. DISEÑO VERSIONAMIENTO IMPLEMENTADO VENTURE VENTI.....	67
FIGURA 5.7. CREACIÓN REPOSITORIOS EN VISUALSVN	68
FIGURA 5.8. IMPORTAR PROYECTO EN VISUAL SVN	68
FIGURA 5.9. CHECKOUT PROYECTO DESDE SERVIDOR VISUAL SVN	69
FIGURA 5.10. VERSIONAMIENTO IMPLEMENTADO EN NETBEANS	69
FIGURA 5.11. ESTRUCTURA PARA VERSIONAMIENTO POWERBUILDER CLIENTE-SERVIDOR.....	71
FIGURA 5.12. CONFIGURACIÓN DE PROYECTO PARA VERSIONAMIENTO POWER BUILDER.....	71
FIGURA 5.13 AGREGAR CONTROL DE VERSIONES A UN PROYECTO DE POWER BUILDER	72
FIGURA 5.14. CONTROL DE CAMBIOS PROCESO GCS.....	76
FIGURA 5.15. DEFICIENCIAS DE PERSONAL EN GCS VENTURE VENTI (ENE-SEP 2011).....	80

GLOSARIO

Tester: Persona encargada de realización de pruebas del software

Release: Versión de un código lista para ser implementada

Dropdown: Menú de texto desplegable

Check in: Liberar un objeto luego de haber sido modificado

Check out: Bloquear un objeto para que pueda ser modificado

NOMENCLATURA UTILIZADA

IC:	Items de la Configuración
OCI:	Orden de cambio de Ingeniería
RUP:	Rational Unified Process (Proceso Racional Unificado)
ECS:	Elemento de Configuración del Software
RTF:	Revisión Técnica Formal
GCS:	Gestión de la Configuración del Software
ISO:	International Standardization Organization (Organización Internacional de Estandarización)
IEC:	International Electrotechnical Commission (Comisión Electrotécnica Internacional)
IEEE:	Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)
IDE:	Integrated Development Environment (Entorno de Desarrollo Integrado)

RESUMEN

El presente trabajo tiene como objetivo realizar el análisis, diseño, e implementación, relacionado al proceso de Gestión de la Configuración del Software para una empresa de desarrollo, en el que inicialmente se habla de conceptos y definiciones utilizados en el proyecto; para así ayudar al lector a tener una idea general y clara sobre el proceso de GCS.

Posteriormente con la descripción de la situación actual de la empresa y los problemas que posee durante el desarrollo del software, se realizó un análisis comparativo de las fases de modelos, o estándares que dentro de sus procesos involucren la GCS; buscando así identificar las fases que mejor se adapten a las necesidades de la empresa y puedan ayudar a solucionar sus molestias; en especial el versionamiento y procesos pruebas que son el mayor problema.

A través de la evaluación y selección de modelos o estándares que se adaptaron en sus diferentes fases de mejor manera a los requerimientos de la empresa, se desarrolló e implementó el proceso de GCS.

Finalmente luego de la implementación de la GCS, se logró dar solución a los problemas relacionados con el desarrollo del software en la organización, generando códigos con fallas reducidas en un 36.36% y actualizados, organización en las bibliotecas de trabajo y pruebas que cumplen satisfactoriamente sus objetivos; aportando así con procesos bien definidos y solventes, mejoras notables a la empresa auspiciante y mayor satisfacción a los clientes con el producto software que se les entrega.

CAPÍTULO I: GENERALIDADES

1.1. INTRODUCCIÓN

En todo desarrollo de una aplicación, el cambio va a ser un factor que se lo encuentra presente durante todo el ciclo de vida de la aplicación. En la mayoría de las ocasiones el desarrollo se lo realiza en grupos, por tal motivo es muy importante llevar un control y registro de los cambios efectuados que sufre una aplicación con la finalidad de reducir errores, aumentar la calidad, productividad, así como evitar los inconvenientes y problemas que puedan causar una sincronización incorrecta y no controlada de dichos cambios al tener relación con otras tareas o elementos del sistema que hayan sido realizadas por otros miembros del equipo de un proyecto.

Para evitar inconvenientes y problemas, la Gestión de la Configuración de software tiene como objetivo principal mantener la integridad de los productos (saber exactamente lo que se ha entregado al cliente, estado, contenido de las líneas base y elementos de la configuración) que se obtengan durante el desarrollo de los sistemas de información, garantizando que no se realizarán cambios no controlados y que a su vez todas las personas del grupo de desarrollo dispondrán de una versión adecuada de los productos que manejan.

1.2. DESCRIPCIÓN DEL PROBLEMA

Actualmente la empresa auspiciante no cuenta con procesos bien definidos para llevar un control de los diferentes cambios que se realizan en el código y con las versiones que se tiene de cada uno de los diferentes clientes que existen. Presentándose así quejas y malestar por parte de los usuarios finales; ya que en la entrega de algún cambio o nuevo requerimiento solicitado existen problemas debido a que las personas encargadas del desarrollo no han actualizado el código de

trabajo con anterioridad o no han informado de cambios que puedan afectar a los diferentes procesos o actividades que forman parte del software del cliente.

La falta de un control adecuado en los cambios realizados, tiene como repercusión inestabilidad y generación de errores; ya que no se hacen las pruebas pertinentes del caso, simplemente la persona encargada del desarrollo de alguna modificación o proceso nuevo únicamente envía el código sin realizar un control adecuado del mismo, causando en el software de los clientes una información incorrecta; la misma que debe ser cubierta como garantía por parte del desarrollador, lo que implica un doble trabajo, inestabilidad del software con sus procesos y pérdida de dinero para la empresa en relación al costo-hora trabajador.

1.3. JUSTIFICACIÓN

Al tener inconvenientes en la calidad del software durante el desarrollo de éste, se ha buscado procedimientos, metodologías o guías para poder solventar el malestar tanto a los usuarios finales, como en las personas involucradas del desarrollo del mismo; aquí es donde la Gestión de la Configuración de Software toma gran importancia y sirve de ayuda a los ingenieros para solventar sus problemas mediante actividades y procedimientos puntuales que se deben seguir.

Según (Bruegge & Dutoit, 2004) la Gestión de la Configuración de Software “Es la disciplina de administrar y controlar los cambios en la evolución de los sistemas de software”; teniendo en cuenta los problemas que se dan en la organización, la implementación de la Gestión de la Configuración de Software que buscará ayudar a mejorar todas estas deficiencias mediante las diferentes actividades que ésta abarca; apoyando al control de cambios, control de versiones, coordinación entre el equipo de desarrollo; para asegurar el contenido del producto sea difundido y se encuentre siempre disponible.

1.4. OBJETIVOS

1.4.1. Objetivo General

Diseñar e Implementar el proceso de Gestión de la Configuración de Software en la empresa de Desarrollo Venture Venti.

1.4.2. Objetivos Específicos

- Identificar los ECS (Elementos de Configuración de Software).
- Diseñar e implementar el proceso de control de cambios.
- Diseñar e implementar procesos de pruebas para los módulos y reportes de las aplicaciones desarrolladas.
- Diseñar e implementar el proceso de control de versiones.
- Definir la mejor herramienta para realizar el control de versiones.
- Mejorar la productividad, y minimizar los errores en el proceso de desarrollo de software.

1.5. ALCANCE

El proyecto busca solucionar los problemas que la empresa auspiciante tiene actualmente mediante la implementación del proceso de Gestión de la Configuración del Software; con la cual se tiene como objetivo llegar a diseñar procesos bien definidos para el control de cambios, control versiones, procesos de pruebas de las aplicaciones que realiza la empresa y con esto mejorar la calidad del software.

Durante el desarrollo del proceso se realizará primeramente un análisis de la situación actual de la empresa, para tener una visión clara de cómo se encuentra trabajando la empresa con actividades relacionadas al proceso de Gestión de la

Configuración de software; se realizará una valoración de procesos de GCS propuestos en metodologías o normas relacionadas y se elegirán las fases que mejor se acoplen a las necesidades de la empresa. Posteriormente se procederá con el diseño del proceso de GCS para la empresa, con la identificación de los diferentes elementos de configuración de software, control de versiones, casos de pruebas, control de cambios, informes de estado; y con esto se podrá establecer a futuro un proyecto para realizar una auditoría de la Gestión de la Configuración.

CAPÍTULO II: MARCO TEÓRICO

2.1. GESTIÓN DE LA CONFIGURACIÓN DE SOFTWARE

“La Gestión de la Configuración del Software (GCS) es una actividad de autoprotección que se aplica durante el proceso del software” (Pressman, 2002). La necesidad de aplicar procesos y actividades durante el desarrollo del software, es la consecuencia de problemas de calidad en los productos software en sus diferentes etapas, dificultades para el mantenimiento y mejoramiento del mismo por parte del equipo de desarrollo que en ocasiones el personal con más experiencia y años en una empresa se va convirtiendo en *insustituible*, por la falta de documentación estas personas son las únicas que pueden desarrollar por completo la realización de cambios en productos entregados, entendimiento de procesos del sistema, entre otras durante etapas avanzadas de desarrollo o implementación.

La GCS enfoca sus prácticas con la finalidad de ayudar a identificar donde un proceso está trabajando y necesita ser mejorado. Como una disciplina de ingeniería según (Keyes, 2004), la GCS provee niveles de soporte, control y servicio en la organización:

- **Soporte:** Entrega soporte a los ingenieros y desarrolladores del sistema, a la organización y en algunas situaciones a los clientes.
- **Control:** Controla especificaciones, documentación, borradores, requerimientos, herramientas, software y otros elementos relacionados.
- **Servicio:** La GCS provee servicios de soporte a las personas y al control de datos. El rol del administrador de GCS es asegurarse que el personal de la GCS está entrenado adecuadamente, posea los

recursos necesarios para realizar un trabajo eficiente y efectivo. Además se brinda un equilibrio adecuado de control y apoyo a la medida de que cada programa.

2.1.1. GCS y la mejora de procesos.

El mejoramiento depende de los procesos que van cambiando continuamente a través del entorno que a estos los acompaña.

El primer paso para mejorar un producto es saber cómo está actualmente siendo producido. El segundo paso es fomentar una atmósfera de trabajo en la cual los cambios puedan ser rápidamente adaptados. Si algún cambio no es posible realizar, entonces el mejoramiento tampoco se alcanzará. Si se pone en práctica estos aspectos se logrará aumentar la productividad, integridad, conformidad y la satisfacción del cliente (Keyes, 2004).

2.1.2. Mediciones y Métricas.

El tomar un aspecto medible en la GCS ayuda a tener una gestión que puede ser visible en los productos software. La medición de la información puede mostrar la ubicación de cuellos de botella en el proceso de desarrollo de software y puede también indicar la madurez de los productos software.

Una medida puede ser definida como “un estándar de medida, el alcance, dimensión, capacidad, de cualquier cosa, especialmente determinado por un estándar, un acto o proceso de medición, como un resultado de una medición” (Starrett, 1998). Ejemplos de medidas incluyen el número de defectos encontrados en una versión o el número de líneas de código fuente realizadas. Una métrica puede ser definida como “un indicador calculado basado en dos o más medidas, o una medida cuantificada en un grado en el

cual un sistema, componente, o proceso posee un atributo dado. Un ejemplo de métrica son los defectos por miles de líneas de código” (Starrett, 1998).

2.1.3. Beneficios de la Gestión de la Configuración del Software

El proceso de la Gestión de la Configuración del software tiene varios beneficios en la organización en la que se lo practique. Desarrolladores, “testers”, jefes de proyecto, personal encargado de la calidad del sistema y los clientes pueden obtener beneficios del proceso de la Gestión de la Configuración del software; según (Keyes, 2004) se encuentran las siguientes:

- Provee la habilidad de dar seguimiento a los cambios durante el desarrollo, sea este secuencial o en paralelo.
- Organiza las tareas y actividades que mantienen la integridad del software.
- Provee la habilidad para dar seguimiento desde el requerimiento hasta la obtención del producto.
- Asegura la configuración correcta del software.
- Asegura que los ingenieros implementen cambios correctamente en la línea base o en la versión del software.
- Ayuda a reducir el costo de la mantención del ciclo de vida del software, que puede fácilmente exceder el costo inicial en el desarrollo.
- Provee información para reportes que pueden ser fácilmente generados.
- Permite realizar auditorías rápidas y fáciles.
- Ayuda en la producción de un software de mayor calidad.

2.1.4. Actividades de GCS

La GCS como se ha mencionado es un elemento importante que ayuda a mejorar la calidad del software durante su desarrollo, donde su proceso se lo pueden definir en cinco tareas:

- Identificación de la Configuración.
- Control de Versiones.
- Control de Cambios.
- Informes de estado de la Configuración.
- Auditoría de la Configuración.

2.2. IDENTIFICACIÓN DE LA CONFIGURACIÓN

La identificación de la Configuración abarca la estructura del software del sistema y los componentes individuales para hacerlos accesibles de alguna forma. La meta de la identificación de la configuración es tener la habilidad de poder identificar los componentes de un sistema a través de su ciclo de vida y dar un seguimiento entre el software y los productos que estén relacionados. Las actividades que se incluyen en la Identificación de la Configuración son (Keyes, 2004):

- Selección de los ítems que pueden ser ubicados dentro del control de la GCS.
- Desarrollo de software jerárquico.
- Identificación y creación de un esquema el mismo que refleje la jerarquía del software.
- Identificación de que versión de un componente puede o no ser incluido en una versión entregada.
- Establecer líneas base de configuración.

Aunque los elementos más importantes que se toman en cuenta al realizar el proceso de GCS son los requerimientos y el código fuente, es importante tener en cuenta para el seguimiento a la información del entorno que le rodea al software y las herramientas de soporte a través del ciclo de vida; para así asegurarse que el software pueda ser puesto en producción sin inconvenientes. Es decir los ítems que típicamente se ponen bajo control de GCS son (Keyes, 2004):

- Módulos de código fuente.
- Archivos de datos del sistema.
- Archivos del sistema compilado y scripts.
- Especificaciones de requerimientos.
- Especificaciones de interfaces.
- Especificaciones de diseño.
- Especificaciones de la arquitectura del sistema.
- Planes de pruebas.
- Procedimientos de pruebas.
- Resultados de pruebas.
- Documentación de usuario.
- Plan de desarrollo de software.
- Planes de Calidad.
- Planes de Gestión de la Configuración.

- Sistemas operativos.
- Herramientas de terceros.
- Otras herramientas de trabajo.
- Procedimientos y estándares de desarrollo.

2.3. CONTROL DE VERSIONES

El control de versiones es la combinación de procesos y herramientas que ayudarán con la gestión de versiones de objetos de configuración creados durante el proceso de ingeniería de software (Pressman, 2002). La versión es una forma particular que adopta un elemento de configuración en un instante determinado como se muestra en la Figura 2.1.

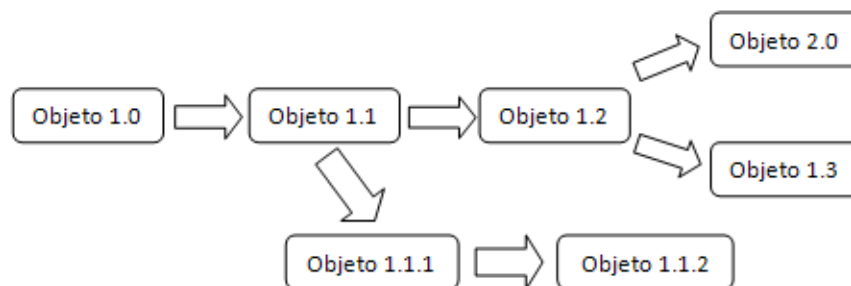


FIGURA 2.1 Versionamiento de un Objeto

Durante el proceso de control de versionamiento se deberá tener en cuenta cierta terminología que se puede emplear en su funcionamiento:

Versión: Instancia de un objeto que es diferente en algo significativo a otras instancias.

Revisión: Modificación secundaria.

Variante: Versiones que coexisten.

Release: Versión que se distribuye a los clientes.

2.4. CONTROL DE CAMBIOS

El control de cambios de la configuración del software comprende el control del lanzamiento de una versión y los cambios de los productos software a través del ciclo de vida del software. Es sin duda el elemento más visible de la Gestión de la Configuración; este es el proceso para gestionar la preparación, justificación, evaluación, coordinación, disposición, e implementación de cambios propuestos de ingeniería y las desviaciones que afectarán los ítems de la configuración y la documentación basada en las líneas base.

El objetivo del control de cambios de la configuración es establecer mecanismos que ayudarán a asegurar que existirá calidad en la producción del software, como también garantizar que cada versión del software contenga los elementos necesarios y que todos los elementos en una versión trabajen correctamente en conjunto.

