UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1

Propuesta para la aplicación de la Gestión de Configuración de Software en el proceso productivo de la UCI

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN CIENCIAS INFORMÁTICAS

AUTOR

Odette Gordillo Vargas

TUTOR

Ing. Ramsés Delgado Martínez

Ciudad de la Habana, junio 2007

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 18 días del mes de junio del año 2007.

_____	_____
Odette Gordillo Vargas	Ramsés Delgado Martínez
(Autora)	(Tutor)

DATOS DE CONTACTO

Ing. Ramsés Delgado Martínez

Especialista Dirección de Calidad

ramsesd@uci.cu

Agradecimientos

Deseo al concluir este trabajo de tesis agradecer a todas las personas que han contribuido en mi formación profesional.
A Yassel porque sin él esta tesis no hubiera sido posible.
A mis abuelos por estar siempre conmigo y ayudarme a alcanzar este sueño.
A mis padres por todo su amor en estos 23 años.
A mi hermana Claudia por querer seguir mis pasos.
A mi tía por ser mi ejemplo y por ayudarme en todo lo que necesité.
A Ernesto por hacerme reír siempre.
A Laura por ser mi hija postiza y quererme tanto.
A Dolmarys y Mary por su amistad durante todos estos años.
A Juan y Sofía por su cariño y todos los consejos que me dieron.
A María y a Delvis por aceptarme y quererme como una hija y hermana.
A Rotny y Mariam por todas las cosas buenas que hemos pasado juntos.
A Lulu y Mirta por brindarme su apoyo y por confiar en mí.
A Eilyn y Elaine porque aunque estén lejos siempre están en mi pensamiento.
A mi tutor por todo su esfuerzo y ayuda.
A todos aquellos que compartieron conmigo un pedacito de su vida y me enseñaron algo.
A Idiaily, Adriana, Yusi, Yusnay, Yayi, Isel, Daily, Yohana, el ruso, Ricardo, Yunior, el chino, fueron buenos momentos.
A todos los profes que durante 5 años nos han trasmitido su conocimiento y ejemplo.
A los que se me olvidan.

MUCHAS GRACIAS

Dedicatoria

A mis padres

A mis abuelos

A Yassel con todo mi amor

Resumen

La Gestión de Configuración de Software (GCS) es una de las actividades claves para que una organización de desarrollo pueda alcanzar el Nivel de Maduración 2 establecido por el Software Engineering Institute (SEI).

Desafortunadamente, la necesidad de implementar estas actividades surge como consecuencia de la aparición de problemas en la calidad de los productos ya desarrollados o en etapas avanzadas del desarrollo, o por dificultades para mantener el ritmo de producción, puesto que en estas circunstancias, el personal asignado a tareas de desarrollo debe también atender pedidos de mantenimiento o mejoramiento de productos finalizados.

Otro factor que contribuye a detectar esta necesidad, es la aparición gradual de programadores, analistas o especialistas, pertenecientes a los grupos de desarrollo, que con el tiempo, se van convirtiendo en “insustituibles”, pues, ante la falta de una documentación completa y coherente, son los únicos que se encuentran en capacidad de implementar cambios en los productos entregados o en etapa avanzada de implementación.

Estas dificultades se corrigen mediante la adopción de las técnicas de Gestión de Configuración que se hallan ampliamente desarrolladas en los libros de texto relacionados con la Ingeniería de Software, pero en ninguno de ellos es posible encontrar una explicación de cómo aplicarlas en productos en proceso de desarrollo o que ya han sido entregados a los clientes.

En tal sentido el objetivo de este trabajo es realizar una propuesta de una práctica que sirva como guía para aplicarla en los proyectos productivos de nuestra universidad.