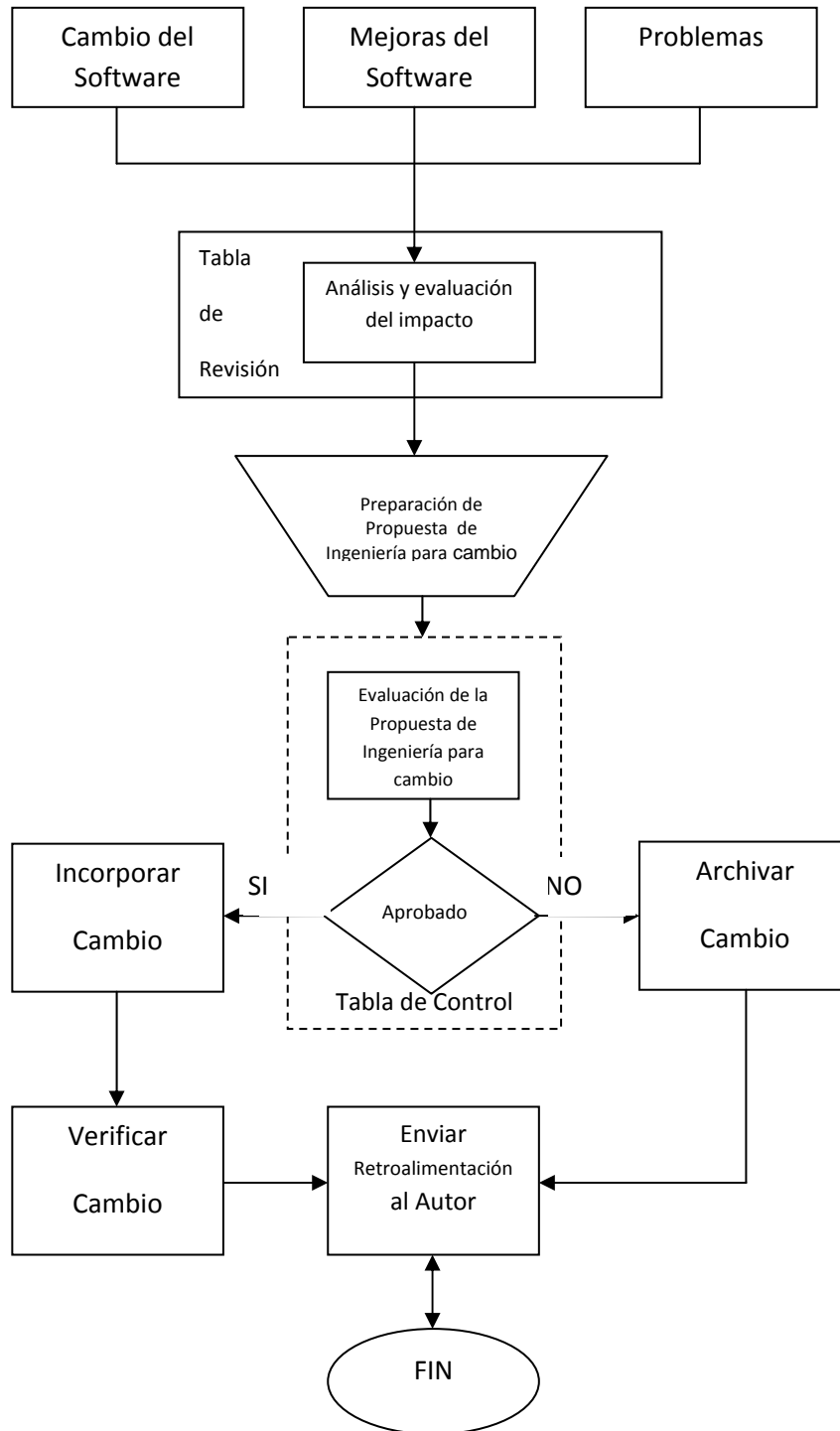


Figura 2.2. Proceso de un cambio genérico (Berlack, 1992)

Las actividades que se incluyen en el control de la configuración de cambios son (Keyes, 2004):

- Definición de procesos de cambios.
- Mantenimiento de líneas base.
- Establecimiento de políticas y procedimientos de control de cambios.
- Procesamiento de cambios.
- Desarrollo de formato de reportes de cambios.
- Control del lanzamiento de un producto.

Los cambios que han sido realizados en las líneas base o ítems basados en éstas, deben ser acorde a lo documentado en el proceso de control. El proceso de control debe especificar (Keyes, 2004):

- La persona que puede iniciar el proceso de requerimientos.
- El criterio para colocar los componentes del software bajo un control formal de cambios.
- El análisis del impacto del cambio esperado para cada petición de cambio.
- Procedimientos para registrar y quitar un cambio.
- Revisiones y pruebas que deben ser realizadas para asegurar que los cambios no hayan causado resultados inesperados en las líneas base.

2.5. PRUEBAS

La realización de pruebas es el resultado de algunos aspectos durante el desarrollo de software: uno es asegurar que algo realizado o que se está realizando funcione correctamente antes de ser puesto en producción, por otro lado los fallos en el funcionamiento del software que se encuentran durante producción, hacen considerarlo deficiente, por ejemplo:

- Fallos del software constantes.
- Consecuencias graves debido a fallos del sistema que sean inaceptables, desde escenarios financieros o que amenacen a la vida de un ser humano.
- Mejoras en sistemas que generalmente son costosos.
- Los costos en la detección y eliminación de defectos son excesivos.

No siempre los fallos se deben al mal desarrollo de un proceso, sino también al incorrecto levantamiento de algún requerimiento donde los fallos serán más notorios y frecuentes. Por todo esto, las pruebas son muy importantes ayudando así a la reducción de fallos en el software.

2.5.1. Pruebas de Caja Negra (Funcionales)

Este tipo de pruebas están basadas en la funcionalidad del programa o del sistema, donde los tester no necesitan saber cómo funciona internamente un método o proceso del sistema, sino simplemente la información de entrada y observar cuál es el resultado de haber insertado una u otra información. El tester únicamente se enfoca en probar la funcionalidad del programa con las especificaciones que se le den (Everett & Raymond, 2007).

La ventaja más grande de este tipo de pruebas es que el tester está orientado a probar lo que el sistema o programa debe hacer, lo cual es

comprensible por cualquiera. Una limitación muy grave es la entrada de datos en cantidades muy extensas que no es alcanzable, debido a que se tendría que probar todas las entradas de datos posibles.

2.5.2. Pruebas de Caja Blanca (Estructurales)

En las pruebas de caja blanca o estructuradas, las condiciones de pruebas están diseñadas basadas en la lógica de los 'caminos'. El probar los datos, está manejado mediante la revisión de la lógica del sistema o del programa, sin tomar en cuenta los requerimientos del mismo. En este caso el tester deberá conocer la estructura interna y lógica del programa y con esto existirá una probabilidad mucho más alta de que se encuentren errores o daños realizados en el código por un programador.

Como se mencionó las pruebas de caja blanca no se encargan de la verificación de que las especificaciones hayan sido correctas, siendo esta una desventaja, ya que solo se preocupa de la lógica interna, más no esta lógica con las especificaciones. Otra desventaja de este tipo de pruebas es que no es posible probar todos los caminos lógicos posibles, por lo tanto esto causaría cantidades exuberantes de pruebas.

2.5.3. Pruebas de Caja Gris (Estructural y Funcional)

Las pruebas de caja gris son la combinación de pruebas de caja negra y blanca, es decir que se busca un equilibrio entre la funcionalidad contra las especificaciones; aquí el tester deberá tener una comunicación con el desarrollador para que pueda entender completamente la estructura de un programa o aplicación, y con trabajo en conjunto se pueda facilitar la revisión de la aplicación (Lewis, 2009). Un ejemplo puede ser cuando el tester

encuentra cierta funcionalidad que parece rehusarse en varias partes de la aplicación, si en este caso el tester se comunica con el desarrollador este le explicará el diseño interno y arquitectura del mismo, el tester podrá así comprender para de esta manera eliminar varios test pudiéndose probar la funcionalidad en una sola ocasión.

2.5.4. Pruebas Manuales versus Automatizadas

Las pruebas manuales son aquellas que no son implementadas en un computador y no ejecutadas por los usuarios, como por ejemplo pruebas de escritorio, inspecciones, revisiones estructuradas (Lewis, 2009).

Por otro lado las pruebas automatizadas son aquellas que se implementan en la computadora como pruebas de cobertura de ramas, pruebas de valores, pruebas de sintaxis.

2.6. GENERACIÓN DE INFORMES SOBRE CAMBIOS DE CONFIGURACIÓN

La generación de informes son actividades que consisten en la información del estado de los ECS y de las solicitudes de cambio. Estos informes ayudan a mejorar la comunicación entre las personas que se encuentran involucradas, ya que mantiene al tanto de los cambios que han sido efectuados o planificados.

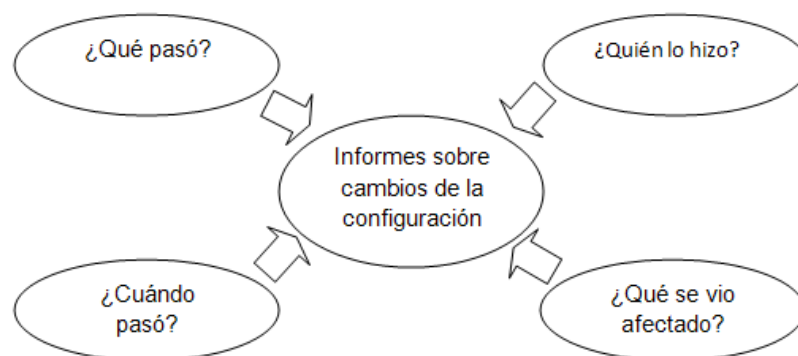


Figura 2.3. Preguntas para generar Informes de Cambios

2.7. AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE

Los cambios pueden ser asegurados y revisados mediante:

- Revisiones Técnicas Formales(RTF) y
- Auditorías de configuración de software.

Las revisiones técnicas formales están enfocadas en la corrección técnica del elemento de configuración que ha sido modificado. Las revisiones evalúan el ECS para determinar la consistencia con otros ECS, las omisiones, o posibles efectos secundarios.

La auditoría de la configuración viene a ser como un estado final de la configuración, que complementa a las RTF, contando con aspectos no considerados por las mismas; en la cual se revisará y controlará que el esfuerzo empleado ha sido alcanzado satisfactoriamente con todos los requerimientos establecidos en las líneas base para los EC y el sistema (Pressman, 2002).

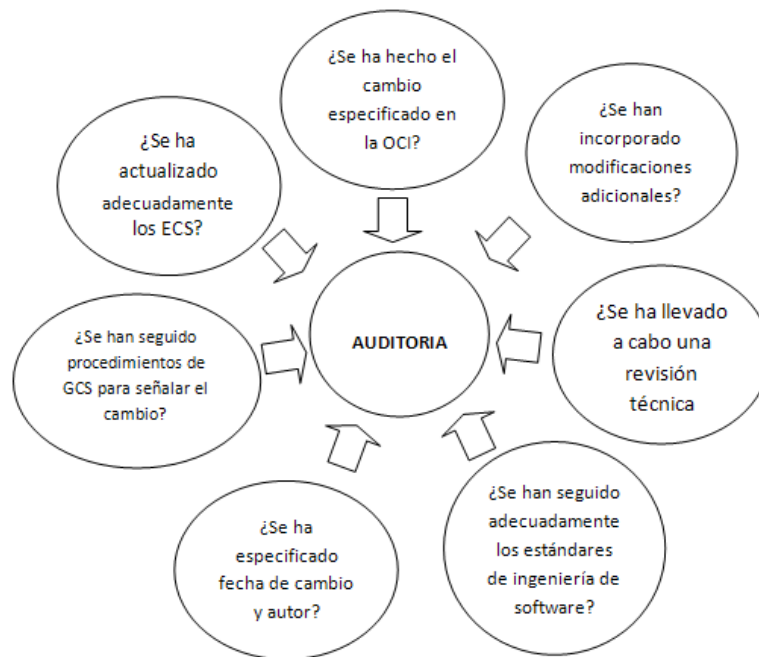


Figura 2.4. Preguntas a responder durante una auditoría

De existir algún tipo de problema es responsabilidad de las actividades de la auditoría el asegurar que todos los elementos hayan sido identificados, dirigidos y cerrados, antes de que una actividad pueda ser considerada como cumplida según sus requisitos.

2.7.1. Auditoría de Configuración Funcional

La auditoría de configuración funcional es usada para verificar el desempeño real de los ECS con los requerimientos que se les ha establecido y certificar que hayan sido cumplidos.

2.7.2. Auditoría de Configuración Física

La auditoría de la configuración física está encargada de verificar la adecuación, integridad y precisión de los elementos físicos de documentación que constituyen la Línea Base.

2.8. PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE.

En la Ingeniería de Software se han desarrollado varios metodologías, estándares y propuestas sobre el proceso de GC, se han recogido varias propuestas que se presentan a continuación:

2.8.1. Métrica Versión 3

Métrica V3 es una metodología útil para la sistematización de actividades que permite definir cuáles son las necesidades de la Gestión de Configuración de los diferentes sistemas de información que puedan existir, los mismos que son recogidos en un plan donde se especifican las actividades de identificación y registro de productos en el sistema de gestión durante el desarrollo y posterior mantenimiento del sistema de información (Ministerio de Administraciones Públicas de España).

La Gestión de la Configuración viene dada en Métrica Versión 3 como una interfaz mostrada en la Figura 2.5, que consiste en la aplicación de procedimientos tanto administrativos como técnicos durante el desarrollo, y posterior mantenimiento del sistema de información. Tiene como finalidad identificar, definir, proporcionar información y controlar los cambios en la configuración del sistema, así como las modificaciones y versiones de los mismos.

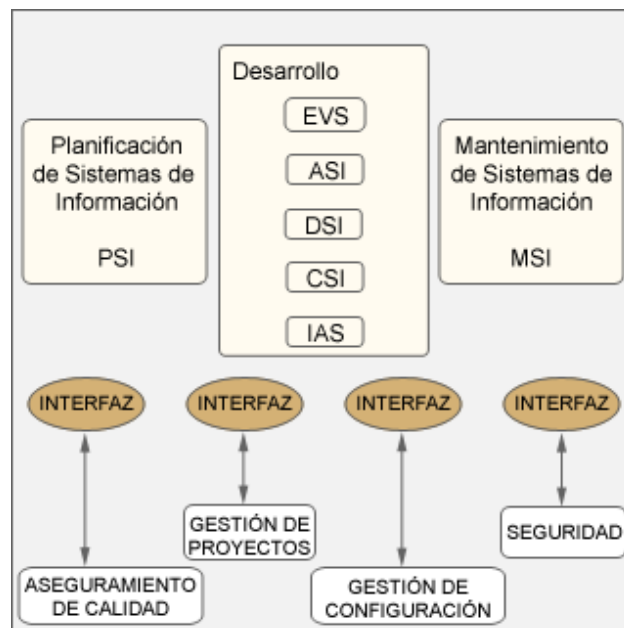


Figura 2.5. Estructura Métrica V3 (Ministerio de Administraciones Públicas de España)

La aplicación de la Gestión de la Configuración en Métrica V3, se dará dentro de los diferentes procesos que conforman parte de la misma según el (Ministerio de Administraciones Públicas de España):

a) Estudio de Viabilidad del sistema(EVS)

Durante el EVS se realizan actividades de GC que permiten obtener el Plan de Gestión de la Configuración para el sistema de información. Con este objetivo se definen en primer término los requisitos de Gestión de la Configuración del sistema de información, los cuales deberán tenerse en

cuenta a la hora de establecer el plan de Gestión de la Configuración para la Solución propuesta (EVS 6.2) como se muestra en la Figura 2.6:

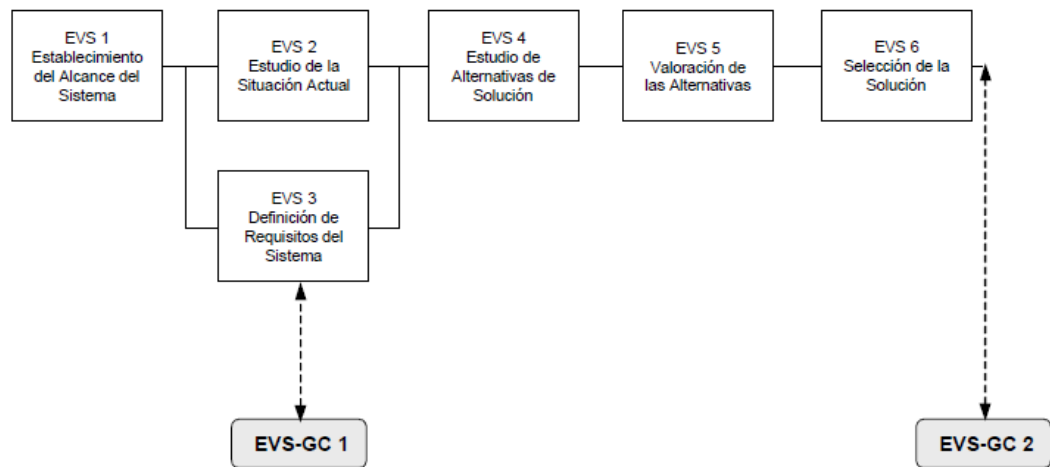


Figura 2.6. Relación de la GC en el EVS (Ministerio de Administraciones Públicas de España)

Actividad EVS – GC1: Definición de los Requisitos de Gestión de la Configuración.

El objetivo es definir los requisitos generales de Gestión de la Configuración y determinar los procesos de control que se ejecutarán para mantener íntegros los productos obtenidos a lo largo de la utilización de Métrica V3.

Actividad EVS – GC2: Establecimiento del Plan de Gestión de la Configuración.

Esta actividad define el Plan de Gestión de la Configuración para la solución propuesta (EVS 6.2) y especificar el entorno tecnológico de soporte para la Gestión de la Configuración.

Una vez establecido el Plan de Gestión de la Configuración se irán registrando los productos que se obtengan en los procesos de Análisis, Diseño, Construcción, Implantación y Aceptación del Sistema de Información y que se hayan determinado en el plan como los productos a incluir en el sistema de Gestión de la Configuración.

b) Análisis, Diseño, Construcción e Implantación y aceptación del sistema de información.

Durante los proceso de Análisis, Diseño, Construcción e Implantación del Sistema de Información se realizan las actividades de identificación y registro previstas en el Plan de Gestión de la Configuración , consiguiendo así mantener la consistencia entre las distintas versiones de los productos de desarrollo. La Figura 2.7 muestra la relación que existe entre los procesos de desarrollo de Métrica V3 y las actividades de la interfaz de Gestión de la Configuración.

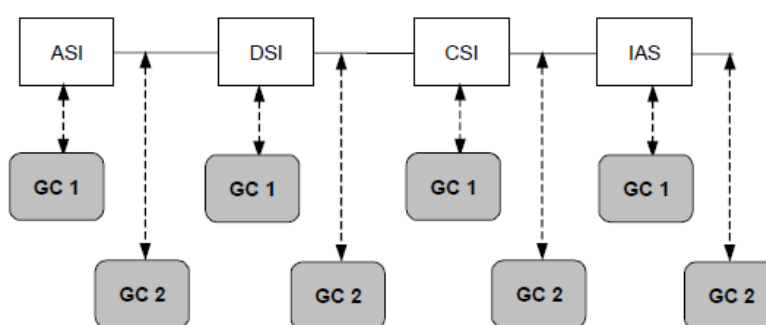


Figura 2.7. Relación entre procesos y actividades de GC en Métrica V3

(Ministerio de Administraciones Públicas de España)

Actividad GC 1: Identificación y Registro de Productos

En esta actividad se identificarán los productos que se obtienen de cada uno de los procesos, dándoles a estos un nombre, código de versión, un estado que indicará como éste se encuentra dentro de su proceso de elaboración, y su localización en el sistema de Gestión de la Configuración para que sean únicos.

Actividad GC 2: Identificación y registro del producto Global

El objeto de la actividad es el de identificar y registrar en el sistema, los productos globales que se obtienen a lo largo del desarrollo de los procesos principales dentro del sistema de Gestión de la Configuración.

c) Mantenimiento del sistema de Información

La Gestión de la Configuración tiene como finalidad en el mantenimiento del sistema de información, el conservar la integridad cuando se producen cambios en el mismo, sean éstos dados por mantenimiento correctivo o evolutivo.

El beneficio de realizar una buena Gestión de la Configuración durante el proceso de mantenimiento es muy elevado, teniendo en cuenta la reducción del tiempo de localización de los problemas, la reproducción de errores y el control de seguimiento de los estados por los que va pasando la petición de mantenimiento. De esta manera se puede conocer en cada momento la situación en la que se encuentra cada cambio en particular y el sistema de información general.

Actividad MSI-GC 1 – Registro del cambio en el sistema de Gestión de la Configuración.

En esta actividad se debe realizar las tareas que se resumen en la Tabla 2.1:

Tabla 2.1. Tareas del Registro del cambio en el sistema de Gestión de la Configuración Métrica V3 (Ministerio de Administraciones Públicas de España)

Tarea		Productos	Técnicas y Prácticas	Participantes
MSI-GC 1.1	Registro del cambio en el Sistema de Gestión de la Configuración	- Registro del cambio		- Técnico de mantenimiento
MSI-GC 1.2	Registro de la nueva versión de los productos afectados por el cambio en el sistema de Gestión de la Configuración	- Registro de la nueva versión de los productos afectados por el cambio		- Técnico de mantenimiento
MSI-GC 1.3	Registro de la nueva versión de los sistemas de información en el sistema de Gestión de la Configuración	- Registro de la nueva versión de los sistemas de información		- Técnico de mantenimiento

2.8.2. ISO/IEC 12207: Proceso del ciclo de Vida del Software

ISO/IEC 12207 es un estándar internacional (Comisión de Reglamentos Técnicos y Comerciales-INDECOPI, 2006) que muestra el proceso del ciclo de vida del software, actividades y tareas para el software que son parte de un gran sistema. Dentro de todo este proceso se encuentra la Gestión de la Configuración como un proceso de apoyo del ciclo de vida del software como muestra la Figura 2.8.

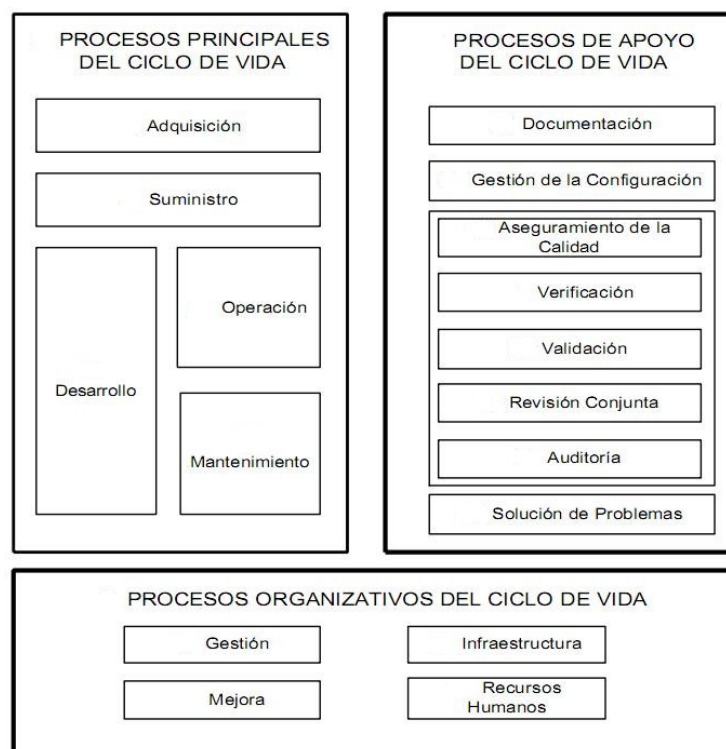


Figura 2.8. Grupos de procesos del Ciclo de vida del Software (Comisión de Reglamentos Técnicos y Comerciales-INDECOPI, 2006)

Dentro de las actividades de Gestión de la Configuración se encuentran:

a) Implementación del proceso

Se preparará un plan de Gestión de la Configuración en el cuál se describirá las actividades, procedimientos y plazos para estas. Se definirá

también la organización responsable de ejecutar las actividades y las relaciones que tengan con otras organizaciones (desarrollo o mantenimiento de software).

b) Identificación de la configuración

Se establecerá un esquema que identificará los elementos software que van a ser sujetos a control en el proyecto. Se generará documentación establecida en la línea base de referencia, como las versiones y otros detalles para cada uno de los elementos identificados.

c) Control de la configuración

Esta actividad es la encargada de la identificación y registro de las peticiones, análisis y evaluación de los cambios; aprobación o rechazo de la petición e implementación, verificación y lanzamiento del elemento software que ha sido modificado. Para los cambios realizados deberá existir algún rastro auditable, el cual permita el posterior control de dónde, quién realizó y aprobó alguna modificación.

d) Determinación del estado de la configuración

Se realizarán informes del estado que indicarán la historia y estado de los elementos software, controlando donde se debe incluir las líneas de referencia. Estos informes contendrán el número de cambios de un proyecto, últimas versiones de los elementos software, identificación, número y comparación entre los lanzamientos.

e) Evaluación de la configuración

La finalidad de esta actividad es asegurar la completitud funcional de los elementos software con los requerimientos realizados y completitud física (en caso de ser necesario)

f) Gestión de lanzamientos y entrega

Esta actividad busca realizar un control formal de los lanzamientos y entregas de productos software, como la documentación que debe presentarse. Se deberá guardar copias tanto del código, como documentos generados durante todo el ciclo de vida del producto software; resaltando la importancia de mantener políticas bien establecidas de seguridad, acceso para documentación y código que contenga funciones críticas.

2.8.3. IEEE 828-1998 – Estándar para elaboración de planes de Gestión de la Configuración del Software

La IEEE 828-1998 (Software Engineering Standards Committee, 1998) es una de las más amplias guías usadas y efectivas para implementación de planes de Gestión de la Configuración del software. La estructura principal de este estándar se muestra en la Tabla 2.2.

Tabla 2.2. Estructura estándar IEEE 828-1998 (Software Engineering Standards Committee, 1998).

Proceso	Descripción
Introducción	Describe el propósito del Plan, alcance de la aplicación, términos claves, y referencias.
Dirección GCS	(Quién?) Identifica las responsabilidades y autoridades para alcanzar las actividades planeadas.
Actividades GCS	(Qué?) Identifica todas las actividades que van a ser realizadas en la aplicación del proyecto.
Calendarios GCS	(Cuándo?) Identifica la coordinación de actividades requeridas de la GCS con otras actividades en el proyecto.
Recursos GCS	(Cómo?) Identifica herramientas, recursos físicos y humanos requeridos para la realización del plan.
Plan de mantenimiento GCS	Identifica como el plan va a mantenerse en efecto.

La información de las actividades de la GCS identifica todas las funciones y tareas requeridas para administrar el sistema de la configuración del software que esté especificado en el alcance del plan. Ambas actividades tanto técnicas, como gerenciales deben ser identificadas. Actividades

generales del proyecto que tengan relación con la GCS deberán ser descritas desde una perspectiva de GCS. Las actividades de GCS son agrupadas de la siguiente manera según (Software Engineering Standards Committee, 1998):

a) Identificación de la Configuración

Estas actividades deberán identificar nombres para describir documentadamente las características físicas y funcionales del código, especificaciones, diseño como también datos de elementos que van a ser controlados por el proyecto. Los documentos son adquiridos para control de la configuración. Los ítems controlados pueden ser salidas parciales o finales (como código ejecutable, código fuente, documentación de usuario, listas de programas, base de datos, casos de pruebas, planes de pruebas, especificaciones y planes de control) y elementos de soporte del área de trabajo (como compiladores, sistemas operativos, herramientas de programación).

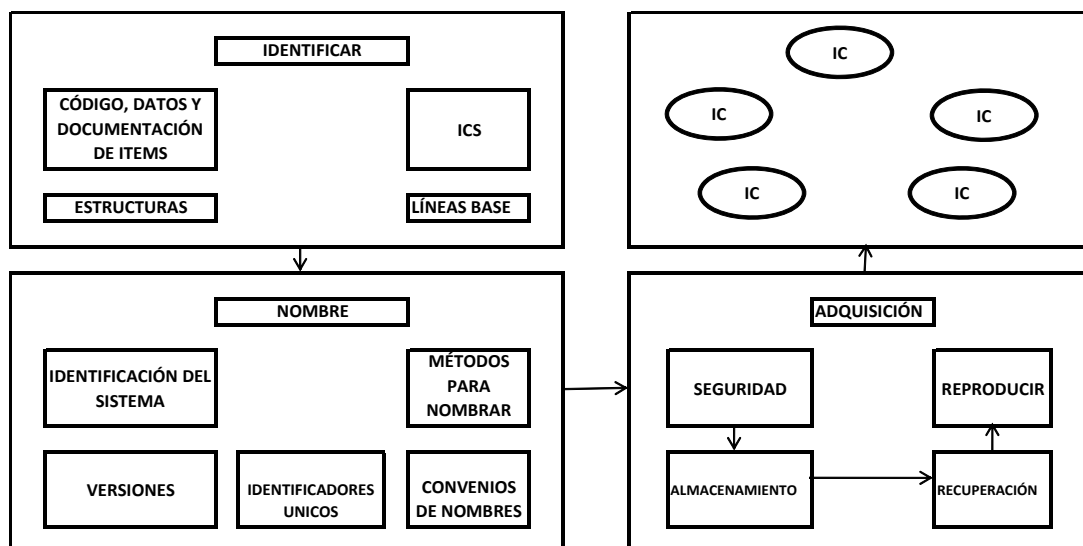


Figura 2.9. Procesos de Identificación de la Configuración (Software Engineering Standards Committee, 1998)

b) Control de la configuración

Las actividades de control de la configuración como petición, evaluación, aprobación, desaprobación e implementación de cambios, son realizadas a los ICs definidos en las líneas base. El nivel de formalidad necesario para el proceso de cambio dependerá de la línea base del proyecto afectada en el impacto del cambio relacionado con la estructura de la configuración.

c) Determinación del estado de la configuración

La determinación del estado de la configuración estará basada en la recolección de información y reportes de los ICs del proyecto. El plan debe incluir la siguiente información:

- Elementos que van a ser revisados y reportados por las líneas base y cambios;
- Tipos de reportes de estado de la configuración van a ser generados y su frecuencia;
- Información que va a ser recogida, almacenada, procesada y reportada;
- Procedimiento de control de acceso a los estados de la información.

d) Configuración de auditorías y revisiones

Las auditorías de la configuración determinan en qué medida el IC actual refleja las características físicas y funcionales requeridas. Las revisiones de la configuración son herramientas administrativas para establecer líneas base.

Según la (Software Engineering Standards Committee, 1998) cada auditoría o revisión planeada deberá definir lo siguiente:

- Su objetivo.
- ICs que serán auditados o revisados.
- Calendario de la auditoría o las tareas de revisión.
- Procedimientos para la conducción de la auditoría o revisión.
- Participantes por título de trabajo.
- Documentación requerida para validar una revisión o dar soporte a una auditoría o revisión.
- Procedimiento para identificar alguna deficiencia y reportar acciones correctivas.
- Criterio de aprobación y acciones específicas que ocurrirán luego de la aprobación.

e) Control de Interfaz

Las actividades de control de interfaz coordinan los cambios a los ICs del proyecto con los ítems que se encuentran fuera del alcance del plan. Hardware, software de sistema, entregables y sistema de apoyo como otros proyectos deberán ser examinados para los potenciales efectos que se puedan dar en la interconexión en el proyecto.

f) Control de subcontratista y vendedor

Estas actividades de control son las encargadas de incorporar los ítems fuera del ambiente de trabajo dentro de los ICs del proyecto. Entre estos se deberá incluir el software desarrollado por contrato y software que se encuentre terminado.

2.8.4. Rational Unified Process(RUP)

Rational Unified Process (Proceso Unificado Racional) es un proceso de ingeniería de software que abarca la asignación de tareas y responsabilidades dentro de una organización de desarrollo (Kruchten, 2000). El objetivo de RUP es producir software de calidad que resuelva las necesidades y requerimientos de los usuarios cumpliendo tiempo y presupuesto establecido.

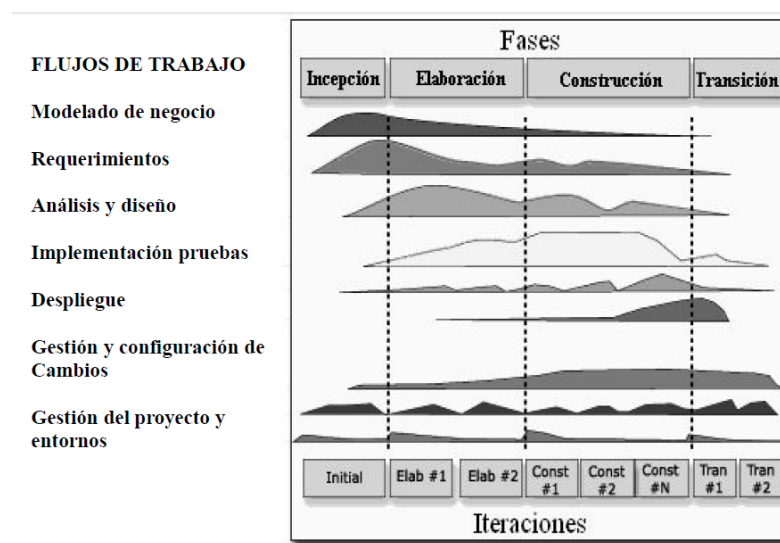


Figura 2.10. Disciplinas, fases, iteraciones de RUP (Rational the software development company, 1998)

RUP dentro de sus procesos incorpora a la Gestión y configuración de cambios mostrado en la Figura 2.10, la cual tiene como objetivo el controlar y mantener la integridad de los productos que incluyen un proyecto. Aquí se definen las siguientes actividades:

- Gestionar las solicitudes de Cambio.
- Plan de configuración del proyecto y control de cambios.
- Crear un entorno de Gestión de la Configuración del proyecto.

- Supervisión e informe del estado de la Configuración.
- Elementos de la configuración de entrega y cambios.
- Control de líneas base y lanzamientos.

a) Gestionar las solicitudes de Cambio

El propósito de un control de cambio documentado y estandarizado permite asegurar que los cambios realizados en un proyecto sean efectuados de una forma coherente y adecuada por las partes interesadas, las cuales son informadas del estado del producto, cambios a ejecutar, costo y cronograma.

b) Plan de configuración del proyecto y control de cambios

Se describe en el plan las actividades relacionadas con la GC y recursos, procedimientos, control de cambios; políticas que se aplicarán para identificar, proteger e informar, sobre todos los artefactos que se generan durante el ciclo de vida del proyecto y con los que se deben contar durante el transcurso del mismo.

c) Crear un entorno de Configuración del proyecto

Esta actividad tiene como objetivo el crear un ambiente de trabajo donde todos los artefactos de desarrollo estén disponibles, la ubicación en la cual el producto va a ser desarrollado, integrado y archivado para un posterior mantenimiento o re-uso. El entorno de trabajo provee a los desarrolladores y personal encargado de la integración, áreas de trabajo privadas y compartidas en las cuales ellos pueden integrar el software.

d) Supervisión y reportes del estado de la Configuración

Como custodio del repositorio del proyecto el director de la GC debe asegurarse que todos los artefactos sean los necesarios y estén seguros. El director de la GC será el responsable de reportar sobre el estado de la configuración, estos reportes pueden aportar con información valiosa para el seguimiento del progreso general del proyecto y las tendencias que se puedan dar.

e) Cambio y entrega de elementos de configuración

Se indicará la manera en la que cualquier trabajador puede crear un área de trabajo. Dentro del área de trabajo, un trabajador tendrá acceso a los artefactos del proyecto y podrá realizar cambios para que estos sean incluidos en el producto global. La realización y entrega de estos cambios, están hechos mediante la integración de varias áreas que incluyen el trabajo de múltiples empleados. La finalidad es integrar trabajos individuales para que sean mostrados al resto de desarrolladores para los posteriores proyectos.

f) Control de líneas base y lanzamientos

Una línea base es una descripción de todas las versiones de los artefactos que componen un producto en cualquier momento dado. Típicamente las líneas base son creadas al final de las iteraciones y en los hitos establecidos. Un producto deberá estar puesto en una línea base cada vez que sea entregado a un cliente.