PALABRAS CLAVES

GCS, Proceso Productivo, Calidad, Ingeniería de Software, Actividades de GCS

ÍNDICE

ÍNDICE

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La época actual conduce a un énfasis marcado en la competitividad, a una fuerte lucha existente por satisfacer las necesidades de los clientes y de la organización. Es en este camino que la innovación tecnológica juega un importante papel. Teniendo en cuenta que existen una serie de tendencias que condicionan el entorno competitivo de hoy en día, y crean un medio comercial internacional que en nada se parece al que existía hace unos pocos años, que la competencia se vuelve impredecible y multifacética, y las ventajas no son capaces de durar, sino que deben regenerarse constantemente, la industria cubana tiene la necesidad de insertarse en ese mundo tecnológico donde el software se ha convertido en un tema crítico en la sociedad moderna mundial. Todos parecen necesitar mejores software en menos tiempo y a menor costo. Sin embargo los métodos de desarrollo de software que se utilizan en muchas organizaciones no son los establecidos, sino que se basan en los métodos que los propios individuos siguen artesanalmente.

Esto es un factor que claramente afecta la calidad de un producto de software, e impide que pueda competir en el mercado, donde la calidad ya no se califica como un elemento más que el producto debe tener, sino que en estos tiempos la calidad se ha convertido en un factor necesario para que una empresa pueda competir y ganar.

Otro elemento a tener en cuenta y que la Industria Cubana del Software (InCuSof) no está ajena a ello, es que existen entre otros, problemas con la productividad de los trabajadores, los tiempos de entrega de los productos y de las documentaciones, la calidad de las pruebas y la falta de comunicación efectiva entre los involucrados (usuarios, desarrolladores, administradores, clientes e investigadores) (MARTÍNEZ and SOCA 2006)

Todos estos problemas se deben a que no se aplican correctamente las técnicas de Ingeniería y Gestión de Software (GS). Esta última tiene como objetivo establecer y mantener la integridad y coherencia del producto software a fin de facilitar el seguimiento de los cambios que sobre él se implementan, asegurando la posibilidad de realizar auditorías de control sobre la evolución de las diferentes configuraciones, en resumen, la

GC identifica los elementos de un proyecto de desarrollo de software (especificaciones, requisitos, arquitecturas, códigos, planes, etc.)

Es por eso que todas las empresas de software cubanas deben corregir todos estos problemas cuanto antes, pues Cuba debe aspirar a abrirse un espacio, aunque sea pequeño, en ese mercado que se ha convertido en un negocio capaz de mover valores monetarios importantes en un solo año.

La Universidad de las Ciencias Informáticas (UCI) es un ejemplo de cómo está evolucionando nuestro país en ese aspecto. Está concebida para formar profesionales y como soporte para la informatización del país, la producción de software y los servicios industriales. Sin embargo tampoco estamos alejados de los problemas existentes. Debemos garantizar la calidad en todos nuestros productos y para eso hay luchar porque se le apliquen las normas y estándares de calidad existentes en el mundo al ser finalmente liberados, para que los mismos puedan competir en el mercado mundial de software.

Como se puede apreciar Cuba no está ajena a muchos de los problemas que existen en el mundo respecto a la calidad y es por eso que para dar solución a estos problemas y lograr el éxito de un proyecto debemos ejecutar correctamente cuatro tipos de funciones (ANTONIO 2001)

- La Gestión de proyecto.
- El desarrollo técnico.
- El Sistema de Calidad, que incluyen las actividades de validación.
- El Sistema de Gestión de Configuración de Software(GCS)

Teniendo en cuenta lo anteriormente planteado, y para hacer más eficiente la producción de software en la UCI se identifica como **Situación Problémica** que en nuestra universidad existen proyectos que no tienen definido que elementos desean controlar. No se realiza correctamente un control de las versiones del software, así como tampoco se

lleva un control adecuado de todos los cambios que ocurren en el mismo. Además no se lleva un control de las líneas bases y no se realizan las auditorías de la configuración. Todos estos problemas están relacionados con la ausencia, entre otros, de modelos de referencia para la introducción y ejecución, de forma iterativa e incremental de los procesos de GCS, dirigidos a mejorar su competitividad en el mercado y disminuir los costos además de que todo lo antes mencionado implica una pérdida de tiempo para el equipo de trabajo a la hora de entregar finalmente el producto y no es posible garantizar que el mismo tenga la calidad requerida.

Después de lo antes expuesto nos planteamos la siguiente interrogante como **Problema de Investigación:**

¿Cómo mejorar las deficiencias que existen en el momento de realizar la Gestión de Configuración en un proyecto?