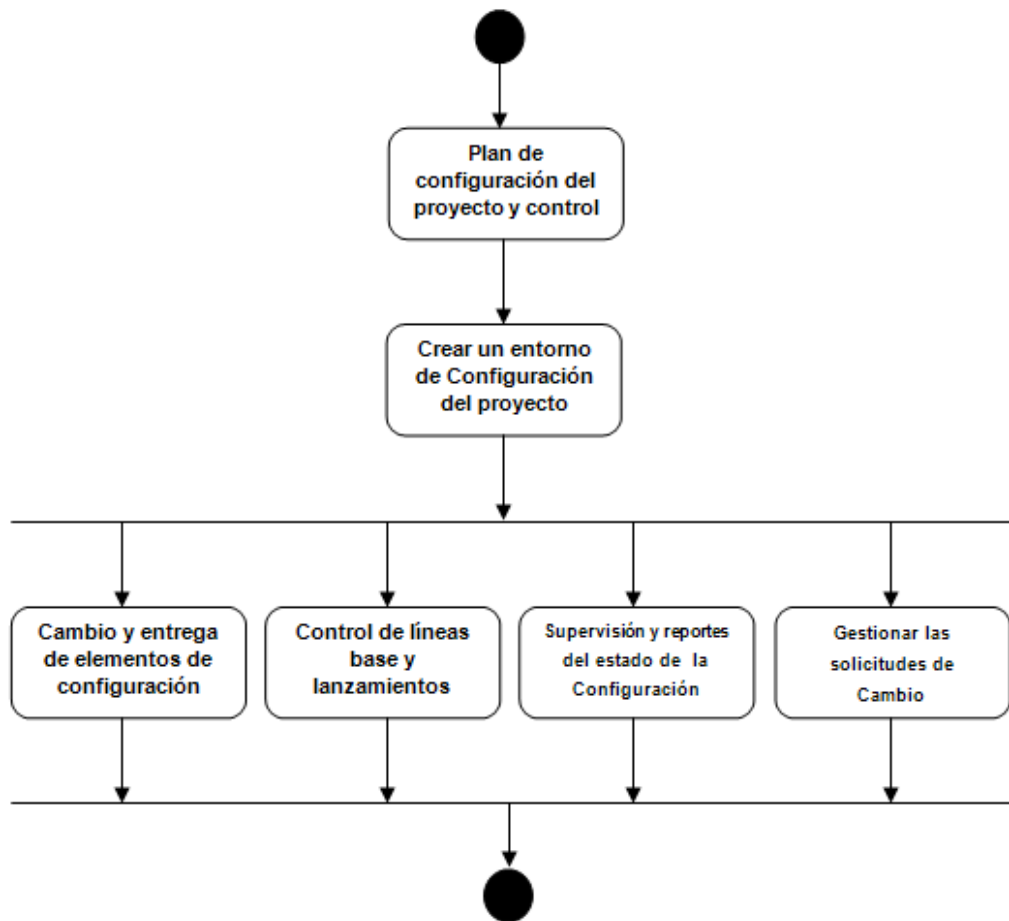


Figura 2.11. Flujo de Trabajo de la Gestión y Configuración de Cambios
(Kruchten, 2000)

2.9. VENTAJAS Y DESVENTAJAS DE ESTÁNDARES Y MODELOS RELACIONADAS AL PROCESO DE GCS

2.9.1. Métrica Versión 3

VENTAJAS:

- Permite acoplar sistemas de configuración previos a nuevos requerimientos, teniendo en cuenta las necesidades que no se encuentren recogidas en el sistema de configuración anterior y sean tomadas en cuenta nuevamente.
- Define roles específicos para la realización de procesos de GCS.
- Permite una mayor capacidad de adaptación a los cambios tomando en cuenta la reutilización de recursos.
- La automatización de actividades es posible ya que las técnicas empleadas están soportadas por una gran variedad de herramientas de ayuda al desarrollo disponibles en el mercado como Diagramas de Estructura, Diagramas de transacción de estado, casos de uso, Diagramas de clases, Análisis de Impacto.

DESVENTAJAS:

- Es un sistema demasiado pesado en sus procesos de mantenimiento, como en los de implementación.
- Proporciona un sin número de información y documentos, lo cual de vez en cuando puede causar confusión en vez de claridad.

2.9.2. ISO/IEC 12207

VENTAJAS:

- Se aplica en organizaciones en las cuales sus actividades incluyan planificación, gestión, control o mejora de los procedimientos de adquisición, suministro, desarrollo, operación, evolución y soporte de software.
- Reduce tiempos y costes de desarrollo.

DESVENTAJAS:

- No existe una visibilidad clara en los procesos de Gestión de la Configuración, lo que al final de los procesos puede llegar a generar situaciones no esperadas.
- La dimensión de la capacidad tiene un grado de dificultad muy alto y pueden existir solapamientos con la dimensión de procesos.

2.9.3. IEEE 828-1998

VENTAJAS

- La ventaja más favorable de este estándar es su flexibilidad además contiene instrucciones específicas que ayudan a evitar que el usuario elimine algún componente del plan sin previamente especificar porque lo desea eliminar.
- Es fácil de usar, lo que permitirá al usuario su entendimiento y poder distinguir entre aquellos ítems que son obligatorios que estén en el plan y los que son opcionales.
- Es un estándar completo debido que los componentes del plan están descritos en su adecuada profundidad, además de tratar sobre todos los componentes necesarios para poder realizar el plan.

DESVENTAJAS

- En la definición de términos hace referencia a otros estándares, en lugar de definirlos en el mismo documento.
- No posee ejemplos suficientes y no trata a detalle el tema de herramientas automatizadas.
- No involucra a todo el proceso de Gestión de la Configuración del Software, simplemente se enfoca en el plan.

2.9.4. Rational Unified Process (RUP)

VENTAJAS

- Al final de cada fase se presenta a los clientes los artefactos finales, para que sean evaluados por este y se puedan generar las siguientes iteraciones de la siguiente fase.
- Asigna tareas y responsabilidades de una forma disciplinada.
- Acelera el ritmo de esfuerzo de desarrollo ya que los desarrolladores trabajan para obtener resultados claros a corto plazo.

DESVENTAJAS

- Es un proceso pesado ya que está basado mucho en la documentación como por ejemplo Diagramas de Casos de uso, Diagramas de Secuencia, Diagramas de Actividades, entre otros; lo cual se puede ver afectada con cambios inesperados que se les puede ocurrir a los clientes.
- Tiene visión para proyectos y equipos grandes que posean roles definidos y con una duración extendida.

CAPÍTULO III: EVALUACIÓN DE LA SITUACIÓN ACTUAL DEL PROCESO DE GESTIÓN DE LA COFIGURACIÓN DE SOFTWARE Y PRUEBAS EN LA EMPRESA

3.1. INTRODUCCIÓN

Como en algunas organizaciones que desarrollan software, Venture Venti tiene problemas en los procesos que se relacionan con la Gestión de la Configuración. La mayoría de sus procesos, especialmente para control de versiones de código no están bien definidos, causando muchas molestias a los clientes, pérdida de dinero y tiempo en la empresa por rectificación de procesos mal realizados en el desarrollo de software.

Se deberá entender que el proceso de Gestión de la Configuración del Software en toda empresa es una actividad esencial ya que se define requerimientos, diseños, compilación, pruebas, documentación del software y no solo el control de versiones. La Figura 3.1 muestra la estructura organizacional de la empresa donde el Gerente de Tecnología es el encargado de tomar las decisiones principales de los cambios que se puedan necesitar en el software, delegando responsabilidades directamente a los Jefes de cada Departamento.

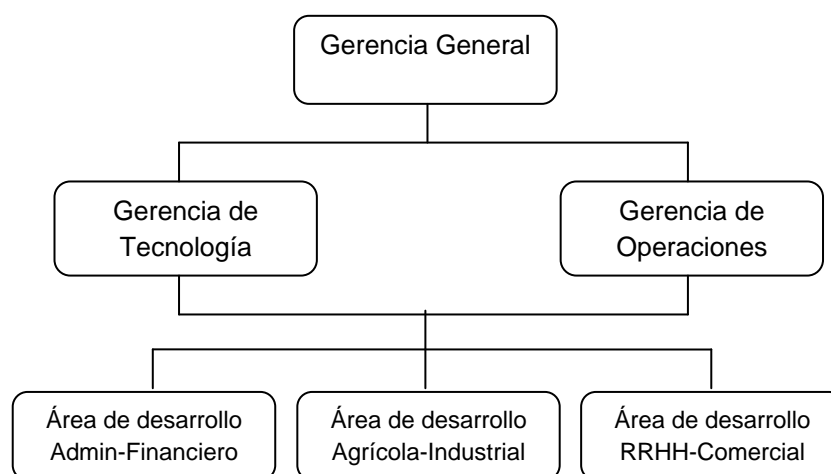


Figura 3.1. Estructura Organizacional de la Empresa Venture Venti

3.2. ASPECTOS TECNOLÓGICOS

El software que se utiliza en los equipos de la empresa es el siguiente:

3.2.1. Software Personal de Desarrollo

Tabla 3.1. Software Personal de Desarrollo Venture Venti

	Software
Sistema Operativo	Windows 2003 R2/ Windows 2008 R2
Desarrollo	Power Builder 11.5 Netbeans 6.5.1 Icefaces 1.8.1 Microsoft SQL 2003
Otros	Microsoft Office Venture ERP Software Symantec End Point Filezilla

3.2.2. Software Gerencia de Tecnología

Tabla 3.2. Software Gerencia de Tecnología Venture Venti

	Software
Sistema Operativo	Windows 2003 R2
Desarrollo	Power Builder 11.5 Netbeans 6.5.1 Icefaces 1.8.1 Microsoft SQL 2003 Power Designer 8
Otros	Microsoft Office Venture ERP Software Symantec End Point

3.2.3. Software Servidor Principal

Tabla 3.3. Software Servidor Principal Venture Venti

	Software
Sistema Operativo	Windows 2003 R2
Otros	Symantec End Point Sybase ASA 8.02 Server Ip switch file transfer

3.3. ASPECTOS PROCEDIMENTALES

3.3.1. Gestión de Cambios por requerimientos nuevos de un cliente

La mayor parte de cambios que se realizan en el software son por requerimientos de clientes, sean estos por la necesidad de implementar, cambiar o mejorar un proceso; corrección de fallas que el software pueda presentar debido a una programación errónea a cargo de los desarrolladores. La Figura 3.2 muestra el proceso realizado para la solicitud de un cambio por parte de un cliente, en el cuál para la realización del mismo intervienen el Gerente de Tecnología y el desarrollador.

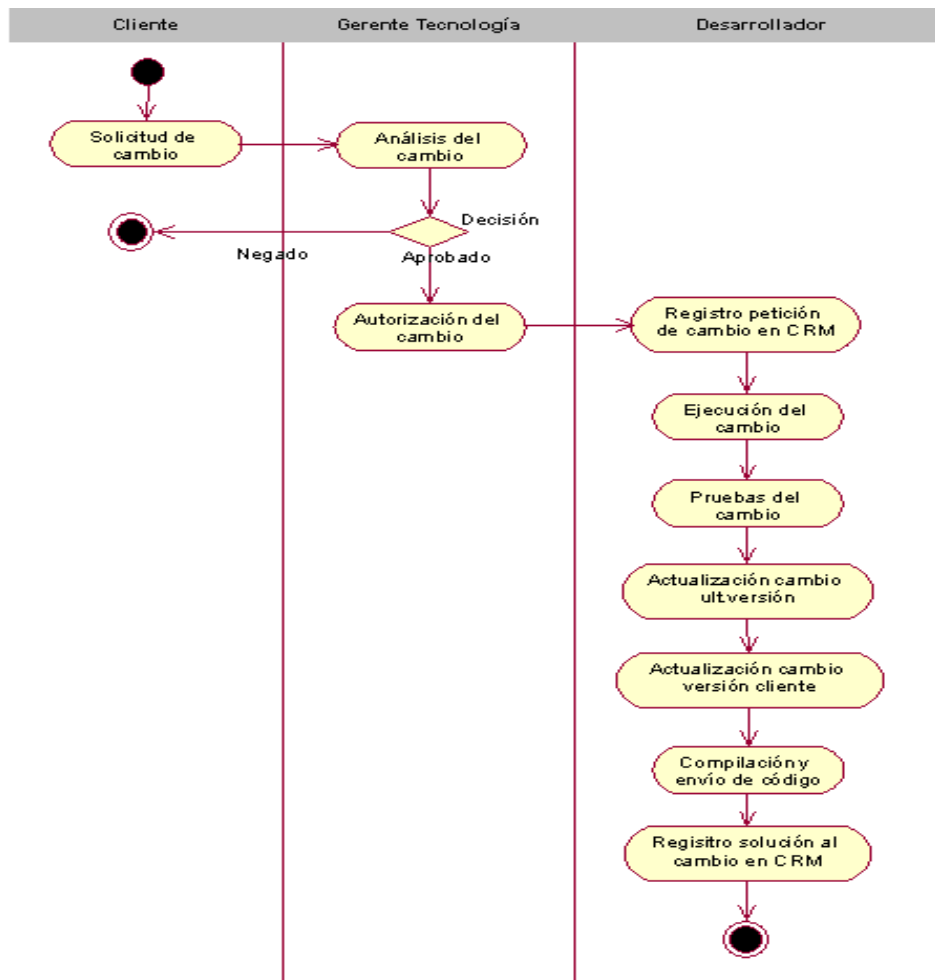


Figura 3.2. Proceso Gestión de Cambios por nuevo requerimiento de un cliente Venture Venti

Descripción del proceso

El proceso inicia con un cambio requerido por el cliente el que será analizado y autorizado por el Gerente de Tecnología, ya que el mismo verificará la factibilidad y si se debe cambiar la estructura principal o algún proceso crítico del software.

El desarrollador registra una tarea en el CRM con los detalles del cambio y tiene la responsabilidad de ejecutarlo, hacer las pruebas respectivas, compilar código, enviar el release al cliente y finalmente indicar que la tarea con el cambio fue desarrollada satisfactoriamente.

Debilidades Detectadas

- Una mala comprensión y documentación no adecuada de un cambio a desarrollarse para un cliente, puede causar que el gerente de tecnología pida la realización de un cambio de manera incorrecta.
- El desarrollador es la única persona que efectúa pruebas de los cambios que ha realizado, por lo tanto si comprendió o se le indicó mal el requerimiento, no existirá quién pueda dar una segunda opinión o hacer alguna objeción.
- Al detectar un error en el código luego de ser compilado y enviado a un cliente, implicará tiempo extra en una tarea para corregir el error, compilar código y enviarlo nuevamente al cliente.

3.3.2. Gestión de Cambios por corrección de error a un cliente

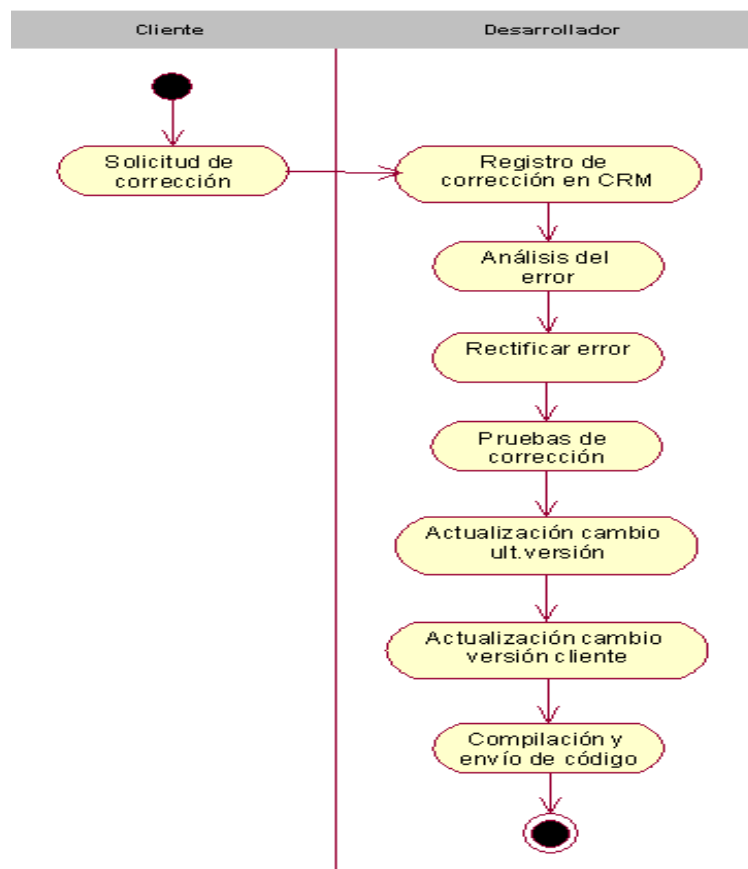


Figura 3.3. Proceso Gestión de Cambios por corrección de errores Venture

Descripción del proceso

El proceso inicia con la solicitud de corrección de un error detectado por el cliente en la aplicación que le fue entregada. El desarrollador responsable registra una tarea en el CRM indicando el requerimiento solicitado, corrige el error que le fue notificado, vuelve a realizar pruebas, compila código y envía el release al cliente.

Debilidades detectadas

- No existe el control de una persona ajena a los cambios efectuados, previo al reenvío de código con las correcciones realizadas por el desarrollador.
- Si el desarrollador tiene varias tareas pendientes, no realiza las pruebas a profundidad buscando otras posibles fallas causadas por un error o la corrección del mismo.
- Existe mucha carga de trabajo para el desarrollador, lo que causa que éste no realice su trabajo eficientemente.
- No se lleva un historial y un control detallado de los errores cometidos por un desarrollador.

3.3.3. Gestión de cambios por procesos nuevos del sistema

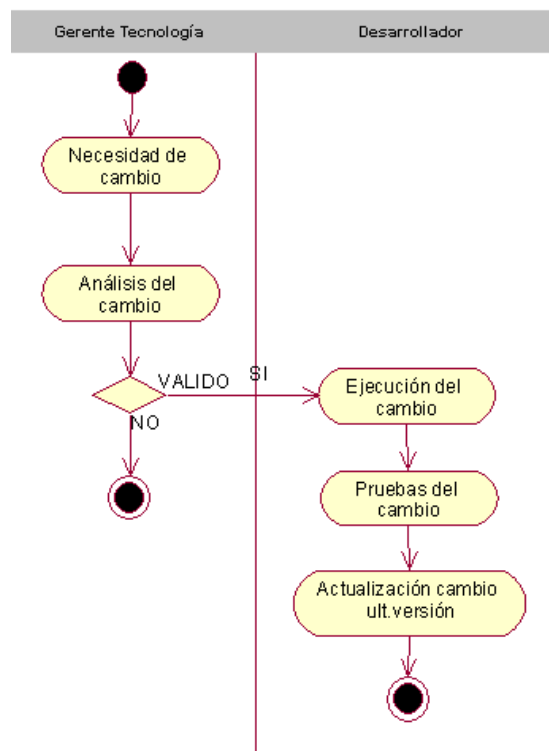


Figura 3.4. Proceso Gestión de Cambios por implementación de nuevos procesos Venture Venti

Descripción del Proceso

Se necesita de un proceso nuevo, el cual puede ser solicitado por el cliente, desarrollador o el gerente de tecnología. Este requerimiento puede ser aprobado directamente por el gerente de tecnología analizando que tan factible es la realización de un nuevo proceso. El requerimiento será elaborado, probado e implementado por un desarrollador.

Debilidades Detectadas

- Al igual que los cambios por cambios o correcciones, el desarrollador es el único que efectúa las pruebas, sin contar con una aprobación previa de alguna otra persona que pueda ratificar que el proceso está realizado correctamente y de acuerdo a las especificaciones iniciales.

- En varias ocasiones no existe ninguna opinión diferente a la del gerente de tecnología sobre algún nuevo proceso que se necesite realizar, evitando que un proceso pueda ser diseñado de una mejor forma.
- Se aprecia mucha carga de trabajo ya que sólo una persona es la encargada del análisis de un nuevo proceso, pudiendo existir varios cambios urgentes en un día.
- Existe dependencia del personal ya que solo existe una persona que puede realizar el análisis y autorizar la creación de un nuevo proceso.

3.3.4. Control de Versiones

El control de versiones se lo lleva a cabo de acuerdo al código en PowerBuilder, como en Java.

Descripción del Proceso

Se tiene una carpeta del código en el servidor principal, la que se encuentra compartida en la red para que los desarrolladores tengan acceso para poder trabajar. Según el área que pertenezca un desarrollador, éste tiene la responsabilidad de coordinar con su grupo de trabajo sobre las clases u objetos en los que va a realizar algún tipo de modificación.

Si se debe enviar código a algún cliente, el desarrollador debe revisar que el resto de personas que se encuentran trabajando en el código cierren la herramienta de desarrollo, sacar una copia en su computador de trabajo y compilar el código para enviarlo a un cliente.

Debilidades detectadas

- Si se da algún problema en la red o variación de voltaje, se pueden producir fallas, daños o pérdida de datos con el código que se trabaja.
- No se optimiza el tiempo de todos los desarrolladores, al tener que obtener una copia del código para que pueda ser compilado y enviado al cliente.
- No se utiliza ningún tipo de proceso de ingeniería o herramienta para controlar los cambios que se realizan en clases y objetos.
- Una versión errónea realizada en el código puede causar problemas e inconvenientes al resto de desarrolladores de otros departamentos.
- Una versión del código puede llevar a ser inestable si no se han realizado las actualizaciones correspondientes del código por parte de un desarrollador.

3.3.5. Pruebas

Las pruebas son realizadas para verificar que los requerimientos han sido desarrollados acorde a lo pedido; estas pruebas están a cargo de los desarrolladores en sus respectivas áreas.

Descripción del Proceso

Luego de haber concluido un requerimiento el desarrollador tiene la obligación de revisar que lo que ha realizado esté funcionando correctamente, para esto se ingresan datos revisando que los resultados obtenidos sean los esperados.

Debilidades Detectadas

- No existe documentación adecuada de los procesos y tipos de pruebas que se deben realizar para garantizar la calidad de un producto software a ser entregado.
- No existen casos de pruebas para poder determinar todo lo que se debe revisar dependiendo del cambio realizado y dónde se lo haya hecho.
- El desarrollador es el único que controla que lo realizado está correcto, pudiendo existir algún problema sin percatarse del error.
- No se realizan pruebas con flujos alternos en el caso de que las entradas de datos sean otras.
- Si existen varias pruebas pendientes, el desarrollador puede no realizar cada una de estas de una forma adecuada, al tener que terminar varias revisiones a la vez.

3.3.6. Problemas actuales de la empresa

Son varios los problemas que la empresa posee actualmente relacionados al proceso de GCS; se ha tomado una muestra de tres desarrolladores que se encuentran actualmente trabajando en la organización en un lapso de 9 meses (abril-diciembre 2010), para poder ilustrar las deficiencias que se buscará corregir como se muestra en la Tabla 3.4 y Figura 3.5

Tabla 3.4. Problemas, cantidades y porcentajes relacionados al GCS

Venture Venti (abr-dic 2010)

PROBLEMA	CANTIDAD	PORCENTAJE
Error de programación nivel 1	7	16,28
Error de soporte nivel 3	3	6,98
Error de Programación nivel 3	6	13,95
Error de Programación nivel 2	2	4,65
No cumplir estándares	12	27,91
No actualizar código en el servidor	9	20,93
No realiza control de calidad 3	3	6,98
No realiza control de calidad 1	1	2,33
TOTAL:	43	100

NOTA: La valoración de los procesos en la Tabla 3.4 está definida con escala de 1 a 3, donde 3 es un tipo de falta de gravedad alta

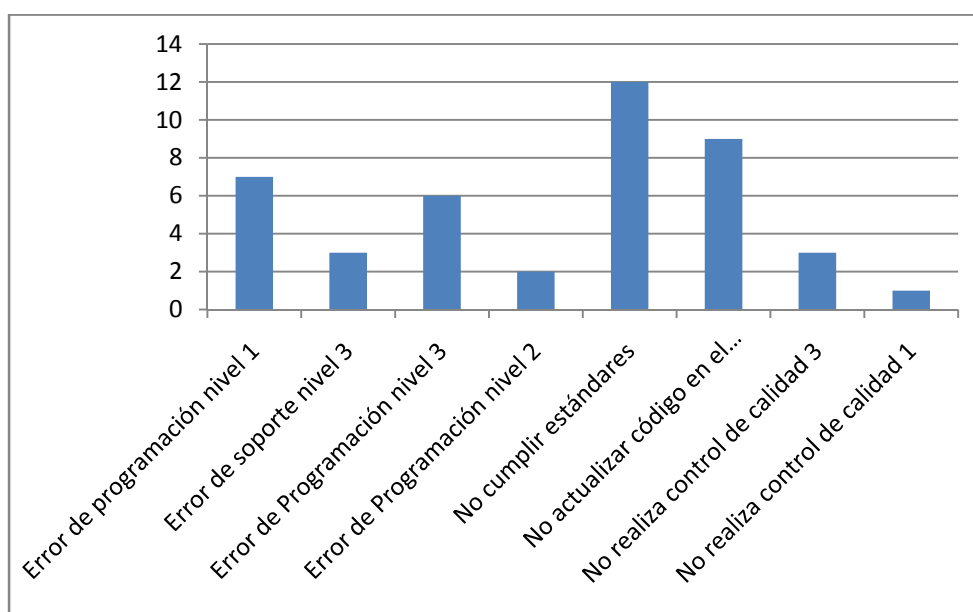


Figura 3.5. Deficiencias de personal en GCS Venture Venti (abr-dic 2010)

La Tabla 3.4 y Figura 3.5 indican los problemas que tienen actualmente en la organización, pudiéndose reconocer que no son simplemente problemas en el desarrollo de software, sino también problemas de control e ingeniería que están involucrados dentro del proceso de GCS como la actualización de código.

CAPÍTULO IV: ANÁLISIS COMPARATIVO ENTRE METODOLOGÍAS Y ESTÁNDARES RELACIONADOS AL PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE.

4.1. INTRODUCCIÓN

Para realizar el análisis comparativo, se seleccionaron metodologías y estándares que cumplen e involucran a la Gestión de la Configuración del Software, con esto se buscará obtener la adaptación mas adecuada de las actividades según los requerimientos y forma de trabajar de la organización en la cual se van a aplicar los procesos de GCS.

4.2. IDENTIFICADORES DE EVALUACIÓN PARA EL PROCESO DE GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

Se tomarán dos aspectos para la evaluación de modelos, metodologías o estándares:

- Especificación de Actividades: Describe las actividades que se deben realizar.
- Establecimiento de Roles: Se indica a las personas involucradas, con sus roles en las diferentes actividades.
- Herramientas: Existen herramientas en el mercado que permitan la automatización de los procesos.

4.3. PESOS PARA EVALUACIÓN

La Tabla 4.1 muestra los valores y su descripción con los cuales se evaluará a las metodologías, estándares o modelos que dentro de sus procesos involucran a la Gestión de la Configuración del Software

Tabla 4.1. Pesos para la Evaluación Gestión de la Configuración del Software

Valor	Título	Descripción
0	Nulo	No existe.
1	Cumple Poco	Provee poca información sobre la forma de ejecutar el proceso.
2	Cumple Medianamente	Provee información de una manera satisfactoria sobre la ejecución del proceso.
3	Cumple Adecuadamente	Provee información de una manera adecuada y clara sobre la forma de ejecutar el proceso.

4.4. EVALUACIÓN: IMPLEMENTACIÓN DEL PROCESO

Para esta evaluación no se tomará en cuenta el factor herramientas, ya que no se requiere del mismo, porque es un documento el que se buscará generar.

Tabla 4.2. Evaluación Implementación del Proceso

	Especi_ fica Activida_ des	Esta_ blece Roles	TOTAL	Anotación
ISO/IEC 12207	3	1	4	Establece actividades de una forma adecuada, describiendo lo que busca cumplir, pero se generalizan mucho los roles de los involucrados
METRICA V3	2	1	3	Establece actividades claras, al igual que roles que intervienen, entradas y salidas; pero las entradas se basan en actividades previas de la metodología Métrica V3
RUP	3	2	5	Especifica a detalle las actividades a realizar dentro del proceso de GCS, al igual que indica los roles y en qué actividades intervienen

Según la evaluación de la Tabla 4.2, con mayor puntuación se encuentra la metodología RUP, ya que la misma especifica con claridad las actividades que se deben realizar durante la Implementación del Proceso, o el Plan de Configuración como tal. Además se establecen roles claros y en las actividades que intervendrán.

4.5. EVALUACIÓN: IDENTIFICACIÓN DE LA CONFIGURACIÓN

Tabla 4.3. Evaluación Identificación de la Configuración

	Especi_ fica Activida_ des	Esta_ blece Roles	Herra_ mientas	TOTAL	Anotación
ISO/IEC 12207	3	1	0	4	Establece las actividades a detalle de GCS en el proceso de apoyo, e indica que elementos deben ser tomados en cuenta.
METRICA V3	1	1	0	2	Se resumen todas las actividades solo en dos tareas de forma general en la interfaz de GCS
RUP	1	1	1	3	Las actividades de GCS están relacionadas con otras actividades(fases RUP) y no se detalla aspectos para identificación de todos los elementos de la configuración

Al concluir con la evaluación según la Tabla 4.3, el estándar ISO/IEC 12207 es el que obtiene mayor puntuación ya que este muestra con claridad las actividades que se deben realizar, e indica los elementos(recomendables tales como el código fuente, líneas base, librerías) que se deberán tomar en cuenta para la Identificación de la Configuración.

4.6. EVALUACIÓN: MANTENIMIENTO Y CONTROL DE LA GESTIÓN DE LA CONFIGURACIÓN

Tabla 4.4. Evaluación del Mantenimiento y Control de la Gestión de la Configuración

	Espe- fica Activid- ades	Esta- blece Roles	Herra- mientas	TOTAL	Anotación
ISO/IEC 12207	2	1	2	5	Se mencionan las actividades de una forma general dentro de los procesos de apoyo de GCS y se establece un único rol generalizado para ejecución de las mismas.
METRICA V3	2	2	2	6	Establece todo el proceso en la interfaz de GCS, mediante tres tareas, en las que interviene un solo participante, y requiere de la integración de procesos anteriores
RUP	3	2	2	7	Establece las actividades a detalle dentro del proceso de GCS, con los roles, las mismas que pueden ser especificadas según su necesidad

Esta es la fase del proceso de GCS que mayor nivel de detalle necesita, donde la metodología RUP obtuvo mayor puntuación, según la Tabla 4.4. RUP muestra las actividades, los roles de una forma clara y detallada lo cual facilita su entendimiento y comprensión acerca de las herramientas, todas poseen calificación con valor 2, teniendo en cuenta que el control de versiones y cambios depende cómo una organización trabaje y las herramientas que se deseen utilizar.

NOTA: En la columna Herramientas, el valor 2 asignado significa que existen en el mercado herramientas que pueden ser utilizadas.

4.7. EVALUACIÓN INFORME DE ESTADO DE LA CONFIGURACIÓN

Tabla 4.5. Evaluación del Informe de Estado de la Configuración

	Espe_ cífica actividades	Esta_ blece Roles	Herra_ mientas	TOTAL	Anotación
ISO/IEC 12207	3	1	0	4	Se establece de forma clara las actividades del proceso de GCS que se deben realizar, y se busca obtener
METRICA V3	0	0	0	0	No posee
RUP	1	1	1	3	Indica las actividades de forma muy general ya que esta fase es parte de la Auditoría de la GCS en los procesos RUP

Al terminar el análisis que se muestra en la Tabla 4.5, para el Informe de Estado de la Configuración, se tomarán las actividades de ISO/IEC 12207 ya que son claras e indican lo que se busca obtener con los informes de estado de la configuración.

4.8. EVALUACIÓN AUDITORÍA DE LA CONFIGURACIÓN

Tabla 4.6. Evaluación Auditoría de la Configuración

	Espe_ cífica actividades	Esta_ blece Roles	Herra_ mientas	TOTAL	Anotación
ISO/IEC 12207	2	1	0	3	Se establecen las actividades de GCS divididas en dos grupos, detallando que se debería hacer en cada grupo
METRICA V3	0	0	0	0	No posee
RUP	2	1	1	4	Detalla las actividades en los procesos de GCS y los pasos para realizar las auditorías con un respectivo rol para los usuarios.

Al analizar las actividades para la Auditoría de la Configuración en la Tabla 4.6, se concluyó que el proceso más adecuado es RUP, debido a que las actividades como la realización de la auditoría física, funcional y las conclusiones del informe son claras, además se define un rol para la ejecución de las mismas.

4.9. CONCLUSIÓN DE LA EVALUACIÓN

Al concluir con el análisis y evaluación de las actividades (resumido en la Tabla 4.7) dentro de los elementos funcionales del proceso de Gestión de la Configuración del Software se escogió las actividades de las metodologías, procesos, o estándares que obtuvieron la mayor puntuación, los cuales puedan ayudar de mejor forma acoplándose a las necesidades de la organización y así solventar los problemas que se buscan solucionar.

Tabla 4.7. Conclusión de Evaluaciones

	Implementación del Proceso	Identificación de la Configuración	Evaluación Mantenimiento y Control de la Gestión de la Configuración	Informe de Estado de la Configuración	Auditoría de la Configuración
ISO/IEC 12207	4	4	5	4	3
METRICA V3	3	2	6	0	0
RUP	5	3	7	3	4

4.10. RESUMEN METODOLOGÍA PROPUESTA

La Tabla 4.8 ilustra la metodología para el proceso de GCS que será implementada en la empresa Venture Venti.