El **Objeto de Estudio** se enmarca en las prácticas y herramientas existentes de GCS en la UCI, teniendo como **Campo de Acción** los Proyectos Productivos de la Universidad.

Como **Pregunta Científica** se planteó si las prácticas de Gestión de Configuración de Software ayudarían a mejorar la producción de software en la Universidad.

Para dar respuesta al Problema de la Investigación, se definió como **Objetivo General:**

- Proponer prácticas para la aplicación de la gestión de configuración de software en los proyectos productivos de la UCI

Los **Objetivos Específicos** que se desprenden de este trabajo son:

- Especificar el estado de la gestión de configuración de software a nivel mundial
- Caracterizar el estado de la gestión de configuración de software en los proyectos productivos de la UCI.
- Realizar una propuesta de capacitación al personal.

- Diseñar una propuesta de prácticas a seguir.
- Evaluar la propuesta.

Para dar cumplimiento a lo anteriormente planteado se realizaron las siguientes **tareas**.

1. Realizar un estado del arte sobre la gestión de configuración de software.
2. Proponer un proceso elemental de gestión de configuración de software para los proyectos de la universidad.
3. Diseño de un curso para la preparación y capacitación de estudiantes y profesores en los temas de gestión de configuración de software.

Al concluir este trabajo se espera como **Posibles Resultados**.

- Obtención de una propuesta de prácticas para mejorar la gestión de configuración del software en los proyectos productivos de la UCI.
- El diseño de un curso optativo que permita la preparación de estudiantes y profesores en los temas de Gestión de Configuración de Software.

En la elaboración de esta investigación se utilizaron métodos teóricos y empíricos que se resumen en la aplicación de encuestas a diferentes proyectos, profesores, líderes de proyectos y estudiantes, la realización de entrevistas a expertos en el tema. Se hizo además una amplia revisión bibliográfica y se valoraron un conjunto de normas que se aplican en el ámbito nacional como internacional que permitieron analizar los fundamentos teóricos.

Este trabajo cuenta con tres capítulos: una introducción, conclusiones y recomendaciones.

En el **Capítulo 1** se hace un recorrido por el estado del arte sobre los temas de gestión de configuración de software.

En el **Capítulo 2** se hace una descripción de la propuesta, la cual es el objetivo principal de este trabajo, además se realiza la propuesta de asignatura de Introducción a la Gestión de Configuración de Software.

En el **Capítulo 3** se realiza la evaluación de la propuesta y los resultados que arrojó esta.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abordan temas referidos al estudio del estado de la Gestión de Configuración de Software a nivel mundial y el papel que juega dentro del proceso de desarrollo de software. Se aborda la situación de la InCuSof y se exponen algunos conceptos de la calidad del software.

1.2 Situación de la Industria Cubana

Actualmente nuestra industria de software se encuentra en vías de desarrollo, la dirección del país, reconociendo el potencial económico que puede representar dicha industria, ha tomado medidas para el aprendizaje de la informática desde edades tempranas. (MORENO 2002) Por la calidad de nuestros profesionales y el prestigio cosechado en diversas esferas se considera que podemos tener mercado en muchos países de Latinoamérica e incluso de Europa en la prestación de servicios informáticos. A pesar de lo antes expuesto el desarrollo de nuestra industria de software puede considerarse aún discreto. Un estudio de los principales problemas que afectan a la misma, de manera general mostró problemas relacionados con:

- Desorganización en las empresas, que no permite aplicar técnicas, modelos o estándares que ayudarían al desarrollo de esta.
- Pocos clientes, y no existe una disciplina para el intercambio con ellos.
- Resulta muy pobre el mercado externo.
- No se establecen modelos de calidad en las empresas.

El origen de algunos de estos problemas se puede encontrar en la ausencia de la definición de los procesos de la empresa, siguiendo los principios de la ingeniería y la gestión de software. Lo anterior, realizado de forma progresiva en las empresas, permite el aumento de la productividad y de la calidad en el desarrollo de las aplicaciones. (INFORMÁTICA 1996)

1.3 Calidad de Software

Todas las metodologías y herramientas tienen como único fin producir software de gran calidad, para Pressman la calidad es “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. (CUEVAS 1999) Una definición muy sencilla de calidad de software es brindada por (BRADLEY 1987): “…de manera general la calidad es la presencia de características deseables y la ausencia total de características no deseables en un producto o en un proceso.”