Tabla 4.8. Resumen de metodología propuesta

	ACTIVIDADES	ROLES	HERRAMIENTAS	ENTREGABLE
IMPLEMENTACIÓN DEL PROCESO	<ul style="list-style-type: none"> - Definir prácticas de Identificación de la Configuración - Definir prácticas para las líneas base - Definir prácticas para archivar 	<ul style="list-style-type: none"> - Líder del Proyecto 		<ul style="list-style-type: none"> - Políticas para la Gestión de la Configuración del Software
IDENTIFICACIÓN DE LA GCS	<ul style="list-style-type: none"> - Seleccionar de los Elementos de la Configuración del Software (ECS) - Identificar las líneas base - Definir esquema de Identificación - Definir y establecer las Bibliotecas Software <ul style="list-style-type: none"> - Biblioteca de Trabajo - Biblioteca de Soporte al Proyecto - Biblioteca Maestra - Biblioteca Backup 	<ul style="list-style-type: none"> - Líder del Proyecto - Gestor de la Configuración del Software - Desarrollador 		<ul style="list-style-type: none"> - Plan de Gestión de la Configuración del Software
MANTENIMIENTO Y CONTROL DE LA GCS	<ul style="list-style-type: none"> - Elaborar el Plan de Pruebas - Elaborar Casos de Pruebas - Establecer Control de Versiones - Establecer Control de Cambios 	<ul style="list-style-type: none"> - Líder del Proyecto - Gestor de la Configuración del Software - Desarrollador - Comité de Control 	<ul style="list-style-type: none"> - Venture ERP Software - Power Builder 11.5 - Netbeans 6.5.1 - Netbeans 6.7 - Visual SVN Server - CollabNet Subversion Client 	<ul style="list-style-type: none"> - Plan de Pruebas - Casos de Pruebas
INFOME DE ESTADO DE LA GCS	<ul style="list-style-type: none"> - Seleccionar reportes basados en peticiones de cambios - Definir frecuencia de reportes 	<ul style="list-style-type: none"> - Líder del Proyecto - Gestor de la Configuración del Software 	<ul style="list-style-type: none"> - Venture ERP Software 	<ul style="list-style-type: none"> - Políticas para la Gestión de la Configuración del Software
AUDITORÍA DE LA GCS	<ul style="list-style-type: none"> - Realizar Auditoría de la Configuración Física <ul style="list-style-type: none"> - Realizar Auditoría de la Configuración Personal - Diseñar Reporte de Novedades 	<ul style="list-style-type: none"> - Gestor de la Configuración del Software 		

CAPÍTULO V: GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

5.1. MODELO Y/O ESTÁNDAR IDENTIFICADO

Según el análisis realizado en el Capítulo IV en la Tabla 4.2, Tabla 4.3, Tabla 4.4, Tabla 4.5 y Tabla 4.6; se escogieron los procesos de GCS acorde a las necesidades y acoplamiento de la empresa auspiciante.

5.2. IMPLEMENTACIÓN DEL PROCESO

Se debe preparar un plan de Gestión de la Configuración. El plan debe describir las políticas con las que se va a trabajar, actividades de Gestión de la Configuración, procesos y responsables que llevarán a cabo dichas actividades.

NOTA: Ver **ANEXO C:** Políticas para la Gestión de la Configuración del Software y **ANEXO D:** Plan de Gestión de la Configuración

5.3. IDENTIFICACIÓN DE LOS ECS(ELEMENTOS DE LA CONFIGURACIÓN DEL SOFTWARE)

La identificación de los ECS es una de las actividades más importantes dentro del proceso de GCS ya que aquí se definirán los elementos que serán controlados como por ejemplo el código fuente, documentos; y cómo estos elementos serán identificados, marcados, numerados, y su estructura de archivos que utilizarán.

5.3.1. Selección de los ECS

Se deberá describir los ECS que serán controlados por las actividades de GCS. Estos elementos se los agrupará de acuerdo a las actividades principales de trabajo de la organización como se indica en la Tabla 5.1.

Tabla 5.1. Elementos Configuración del Software Venture Venti

Disciplinas Básicas	Código	Nombre EC
Requisitos	ERS	Especificación de requerimientos de software
	DVP	Documento visión del proyecto
Análisis	MA	Modelo de Análisis
Diseño	MBDC	Modelo Base de Datos Conceptual
	MBDF	Modelo Base de Datos Físico
	DC	Diagrama de clases
Implementación	CF	Código fuente
	CE	Código ejecutable
	CFW	Código fuente web
	CEW	Código ejecutable web
	SBD	Script de implementación de la base de datos
Pruebas	PP	Plan de Pruebas
	CP	Caso de Pruebas
Implantación	MU	Manual de Usuario
	MUW	Manual de Usuario Web
Disciplina de Gestión	PGC	Plan Gestión de la Configuración

NOTA: Ver **ANEXO D:** Plan de Gestión de la Configuración

5.3.2. Identificación Líneas Base

Las líneas base serán creadas con una versión única de todo el sistema, teniendo en cuenta que será creada cuando un ECS sufra un cambio significativo para el Líder del Proyecto. Se tomará los siguientes parámetros para el registro de las Líneas Base tal cual se lista a continuación:

- Versión
- Fecha
- Autorizado por
- Cambios(Módulo y que cambios se hicieron)

Además, la nomenclatura a utilizar se muestra en la Tabla 5.2:

Tabla 5.2. Nomenclatura Líneas Base

Código	Nombre Línea Base
LBA	Análisis
LBD	Diseño
LBC	Implementación/Construcción
LBP	Pruebas
LBI	Implantación
LBGC	Gestión de la Configuración y cambios

NOTA: Ver **ANEXO D:** Plan de Gestión de la Configuración

5.3.3. Esquema de Identificación

Elementos de configuración del software: Los ECS de los proyectos serán identificados mediante la siguiente información:

- Código del ECS
- Nombre del ECS
- Autor
- Nombre del Proyecto al que pertenece el ECS
- Identificación de la línea base a la que pertenece el ECS
- Localización
- Tipo de ECS(documento, software)
- Fecha de creación
- Identificación de la disciplina en la que se creó

Versiones y Variantes: Se aplicará el siguiente esquema de identificación de versiones y variantes para todos los ECS que se han identificado en la sección anterior, de tal forma que en todo momento tenga una tabla actualizada con información de los elementos:

- Código ECS
- Descripción del ECS
- Número de versión o variante (secuencial)
- Fecha de Creación
- Autor o Autores
- Localización
- Observación.
- Variante de plataforma operativa.

NOTA: Ver **ANEXO D:** Plan de Gestión de la Configuración

5.3.4. Definiciones y Establecimiento de Bibliotecas Software

Una biblioteca de Software (SW) es una colección controlada de SW y/o documentación relacionada, cuyo objetivo es ayudar a un desarrollo y mantenimiento del sistema.

- **Biblioteca de Trabajo.** Comprende el área de trabajo donde los diseñadores elaboran documentos del proyecto y donde los programadores desarrollan. Aquí se realiza la codificación y pruebas unitarias. Una vez realizadas las revisiones o pruebas y el ECS en cuestión ha sido revisado y aprobado, se lo transfiere a la “Biblioteca de Soporte” que posee la estructura que muestra la Figura 5.1.



Figura 5.1. Estructura Biblioteca de Soporte Venture Venti

- **Biblioteca de Soporte al Proyecto.** Se almacenan los ECS aprobados y transferidos desde la “Biblioteca de Trabajo”. Cuando un ECS pasa a esta biblioteca se encuentra sujeto a un control de cambios interno, esta biblioteca tendrá la estructura que muestra la Figura 5.2.



Figura 5.2. Estructura Biblioteca de Soporte al Proyecto Venture Venti

- **Biblioteca Maestra.** Se usa para almacenar ECS liberados para su entrega al cliente o distribución en el mercado, la cual contará con la estructura que muestra la Figura 5.3.



Figura 5.3. Estructura Biblioteca Maestra Venture Venti

NOTA: Ver **ANEXO D:** Plan de Gestión de la Configuración

- **Biblioteca Backup.** Debe estar adecuadamente identificada aunque su contenido no está sujeto a GCS, cabe resaltar que los respaldos están ya definidos en el **ANEXO C:** Políticas para la Gestión de la Configuración del Software.

5.4. PRUEBAS

5.4.1. Planeamiento de Pruebas

La mejor manera de llevar un control y realizar pruebas es documentar el esfuerzo desde dos entornos diferentes. El primero contiene de manera general el plan de pruebas, mientras que el segundo contiene de forma detallada la ejecución de las instrucciones.

a) Plan de Pruebas

El alcance del plan hace referencia a toda la estrategia de pruebas, donde las aplicaciones, módulos van a ser probados y por qué. Un plan típico deberá incluir lo siguiente:

- Aplicación(es)/sistema(s) a ser probado.
- Objetivos de las pruebas y su racionalización (riesgo y requerimientos).
- Alcance y limitaciones del plan de pruebas.
- Fuentes de conocimiento del negocio para la planeación y ejecución de pruebas (opcional).
- Fuentes de datos.
- Ambiente de pruebas y su gestión.

b) Casos de Pruebas

El propósito del caso de prueba es documentar para tomar una decisión colectiva considerando el 'Qué', 'Por qué' y 'Cómo' para una aplicación, mediante el diseño de casos de pruebas que deberán incluir lo siguiente:

- ID único del caso de prueba.
- Título Único.
- Descripción rápida.
- Fase de desarrollo en la cual el caso de prueba será ejecutado.
- Especificar las metas de las pruebas y sus medidas de rendimiento.
- Datos para pruebas sugeridos.
- Procedimiento para inicializar las pruebas.
- Procedimiento para cerrar las pruebas.
- <Marcador>Número de intentos para ejecutar con éxito un caso de prueba.
- <Marcador>Lista de defectos del software encontrados en el caso de pruebas.
- <Marcador>Es el caso de prueba ejecutado satisfactoriamente? (si/no).

c) Escribir el Plan y Casos de Pruebas

El Plan de pruebas y casos de pruebas serán diseñados de acuerdo a las necesidades de la organización, tomando en cuenta los requisitos mínimos mencionados anteriormente. Este Plan de Pruebas está Basado en la IEEE 829-1998¹.

NOTA: Ver **ANEXO A:** Plan de Pruebas y **ANEXO B:** Caso de Prueba (ejemplo)

¹ IEEE 829-1998 Plantilla Plan de Pruebas

5.4.2. Realización de Pruebas

La realización de cualquier tipo de prueba que abarque la aplicación está basada en la necesidad de encontrar errores, constar que un procedimiento o requerimiento nuevo cumple con las modificaciones solicitadas. Dentro de este proceso se verá involucrado el desarrollador, gestor de la Configuración del Software y el líder del proyecto (en el caso de que un nuevo requerimiento o proceso aparezca). El diagrama de flujo de la Figura 5.4 muestra las actividades generales para el manejo de defectos entre los desarrolladores.

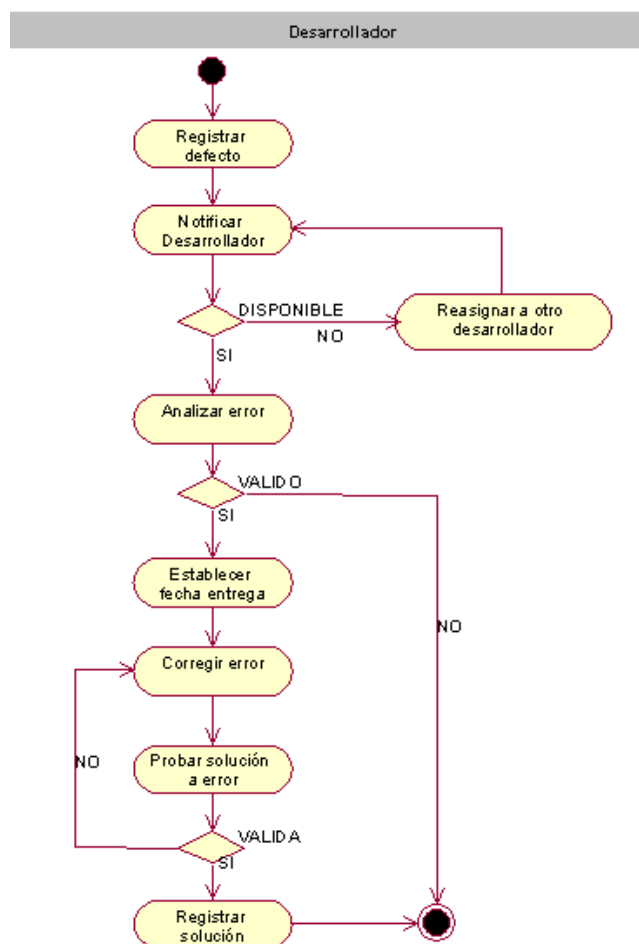


Figura 5.4. Manejo de Defectos del Desarrollador

Para las pruebas que realizará el Gestor de la configuración del Software en conjunto con el desarrollador previa la aprobación de algún cambio o corrección se manejará el siguiente flujo de trabajo como se muestra en la Figura 5.5.

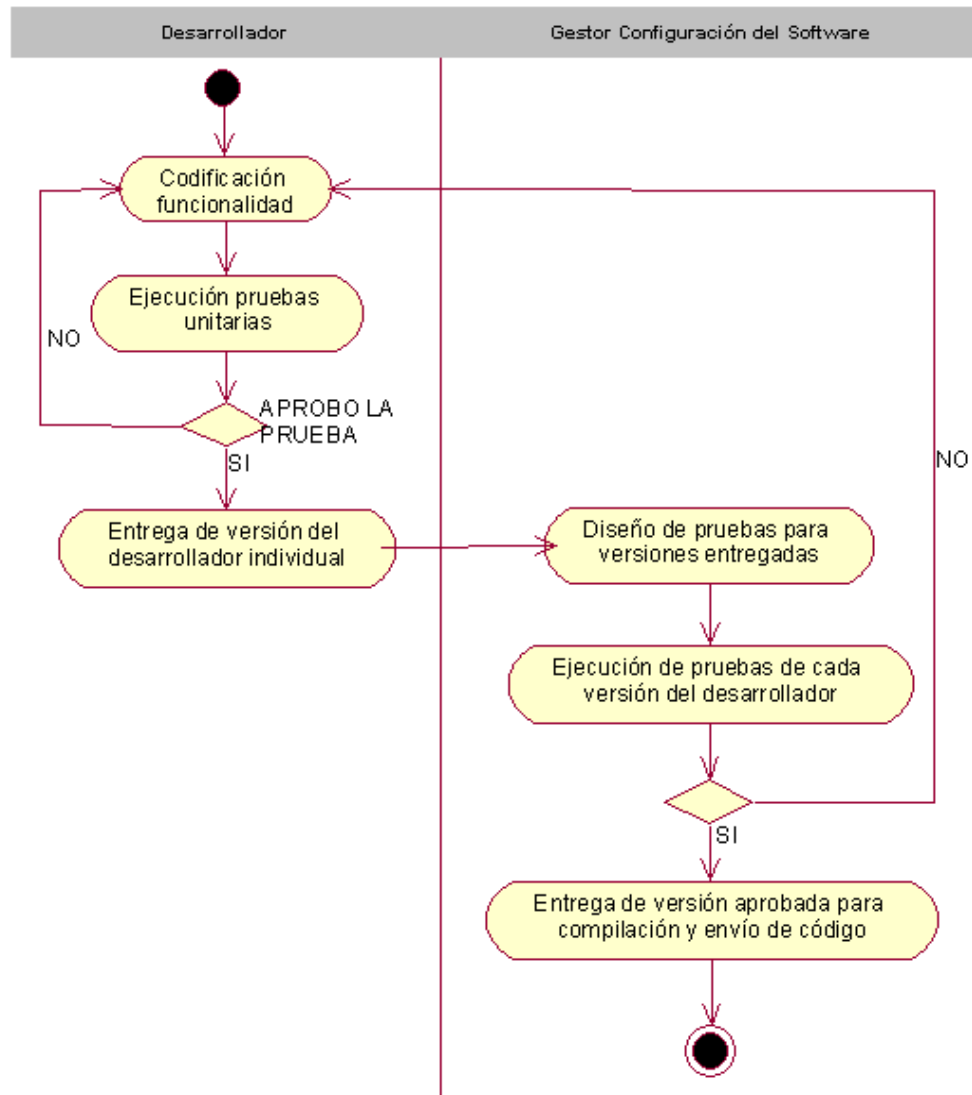


Figura 5.5. Flujo de trabajo pruebas Gestor de la configuración y desarrollador del Software

5.4.3. Entorno de Pruebas

Un entorno de pruebas adecuado permitirá observar los resultados de la ejecución de pruebas en un ambiente simulado de producción, teniendo en cuenta tanto el hardware como el software antes que éste último sea puesto en producción real. Las simulaciones son la manera de realizar pruebas en la organización, pero el valor informativo que estas brindan no será bien aprovechado sin un buen entorno de pruebas.

La simulación se realizará tomando en cuenta los requerimientos funcionales del software, enfocándose en lo que se busca comprobar y cuáles son los resultados que se buscan obtener. En virtud a los requerimientos se busca simular las pruebas acercándose lo más posible a la realidad de un giro del negocio de manera funcional (técnicas de caja blanca) la que se verificará con los resultados obtenidos.

NOTA: Ver **ANEXO A:** Plan de Pruebas

5.5. CONTROL DE VERSIONES

5.5.1. Herramientas de Versionamiento

El control de versiones se llevará en los dos entornos de trabajo que son manejados por la empresa, uno web y otro de ventanas. Al buscar una herramienta de versionamiento para cada uno de estos, se indagó en la web que sean compatibles con las características de los computadores y se puedan integrar con los IDEs² de desarrollo; tomando en cuenta que estos sean gratuitos para ahorrar gastos a la empresa.

² Integrated Development Environment (Entorno de Desarrollo Integrado)

Los IDEs con los que se trabaja en la organización son Netbeans para el entorno web, y PowerBuilder para el de ventanas. La Figura 5.6 muestra el diseño para el versionamiento que se implementará para ambos ambientes de trabajo:

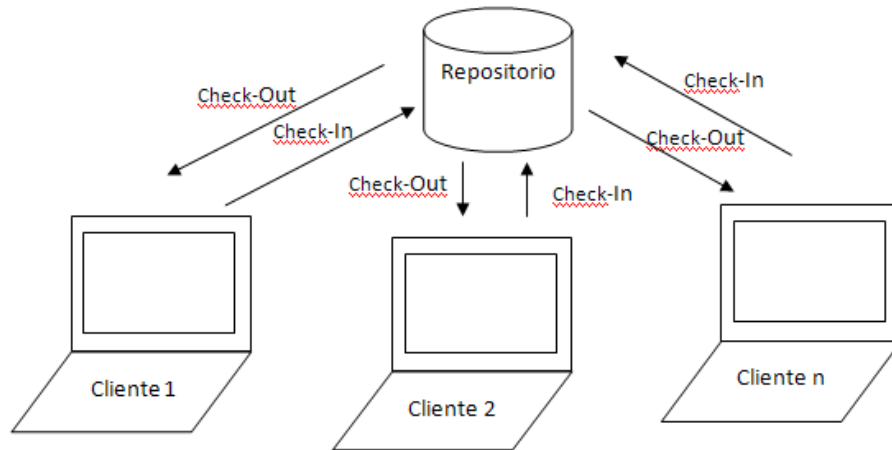


Figura 5.6. Diseño Versionamiento implementado Venture Venti

a) Netbeans

Netbeans es el IDE para el desarrollo de aplicaciones web, la Tabla 5.3 muestra el software que se utilizará para el versionamiento de las aplicaciones.

Tabla 5.3. Herramientas Versionamiento aplicaciones web

Software	Descripción
Visual SVN Server	Herramienta para la creación y administración de repositorios, creación de usuarios, que llevará el control de versiones desarrolladas en Netbeans 6.5.1 o 6.7.
CollabNet Subversion Client	Plug-in que permite la integración de Visual SVN con Netbeans

Configuración Versionamiento Netbeans

- 1) Al haber sido instalado Visual SVN en el servidor, se procederá a configurar la herramienta (Visual SVN, 2005), creando los repositorios que utilizarán como se muestra en la Figura 5.7

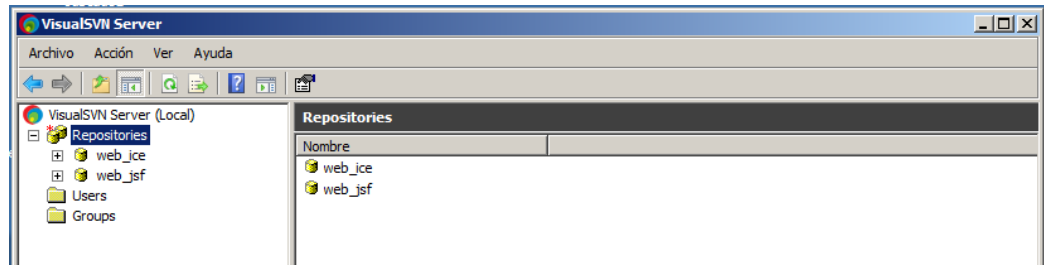


Figura 5.7. Creación repositorios en VisualSVN

- 2) Luego de Instalar CollabNet Subversion Client en las estaciones de trabajos se deberá subir el código desde Netbeans señalando el proyecto, luego click derecho Versioning/Import Into Subversion Repository como muestra la Figura 5.8

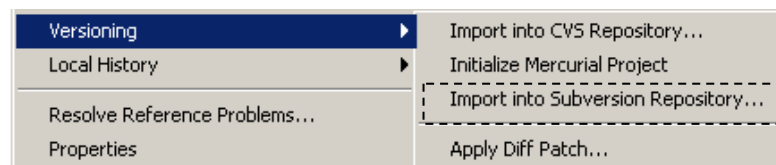


Figura 5.8. Importar Proyecto en Visual SVN

- 3) En cada uno de los clientes en los cuales se desea trabajar con un control de versiones se deberá hacer un CheckOut (Versioning/Subversion/Checkout) del Proyecto desde el Servidor Visual SVN, e indicar el proyecto, usuario(creado previamente) con el que se va a registrar y donde se va a guardar localmente el proyecto como muestra la Figura 5.9

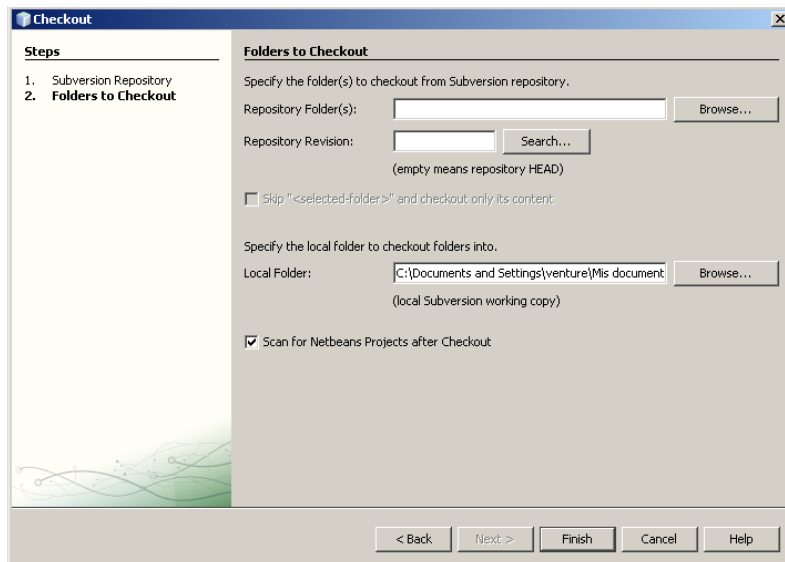


Figura 5.9. Checkout proyecto desde servidor Visual SVN

Finalmente, al llenar los campos requeridos, se procederá a descargar el proyecto desde el repositorio para ya empezar a trabajar con control de versiones, como muestra la Figura 5.10.

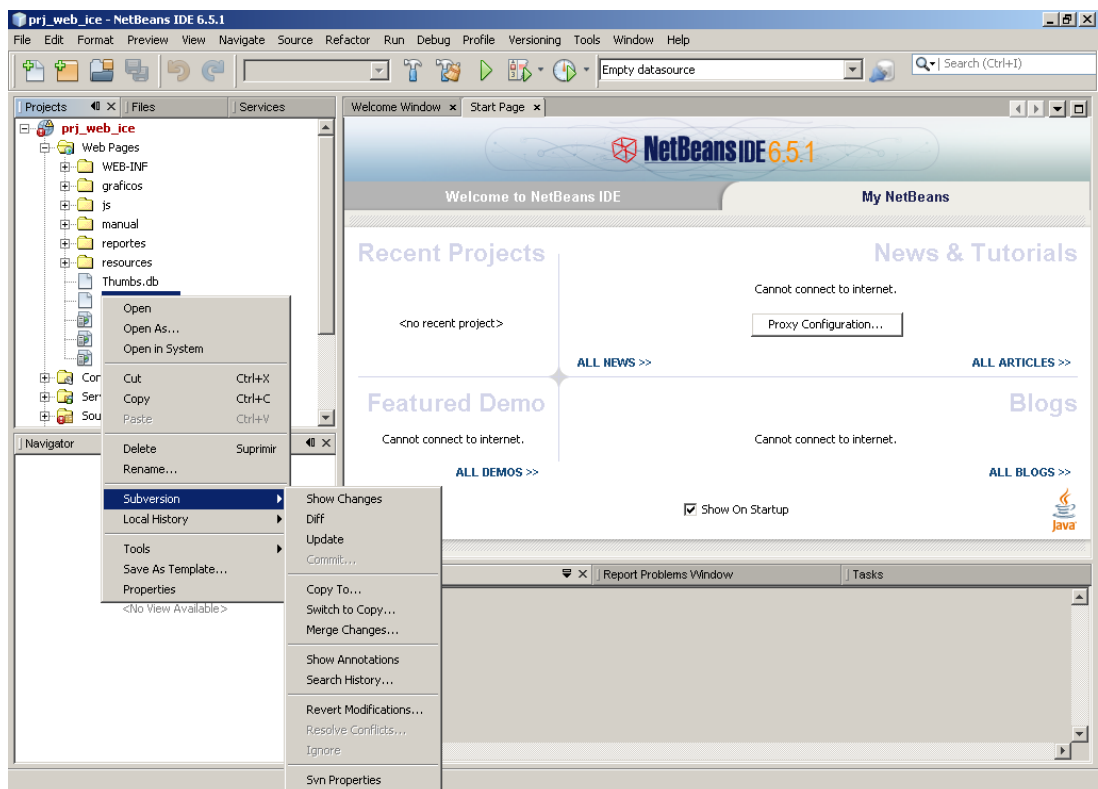


Figura 5.10. Versionamiento implementado en Netbeans

La Tabla 5.4 muestra los comandos con los que se trabajará en el versionamiento de Netbeans.

Tabla 5.4. Comandos para versionamiento Netbeans

Comando	Descripción
Show Changes	Muestra los elementos que han sido modificados o creados en el proyecto
Diff	Muestra los cambios entre una clase que se encuentra en el repositorio y la clase de trabajo local
Update	Actualiza los cambios que se encuentren almacenados en el repositorio.
Commit	Actualiza los cambios locales al repositorio
Merge Changes	Une y muestra los cambios de una o varias clases modificadas en el repositorio en la copia de trabajo local.
Show Anotations	Muestra las anotaciones que tiene una clase
Search History	Muestra la lista de cambios con su descripción, y usuario en un rango de fechas que se encuentren en el repositorio
Revert Modifications	Revierte los cambios actualizados desde el repositorio, o localmente
Resolve Conflicts	Muestra los conflictos generados por cambios en una misma clase realizado por varios desarrolladores.

b) PowerBuilder y su configuración

Para el caso de PowerBuilder, este IDE de desarrollo viene con su propia herramienta de versionamiento incluida la cual únicamente requiere ser configurada.

- 1) Se creará una carpeta en el servidor, la cual se utilizará como repositorio y para el control de cambios, versiones y una carpeta donde se trabajará localmente. La carpeta del repositorio en el servidor deberá estar compartida y con permisos de lectura-escritura. La Figura 5.11 ilustra la estructura:

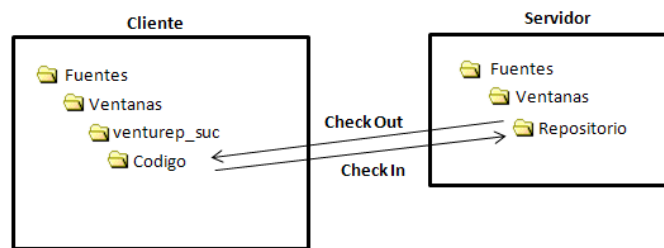


Figura 5.11. Estructura para Versionamiento PowerBuilder Cliente-Servidor

- 2) Se abrirá el código en una máquina cliente y se configurará el control de versiones, dando click derecho sobre el proyecto /Properties/SourceControl, y se ingresará un nombre que identificará al usuario que realiza un cambio, la ubicación local del trabajo y cuál es la carpeta que llevará el control de versiones (SYBASE, 2008) como muestra la Figura 5.12.

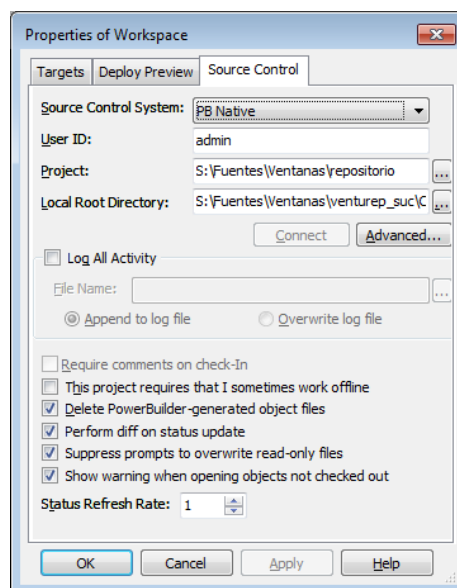


Figura 5.12. Configuración de proyecto para versionamiento Power Builder

- 3) Se agregará el código al repositorio, para esto dar click derecho sobre el Proyecto /Add to Source Control como se indica en la Figura 5.13

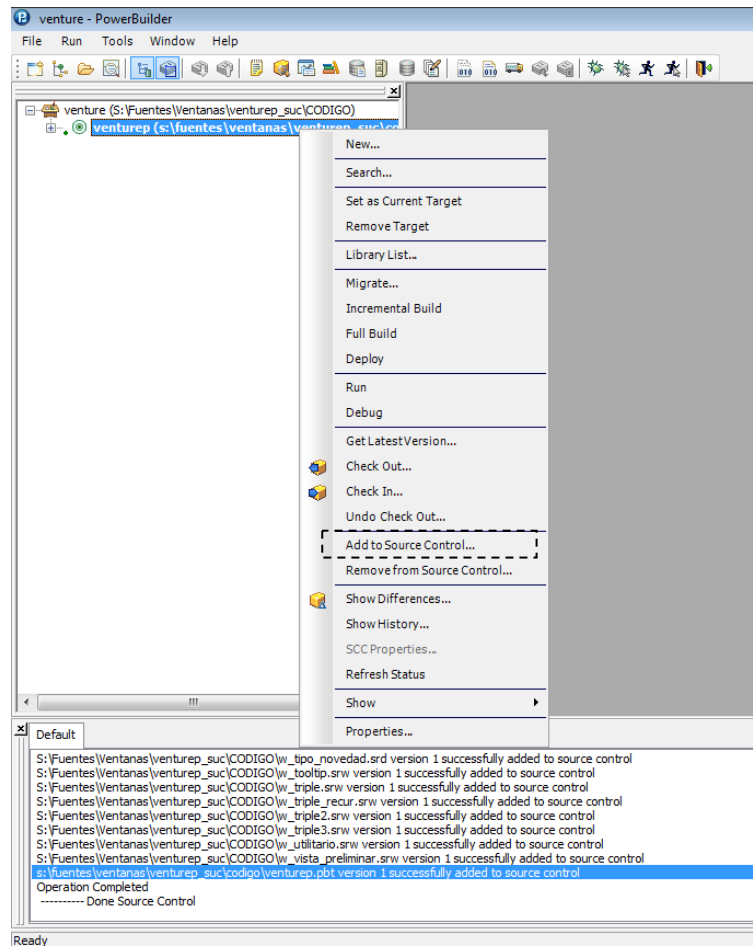


Figura 5.13 Agregar Control de Versiones a un proyecto de Power Builder

- 4) Los equipos que vayan a trabajar con el versionamiento, deberán realizar una copia local del código que fue subido al repositorio, y realizar la configuración mostrada en los pasos 1 y 2.

5.6. CONTROL DE CAMBIOS

5.6.1. Petición de cambio

Los cambios en objetos de desarrollo son propuestos con la petición de cambios. La petición de cambios es usada para documentar y rastrear defectos, mejorar la solicitud de cambios, y cualquier otro tipo de petición para un cambio de un producto.

Si se encuentra un defecto, se buscará dar una solución estableciendo una acción correctiva y realizar el seguimiento a la misma. La petición de cambios será usada por: El líder del proyecto, el Gestor de la configuración del software, y los desarrolladores cuando vean la necesidad de realizar un cambio.

5.6.2. Evaluación del cambio

Los siguientes atributos son útiles a la hora de tomar la decisión cuando se produzca una petición de cambio:

- **Tamaño**
 - a) ¿Cuánto del trabajo existente va a sufrir el cambio?
 - b) ¿Cuánto trabajo nuevo se necesitará realizar?
- **Alternativas**
 - a) ¿Existe alguna alternativa de realizar el cambio?
- **Complejidad**
 - a) ¿Es el cambio propuesto fácil de realizar?
 - b) ¿Cuáles son las posibles ramificaciones de realizar el cambio?
- **Severidad**
 - a) ¿Cuál es el impacto de implementar o no el cambio?
 - b) ¿Existe alguna pérdida de datos o trabajo involucrado?
 - c) ¿Existe alguna molestia causada?

- **Calendarios**
 - a) ¿Para cuándo se requiere el cambio?
 - b) ¿Es factible?
- **Impacto**
 - a) ¿Cuáles son las consecuencias de realizar el cambio?
 - b) ¿Cuáles son las consecuencias de no realizar el cambio?
- **Costo**
 - a) ¿Cuál es el ahorro de realizar el cambio?
- **Relación con otros cambios**
 - a) ¿Existirá algún otro cambio que se vea afectado por la realización del cambio?
- **Pruebas**
 - a) ¿Existe algún tipo de prueba especial?

5.6.3. Informe de Cambios

Luego de realizar la evaluación del cambio pedido, se presentará un informe oral por parte del desarrollador al Líder del proyecto o al Comité de Control de cambios, para que él sea encargado de tomar la decisión si el cambio se realizará o no.

5.6.4. Generación orden de Cambio de Ingeniería

Las peticiones de cambios luego de ser analizadas y aprobadas serán registradas como una tarea a un desarrollador dentro de CRM/Tareas CRM del sistema Venture. El Gestor de la Configuración del Software dará seguimiento a la tarea hasta que sea completada dentro de las fechas establecidas.

5.6.5. Objeto dado de baja

Cuando el líder del Proyecto llegue a la conclusión que por algún cambio que se desee realizar, es más fácil crear un nuevo objeto que modificar uno existente en su codificación interna, éste pasará a ser dado de baja.

5.6.6. Realización del cambio

La realización de un cambio es una actividad donde un miembro del equipo de desarrollo tiene acceso a un objeto para la modificación del mismo con la finalidad de cumplir un requerimiento de trabajo.

Como primer paso se deberá registrar una tarea en la pantalla de tareas del CRM del Sistema Venture, en la cual se describirá las actividades que se deben realizar, indicando fecha de entrega, responsable, y qué persona ordenó el cambio.

Como siguiente paso, el desarrollador tendrá como responsabilidad el crear u obtener los objetos que necesite para cumplir con la tarea encargada (código, manuales, requerimientos).