Estas características, cuya ausencia o presencia denotan calidad, son completamente dependientes de la situación específica de cada producto y de cada empresa. En esencia, la calidad es relativa. (CROSBY 1979)

Para (PRESSMAN 2001) a la hora de definir quién tiene la responsabilidad de lograr calidad en los productos de software existen tres tendencias:

- La calidad sólo se puede abordar a través de las vías estadísticas.
- Sólo los especialistas de calidad están capacitados para tratar los problemas de la mejora de la calidad.
- La calidad es asunto de todos los que intervienen de alguna manera en la producción de software.

La tendencia actual sobre la calidad en el desarrollo de los productos de software se identifica con el tercer criterio.

De manera general existe consenso de que la calidad redunda en la conformidad que debe tener el usuario con el producto terminado. Además, la empresa en alguna medida debe ser capaz de predecir y medir los aspectos relacionados con este tema ya que la calidad del software no puede ser alcanzada después de terminado el producto. Por esta razón, el aseguramiento de la calidad del software comienza con la aplicación de los mejores métodos, técnicas y herramientas que ayuden a obtener especificaciones de calidad de los proyectos y mejorar estas para obtener un diseño de mayor calidad.

Todo esto requiere el establecimiento del sistema de calidad para la empresa de software.

1.4 Modelos de Calidad

Las empresas de la InCuSof no están ajenas a la tendencia de buscar sistemas de calidad e imponerlos, lo más usual es la aplicación de sistemas de calidad relacionados con el estándar de ISO 9000. Sin embargo, la opinión que se tiene de ellos, y el uso que se les ha dado, ha generado una extensa polémica específicamente en la Industria de Software. Por estas razones han surgido otros esquemas de certificación como el TickIT (Esquema Británico de Certificación) y otros modelos de calidad y mejora de procesos como el ISO/SPICE (Software Process Improvement and Capability Determination) y CMMI (Capability Maturity Model Integration)

1.5 Gestión de configuración de Software

La GCS es una disciplina para manejar la evolución de los productos de software a lo largo de su ciclo de vida, desde las etapas del desarrollo, hasta que el producto sale del ambiente de explotación. (HUMPHREY 2000)

El diccionario de términos IEEE en (PIATINI 1996) define GCS como el proceso de identificar y definir los elementos de configuración en un sistema, controlando la entrega y el cambio de estos elementos a través del ciclo de vida del sistema, almacenando el estado de los elementos de configuración y de las solicitudes de cambio, y verificando la completitud con respecto a los requerimientos especificados.

En resumen la GCS se puede definir como un área de la Ingeniería de Software cuya misión es controlar la evolución de un producto desarrollado, involucrando un conjunto de técnicas para gestionar con eficiencia los cambios que se realicen sobre ese producto a lo largo de su ciclo de vida.

La norma ISO 10007 (ISO-10007) define como el objetivo de las actividades de Gestión de Configuración, establecer y mantener la integridad y coherencia del producto software a fin de facilitar el seguimiento de los cambios que sobre él se implementan, asegurando la posibilidad de realizar auditorías de control sobre la evolución de las diferentes configuraciones.

La Gestión de Configuración, en resumen, identifica los elementos de un proyecto de desarrollo de software (especificaciones, requisitos, arquitecturas, código, planes, etc.) proporcionando el control de los elementos identificados y la generación de informes de estado de la configuración, consiguiendo, al mismo tiempo, claridad de gestión, al asignar responsabilidades al personal encargado de las tareas de control a lo largo del ciclo de vida del producto.

La Gestión de Configuración, no es un aspecto independiente de la Ingeniería de Software (IS), por el contrario, se encuentra fuertemente relacionada con otras áreas, entre las que se destacan (BERLACK 1997):

- El mantenimiento del producto software.
- La calidad del producto.
- El entorno de desarrollo.

- El modelo de proceso.
- La organización que desarrolla el producto.

1.6 Configuración del Software

Durante el ciclo de vida de un producto software se genera información de diferente naturaleza, que abarca desde las especificaciones del sistema hasta el plan de retiro e incluye documentación, archivos, bases de datos, cursos, etc. Todo este conjunto de información producida durante el desarrollo del proyecto recibe el nombre de Configuración del Software.

Cada uno de los componentes de la Configuración del Software recibe el nombre de Elemento de Configuración de Software (ECS). Un ECS debe ser un elemento que se pueda definir y controlar de forma separada. Es decir, debe ser una unidad en sí mismo.(ANTONIO 2001).

Se pueden considerar como ECS, entre otros, a los siguientes componentes:

- Especificaciones del Sistema.
- Estimaciones y Planes.
- Especificación de requisitos software.
- Diseño arquitectónico.
- Diseño detallado.
- Prototipos generados.
- Código fuente.
- Documentación relacionada con la determinación de los factores de riesgo y su gestión a efectos de minimizar sus consecuencias.
- Programas ejecutables y librerías asociadas.
- Manuales del usuario, de operación e instalación.

- Documentación relacionada con cursos de formación en el uso del producto.
- Plan de pruebas.
- Casos de Prueba y resultados obtenidos.
- Estándares y procedimientos de Ingeniería de Software utilizados.
- Informes de incidencia.
- Pedidos de mantenimiento.
- Órdenes de cambio.
- Documentación del Software y Hardware utilizados como herramientas de desarrollo.
- Diseño de bases de datos.
- Bases de Datos.
- Información del entorno de desarrollo y de implantación.
- Contenidos iniciales de las bases de datos.

En cualquier caso, para cada proyecto concreto se debe determinar qué se va a considerar como ECS. (ANTONIO 2001).

1.7 Líneas Base

Por la naturaleza propia del software, un factor que siempre está presente cuando se trabaja con esta clase de productos es la necesidad de realizar cambios. El requerimiento de cambio se presenta en cualquier momento del ciclo de vida y generalmente es justificado, por lo que resulta deseable, o imprescindible en proyectos de gran envergadura, contar con herramientas que permitan realizar un efectivo control y gestión de los cambios. Esa herramienta es la Gestión de Configuración. Con el propósito, entonces, de controlar los cambios se utiliza el concepto de Líneas Base.

Las Líneas Base pueden ser vistas como hitos en el proceso de desarrollo que se constituyen en función de la aprobación de uno o varios ECS mediante la ejecución de

revisiones técnicas formales. La dinámica del procedimiento permite cambios rápidos e informales sobre los ECS antes que estos pasen a formar parte de una Línea Base, pero una vez establecido el hito el cambio sobre el ECS sólo puede ser realizado siguiendo un procedimiento formal y rígido que tiene por finalidad evaluar y verificar cada cambio (RATIONAL 2003a)

El IEEE define una línea base como una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios. (IEEE 1987)

Según (ANTONIO 2001) la idea consiste en permitir cambios rápidos e informales sobre un Elemento de Configuración del Software antes de que se pase a formar parte de una línea base, pero en el momento en que se establece una línea base se debe aplicar un procedimiento formal para evaluar y verificar cada cambio.

1.8 Actividades de la Gestión de Configuración

De acuerdo con lo especificado por la IEEE, la Gestión de Configuración contempla las siguientes actividades (IEEE 1987):

- Identificación de la configuración.
- Control de cambios en la configuración.
- Generación de informes de estado.
- Auditorías de configuración.
- Control de versiones.

1.8.1 Identificación de la configuración

Consiste en identificar la estructura del producto, sus componentes y tipo, haciéndolos únicos y accesibles mediante algún procedimiento.

Esta actividad engloba dentro otras actividades: (ANTONIO 2001)

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

Para controlar y gestionar los ECS se puede seguir un enfoque orientado a objetos. Se pueden identificar dos tipos de objetos: (NAVARRO 2004)

- Objetos básicos.
- Objetos compuestos.

Un objeto básico es una unidad de texto creada por un miembro del equipo durante el proceso de IS, por ejemplo

- El plan del proyecto.
- Una parte del diseño.
- Un listado de código.
- Un conjunto de casos de prueba.