Al tener el proyecto un control de cambios, se deberá realizar las operaciones para la obtención de código actualizado, y posteriormente subir los cambios realizados. Las operaciones primarias de cualquier persona de desarrollo son las siguientes:

- a) **Check Out:** Obtener permiso para realizar el cambio a un elemento.
- b) **Check In:** Guarda una nueva versión de los cambios realizados en un elemento que fue hecho previamente un "Check Out".
- c) **Agregar a Control de fuentes:** Crea un nuevo archivo o directorio con control de versiones, creando una versión inicial (esta acción se realiza una sola y única vez).
- d) **Entrega:** Envía los cambios al integrador (Repositorio).

5.6.7. Revisión del cambio

En la revisión de cambios, se controlará la modificación de los elementos, y que estos cumplan con los requerimientos que fueron establecidos para la realización del cambio, esta actividad estará a cargo del Gestor de la Configuración quién será el responsable de validar su realización.

El proceso de cambios de la Figura 5.14 indica actividades y roles de los encargados del proceso de GCS.

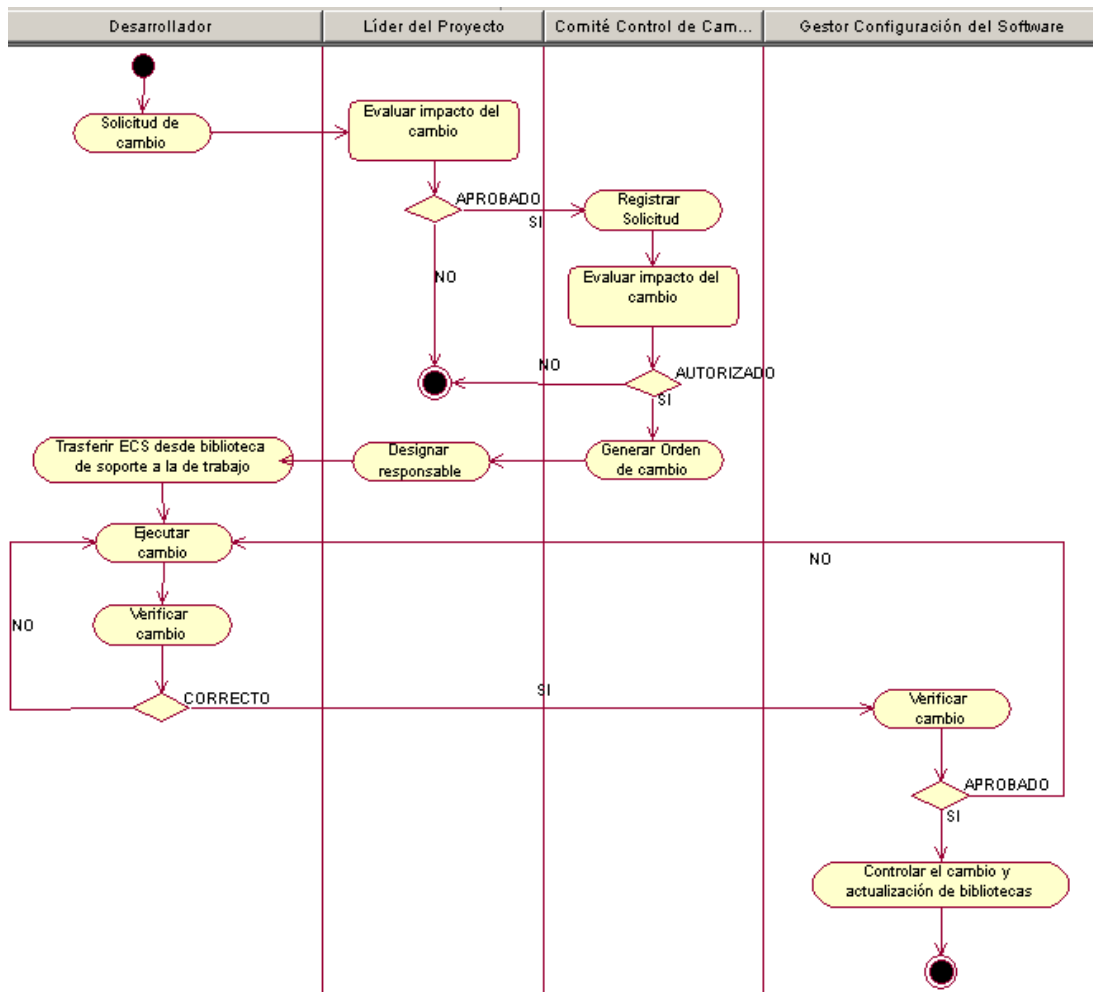


Figura 5.14. Control de Cambios proceso GCS

NOTA: Ver ANEXO D: Plan de Gestión de la Configuración

5.7. GENERACIÓN DE INFORMES SOBRE CAMBIOS DE CONFIGURACIÓN

La generación de los informes sobre cambios de configuración del software, tiene como finalidad el reportar el status de los cambios propuestos, el estado de una implementación que ha sufrido cambios, para facilitar la revisión de defectos mediante la generación de reportes. Estos informes serán generados basándose en políticas que la organización establezca como se muestra en el Anexo C: Políticas para la Gestión de la Configuración del Software.

Ver **ANEXO C:** Políticas para la Gestión de la Configuración y **ANEXO D:** Plan de Gestión de la Configuración

5.8. AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE

5.8.1. Introducción

El propósito de las auditorías es determinar que las líneas base contengan todos los elementos requeridos y cumplan con los requerimientos indicados.

5.8.2. Realizar Auditoría de la Configuración Física

Esta auditoría identifica los componentes de un producto que va a ser obtenido del Repositorio del Proyecto. Los pasos son:

- a) Identificar las líneas base que van a ser desplegadas (típicamente solo el nombre y/o número).
- b) Confirmar que todos los elementos requeridos, como los códigos fuentes se encuentren en la línea base. Los elementos faltantes serán mostrados en los Reportes de Novedades.

5.8.3. Realizar Auditoría de la Configuración Funcional

La Auditoría Funcional, confirma que las líneas base cumplen con los requerimientos asignados. Los pasos para realizar esta auditoría son:

- a) Preparar un reporte que liste cada requerimiento asignado a una línea base, su procedimiento de prueba correspondiente y el resultado de la prueba para la línea base.
- b) Confirmar que cada requerimiento tiene su prueba y que todas las pruebas del requerimiento hayan sido cumplidas. Listar cualquier requerimiento que no tenga procedimiento de pruebas, pruebas fallidas, o incompletas en los Reportes de Novedades.
- c) Generar una lista de actividades incompletas dentro de la lista de requerimientos en los Reportes de Novedades.

5.8.4. Reporte de Novedades

De existir alguna discrepancia, serán colocadas en el Reporte de Novedades. Los siguientes pasos complementarios deberán ser realizados:

- a) Identificar acciones correctivas. Esto requiere de la intervención de varios miembros del equipo para poder identificar el motivo de la discrepancia y tomar correcciones.
- b) Para cada acción correctiva, asignar responsabilidades para la corrección y establecer fechas plazo.

Los tópicos mostrados en la Tabla 5.5 deben ser asignados en el reporte de novedades de la auditoría de la configuración

Tabla 5.5. Tabla de Tópicos en Reporte de Novedades

Proceso	Tópicos
Introducción	Fecha de la Auditoría El propósito de la Auditoría
Auditoría Física de la Configuración	Identificación línea base Elementos faltantes
Auditoría Funcional de la Configuración	Requerimientos no probados Requerimientos fallidos Peticiónes de cambios realizadas
Acciones Correctivas	Acción tomada Persona encargada Fecha de terminación

5.9. SEGUIMIENTO GCS EN LA EMPRESA

5.9.1. Resultados obtenidos con la aplicación de la GCS

La Tabla 5.6 y Figura 5.15 muestran los resultados en la empresa al haberse implementado la GCS.

NOTA: La valoración de los procesos en la Tabla 5.6 y Tabla 5.7 está definida con escala de 1 a 3, donde 3 es un tipo de falta de gravedad alta

Tabla 5.6. Problemas, cantidades y porcentajes relacionados al GCS

Venture Venti (ene-sep 2011)

PROBLEMA	CANTIDAD	PORCENTAJE
Error de programación nivel 1	5	25,00
No informar resultado visita cliente	2	10,00
Error de programación nivel 2	2	10,00
No cumplir estándares	3	15,00
No actualizar código en servidor	8	40,00
TOTAL	20	100,00

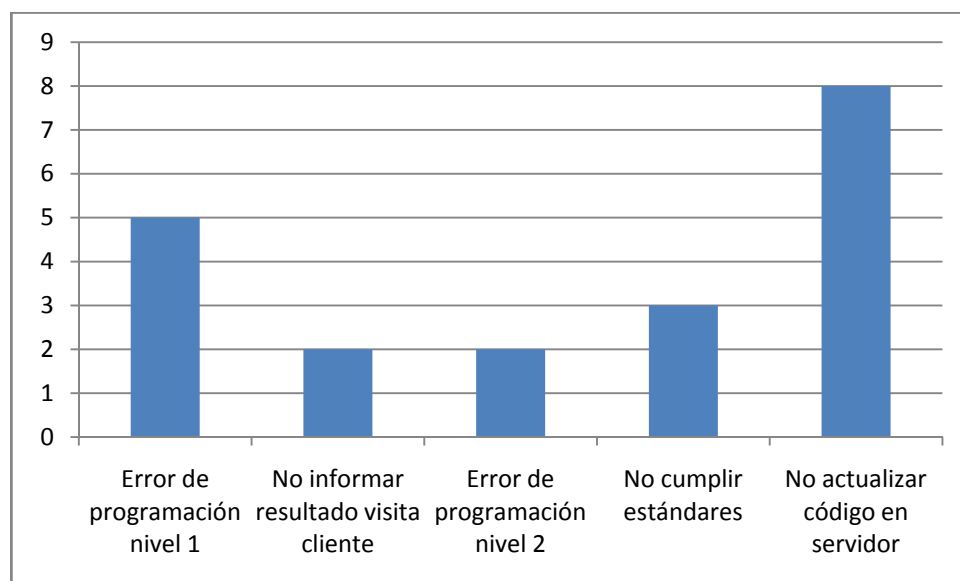


Figura 5.15. Deficiencias de personal en GCS Venture Venti
(ene-sep 2011)

Al comparar los resultados obtenidos luego de la implementación del proceso de GCS con la Tabla 3.4 se puede notar diferencias en todos los problemas que se poseía anteriormente con la GCS, existiendo la mejora por completo o reduciendo gradualmente las molestias e inconvenientes con el proceso de GCS como se muestra en la Tabla 5.7.

Tabla 5.7. Mejoras relacionadas al proceso de GCS Venture Venti (ene-sep 2011)

PROBLEMA	ABR-DIC 2010	ENE-SEP 2011	DIFERENCIA
Error de programación nivel 1	7	5	-2
Error de Programación nivel 2	2	2	0
Error de Programación nivel 3	6	0	-6
Error de soporte nivel 3	3	0	-3
No actualizar código en el servidor	9	8	-1
No cumplir estándares	12	3	-9
No informar resultado visita cliente	0	2	+2
No realiza control de calidad 1	1	0	-1
No realiza control de calidad 3	3	0	-3

Hay que mencionar que el aumento de 2 novedades en 'no informar resultado visita a cliente' se debe a procesos de carácter administrativos que no tienen ninguna relación directa con la GCS.

5.9.2. Análisis de los resultados obtenidos.

Al haber finalizado la implementación de procesos de GCS en la empresa auspiciante se puede evidenciar una mejora considerable en los últimos 9 meses con los reportes basados de registro de novedades del sistema ERP de Venture. Indicándonos mejoras al usar la Gestión de la Configuración del software en una forma gradual pudiéndose evidenciar que las fallas más graves como el error de programación nivel 3, error de soporte nivel 3 y no realizar control de calidad nivel 3 se disminuyeron en un 100%; mientras que otros problemas se redujeron en porcentajes desde un 11.11% hasta un 75% lo que son unos claros indicadores de que el proceso ha ayudado a mejorar los procesos de GCS y calidad del software en la organización.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- En base al análisis comparativo considerando el proceso de GCS propuesto por varias metodologías y estándares, se diseñó e implementó un proceso personalizado de GCS en la empresa de desarrollo Venture Venti. El cual ayudó a solventar de una forma muy satisfactoria los problemas que se tenía durante el desarrollo de software.
- Se diseñó un proceso de control de cambios bien definido, el mismo que especifica roles de trabajo y las actividades que cada participante realizará, buscando siempre mejorar la calidad del trabajo individual y colectivo de los involucrados.
- En cuanto al proceso de pruebas se planteó un plan y definieron casos de pruebas que empleará el personal encargado del desarrollo, como los responsables de su control mediante el uso de plantillas, dando como resultado un mejor control de los cambios efectuados sobre el software.
- Se implementaron herramientas de versionamiento con el objetivo de evitar problemas de códigos incompletos o con errores para los IDEs de desarrollo con los que trabaja la empresa, obteniendo una mejor calidad del código desarrollado y reducción de quejas por parte de los clientes.
- Al establecer roles específicos y claros para el personal en las diferentes etapas del proceso de GCS, se logró reducir la carga de trabajo obteniendo un mejor desempeño laboral del personal.

- Con la integración de las fases diseñadas para el proceso de GCS en la empresa auspiciante, se logró establecer un mejor entorno de trabajo entre los empleados creando más confianza en sí mismo, seguridad de su trabajo y del resto de personas; ya que los procesos son claros y las herramientas de control útiles.

6.2. RECOMENDACIONES

- Mantener actualizado constantemente las Políticas, Plan de Gestión de la Configuración, Plan de Pruebas, Casos de Pruebas y todos los procesos relacionados a la GCS según se vayan presentando nuevos requerimientos o cambios, ya que de existir algún cambio importante y no registrarlo podría causar deficiencia en procesos que se encuentran trabajando actualmente con normalidad.
- Motivar al personal de trabajo sobre la importancia de realizar correctamente los procesos de GCS e indicarles los beneficios que se obtienen para ellos, la empresa y los clientes.
- El aplicar una auditoría de la configuración del software ayudará a complementar la finalidad de los informes de estado y terminar por completo el proceso de la GCS, llevando un control amplio de todas las fases de la GCS.
- Con la culminación del presente trabajo se obtiene mejoras notorias en los procesos de desarrollo de software; las cuales podrían ser una base adecuada para buscar la aplicación de un estándar de calidad de software como la ISO/IEC 9126 (estándar internacional para la evaluación de la calidad del software), dando así a la organización un fortalecimiento en el mercado de los productos software que desarrollan.

BIBLIOGRAFIA

- Aiello, B., & Sachs, L. (2010). *Configuration Management Best Practices*. Boston: Pearson.
- Amescua, A. (2005). *SPICE: Un Marco para la Evaluación de Procesos Software*. Obtenido de <http://www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc>
- Basurto, C. (s.f.). CONFERENCIAS IMPLEMENTACIÓN Y EVALUACIÓN DE LA NTP ISO/IEC 12207. *Evaluación de procesos con ISO/IEC 15504* .
- Berlack, R. H. (1992). *Software Configuration Management*. New York: John Wiley & Sons.
- Bruegge, B., & Dutoit, A. (2004). *Object-Oriented Software Engineering*. Prentice Hall.
- Cárdenas, D. (2008). Gestión de cambios y gestión de la Configuración de Software en el Departamento Informático de Transelectric. Quito, Pichincha, Ecuador.
- Comisión de Reglamentos Técnicos y Comerciales-INDECOPI. (2006). NTP-ISO/IEC 12207. Lima.
- Everett, G., & Raymond, M. (2007). *Software Testing, Testing Across the Entire Software Development Life Cycle*. Hoboken: John Wiley & Sons.
- Keyes, J. (2004). *Software Configuration Management*. Boca Raton: CRC Press.
- Kruchten, P. (2000). *The Rational Unified Process An Introduction, Second Edition*. Addison Wesley.
- Lewis, W. E. (2009). *Software Testing and Continuous Quality Improvement*. Boca Raton: CRC Press.
- Ministerio de Administraciones Públicas de España. (s.f.). *Dirección General para el Impulso de la Administración Electrónica*. Obtenido de http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es
- Platinum. (1998). Obtenido de www.platinum.com
- Pressman, R. S. (2002). *Ingeniería del Software, un enfoque práctico*. Madrid: Mc Graw Hill.
- Rational the software development company. (1998). Obtenido de www.rational.com
- Software Engineering Standards Committee. (1998). *IEEE Standard for Software*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- Software Engineering Standards Committee. (1998). *IEEE Standard for Software Configuration Management Plans*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- Starrett, E. C. (1998). The Defense Journal of Software Engineering. *CrossTalk* .

SYBASE. (2008). *Sybooks Online*. Recuperado el 2011, de http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dc00844_1150/html/pbug/CHDBFFEJ.htm

Visual SVN. (2005). *VisualSVN - Subversion-based source control for Windows*. Recuperado el 2010, de <http://www.visualsvn.com/server/>

BIOGRAFÍA

Wilson Santiago Paredes Gaibor

Nació el 6 de Abril de 1985 en la ciudad de Ambato, realizó sus estudios primarios hasta tercer año en la Escuela “Juan León Mera” La Salle. Posteriormente se cambió a la Escuela y Colegio Atenas donde finalizó sus estudios primarios, y a la vez su educación secundaria obteniendo el título de Bachiller en ciencias en el año 2003.

Viajó a Inglaterra en Octubre del 2003 por nueve meses, a obtener la suficiencia en Inglés por un año en el Instituto Internacional EF, y rindió de igual manera exámenes de certificación en suficiencia del idioma inglés en la Universidad de Cambridge, ambos obtenidos en el año 2004.

En Octubre del 2004 ingreso a la Escuela Politécnica del Ejército a realizar sus estudios en la carrera de Ingeniería en Sistemas e Informática, egresando en Febrero del 2010.

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Wilson Santiago Paredes Gaibor

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña

Lugar y fecha: _____