Un *objeto compuesto* es una colección de objetos básicos y de otros objetos compuestos. Conceptualmente se pueden ver como una lista de referencias que identifican los objetos básicos que los componen.

Cada objeto tiene un conjunto de características que lo identifican:

- Nombre.
- Descripción.
- Lista de recursos.
- Lista de realización.

El nombre del objeto es una cadena de caracteres que identifican al objeto sin ambigüedad.

La descripción es una lista que identifica el tipo de ECS, un identificador de proyecto, la información de la versión y/o cambio.

Los recursos son entidades que proporciona, procesa, referencia o son, de alguna otra forma requeridos por el objeto.

La realización es una referencia a la unidad de texto para objetos básicos y nulos para objetos compuestos.(NAVARRO 2004).

Las relaciones que un ECS mantiene con los demás son básicamente de agregación y dependencia.

1.8.2 Control de cambios en la configuración

Consiste en controlar las versiones de un producto y los cambios que sobre él se producen a lo largo de su ciclo de vida.

Por lo general se establecen varios niveles de control de cambios (ANTONIO 2001):

- **Control de cambios informal:** Antes de que el Elemento de Configuración del Software pase a formar parte de una línea base, aquel que haya desarrollado el

Elemento de Configuración del Software podrá realizar cualquier cambio justificado sobre él.

- **Control de cambios al nivel del proyecto o semi-formal:** Una vez que el Elemento de Configuración del Software pasa la revisión técnica formal y se convierte en una línea base, para que el encargado del desarrollo pueda realizar un cambio debe recibir la aprobación de: el director del proyecto, si es un cambio local o del Comité de Control de Cambios (CCC) si el cambio tiene algún impacto sobre otros ECS.
- **Control de cambios formal:** Se suele adoptar una vez que se empieza a comercializar el producto, cuando se transfieren los ECS a la Biblioteca Maestra. Todo cambio deberá ser aprobado por el Comité de Control de Cambios.

En el control de cambios tanto formal como semi-formal, aparece una nueva figura en la Organización, el Comité de Control de Cambios, este es una persona o grupo encargado de tomar las decisiones finales acerca del estado y la prioridad de las peticiones de cambio.

El proceso de control de cambios puede ser considerado como la actividad más trascendente de la Gestión de Configuración, ya que proporciona un mecanismo rígido para el control de los cambios a realizar sobre un producto a lo largo de todo su Ciclo de Vida.

“El control de cambio es vital. Pero las fuerzas que lo hacen necesario también lo hacen molesto. Nos preocupamos por el cambio porque una diminuta perturbación en el código puede crear un gran fallo en el producto. Pero también puede reparar un gran fallo o habilitar excelentes capacidades nuevas. Nos preocupamos por el cambio porque un desarrollador pícaro puede hacer fracasar el proyecto; sin embargo las brillantes ideas nacidas en la mente de estos pícaros, y un pesado proceso de control de cambio pueden disuadirle de hacer un trabajo creativo.” (PRESSMAN 2001).

En el anexo 1 observamos cómo se realiza el proceso de control de cambios.

1.8.3 Generación de Informes de Estado: Consiste en la producción de informes sobre el estado de los ECS de un producto y de las solicitudes de cambio realizadas sobre ellos. Mantiene a los usuarios, a los gestores y a los desarrolladores al tanto del estado de la configuración y su evolución.

Esta actividad implica, por tanto, la realización de tres actividades básicas: (ANTONIO 2001)

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

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

- ¿Qué pasó?
- ¿Quién lo hizo?
- ¿Cuándo pasó?
- ¿Qué más se vio afectado?

La generación de informes de estado de la configuración desempeña un papel vital en el éxito del proyecto de desarrollo de software. Cuando aparece involucrada mucha gente es muy fácil que no exista una buena comunicación. Pueden darse errores entre las personas desarrolladoras del software. (NAVARRO 2004).

1.8.4 Auditoría de Configuración

Consiste en la validación de la integridad de un producto, manteniendo la consistencia entre sus componentes. Una auditoría es una verificación independiente de un trabajo o del resultado de un trabajo o grupo de trabajos para evaluar su conformidad respecto de especificaciones, estándares, u otros criterios. La auditoría de la Configuración es la forma de comprobar que efectivamente el producto que se está construyendo es lo que pretende ser.

Dentro de ella se pueden realizar tres tipos de actividades: (ANTONIO 2001)

- **Revisiones de fase:** Se realizan al finalizar cada fase del desarrollo y su objetivo es examinar los productos de dicha fase. Las revisiones propias de la Gestión de configuración son aquellas en las que se establecerán las líneas base. El objetivo de esta revisión es descubrir problemas, no comprobar que todo está bien. Hay que ser capaz de desenmascarar los problemas ocultos y sutiles, no sólo los que son obvios.
- **Revisiones de cambios:** Se realizan para comprobar que los cambios aprobados sobre una línea base han sido hechos correctamente.
- **Auditorías:** Se realizan al final del proceso de desarrollo de software y su objetivo es examinar el producto en su conjunto.

La identificación, control de versiones y control de cambios promueven un seguimiento hasta la generación de la orden de cambio de ingeniería (OCI). Para asegurarnos que el cambio se ha implementado correctamente debemos realizar primeramente revisiones técnicas formales y auditorías de configuración de software.

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

Una auditoría de configuración del software complementa la revisión técnica formal al comprobar características que generalmente no tiene en cuenta la revisión. La auditoría se plantea y responde con las siguientes preguntas: (PRESSMAN 1997)

- ¿Se ha hecho el cambio especificado en la OCI? ¿Se han incorporado modificaciones adicionales?
- ¿Se ha llevado a cabo una revisión técnica formal para evaluar la corrección técnica?
- ¿Se han seguido adecuadamente los estándares de ingeniería de software?
- ¿Se ha "recalcado" los cambios en el ECS? ¿Se ha especificado la fecha del cambio y el autor? ¿Reflejan los cambios los atributos del objeto de configuración?
- ¿Se han seguido procedimientos del GCS para señalar el cambio, registrarlo y divulgarlo?
- ¿Se han actualizado adecuadamente todos los ECS relacionados?

Sin embargo, otros autores plantean que los sistemas que automatizan la gestión de configuración suelen incluir algunas funciones adicionales, lo que nos lleva a ampliar la definición estándar con las siguientes actividades: (APPLETON and BERCZUK 2002)

- Construcción: Consiste en gestionar la compilación y enlazado de los distintos componentes del producto software de una forma lo más eficiente posible.
- Control del Trabajo en Equipo: Consiste en controlar las interacciones que se producen entre los múltiples desarrolladores de un producto, sobre todo cuando deben compartir ciertos componentes del producto.
- Control de Versiones: Consiste en mantener un registro histórico de las diferentes versiones por las que pasan los componentes de un producto, que permita la recuperación de cualquiera de ellas.

- Gestión de Problemas: Consiste en realizar un seguimiento de la evolución de los problemas que afectan al producto.

1.8.5 Control de versiones

Cuando se finalice un cambio o se realice una modificación, por consecuencia traerá consigo que se haga un cambio de versión del producto de software. Como no se puede definir cuantas versiones podría tener un producto es necesario realizar un control de versiones que según (PRESSMAN 1997) la describe en el contexto de la Gestión de la Configuración "La gestión de configuración permite a un usuario especificar configuraciones alternativas del sistema software mediante la selección de versiones adecuadas. Esto se puede gestionar asociando atributos a cada versión del software y permitiendo luego especificar y construir una configuración describiendo el conjunto de atributos deseado."

1.9 Herramientas CASE (Computer Aided Software Engineering).

Como se había mencionado, actualmente la tendencia es hacia un mundo heterogéneo en el cual convivan diversos productos que se complementen y en ese contexto contar con herramientas de desarrollo abiertas con conectividad a diversas plataformas, basadas en tecnología orientada a objetos y que permitan la reutilización del software. De este modo, la mayoría de las empresas se han extendido a la adquisición de herramientas CASE (*Computer Aided Software Engineering*, Ingeniería Asistida por Computadora) con el fin de automatizar los aspectos clave de todo lo que implica el proceso de desarrollo de un sistema e incrementar su posición en el mercado competitivo. Sin embargo, en algunos se obtienen elevados costos tanto en la adquisición de herramientas y costos de entrenamiento de personal, como a la falta de adaptación de tal herramienta a la arquitectura de la información y a metodologías de desarrollo utilizadas por la

organización. Con el objetivo de identificar rápidamente a todos los ECS de una determinada versión, existen herramientas CASE que permiten automatizar la gestión de configuración de un proyecto en lo que respecta a información de archivos, programas, diagrama de flujo de datos, bases de datos, etc. pero no contemplan la información volcada en la documentación.

El proceso de desarrollo de un producto software a menudo es largo y no menos complicado. A través de éste se deben realizar varias actividades como planificación, modelado, construcción, revisiones, actividades relacionadas con la GCS, entre otras. La realización de las mismas de manera manual atenta contra la eficiencia y productividad del equipo de desarrollo, e incluso puede llegar al límite de lo imposible si se trata de equipos muy grandes y/o geográficamente dispersos.(MARTÍNEZ and SOCA 2006). Desde el inicio de la escritura de software, ha existido un conocimiento de la necesidad de herramientas automatizadas para ayudar al diseñador del software. Inicialmente, la concentración estaba en herramientas de apoyo a programas como traductores, recopiladores, ensambladores, procesadores de macros, y montadores y cargadores. Este conjunto de aplicaciones que pueden informatizarse, aumentó dramáticamente en un breve espacio de tiempo, causando una gran demanda por nuevo software a desarrollar.

A medida que se escribía un nuevo software, había ya en existencia millones y millones de líneas de código que necesitaban ser mantenidas y actualizadas. Esto causó a la industria de las computadoras muchos problemas, no podía cubrir el incremento de la demanda con los métodos que se estaban usando. Esto fue reconocido como una crisis de software, es por eso que para dar solución a estos problemas se desarrollaron las llamadas herramientas CASE (Computer Aided Software Engineering) con el objetivo de apoyar a las personas en el desarrollo del software.

Una de las posibles definiciones de herramientas CASE es brindada por el Instituto de Ingeniería de Software (SEI 2007) la cual plantea de forma general que las herramientas CASE brindan soporte al proceso de desarrollo basado en el uso del ordenador.

Otra definición más detallada plantea que las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (WIKIPEDIA 2007)

1.9.1 Clasificación de las Herramientas CASE

Según (WIKIPEDIA 2007) no existe una única clasificación de las herramientas CASE, pero podrían clasificarse de acuerdo con los parámetros siguientes:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- La arquitectura de las aplicaciones que producen.
- Su funcionalidad.

Sin embargo la clasificación más habitual está basada en las fases del ciclo de desarrollo que cubren:

Herramientas de Alto Nivel, U-CASE (Upper CASE o CASE Superior) también conocidas como front-end, están orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo, (análisis y diseño).

Herramientas de Bajo Nivel, L-CASE (Lower CASE o CASE inferior) también son conocidas como back-end y están orientadas a las últimas fases del desarrollo (construcción e implementación).

Juegos de herramientas o Tools-Case, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida de desarrollo. Son también llamadas CASE workbench. Estas herramientas poseen como ventaja que son capaces de integrar el ciclo de vida, además permiten lograr importantes mejoras de productividad a mediano plazo así como un eficiente soporte al mantenimiento de sistemas. Las herramientas I-CASE se basan en una metodología, poseen un repositorio y aportan técnicas estructuradas para todas las fases del ciclo de vida.

1.9.2 Herramientas para la Gestión de Configuración de Software

La GCS se encuentra en el núcleo de todos los entornos CASE. Las herramientas pueden ofrecer su asistencia en las cinco tareas principales de GCS: identificación, control de versiones control de cambios, auditorías y contabilidad de estados. La base de datos CASE proporciona un mecanismo para identificar todos los elementos de configuración y relacionarlo con otros elementos; un acceso sencillo a los elementos de configuración individuales facilita el proceso de auditoría; las herramientas de comunicación CASE pueden mejorar enormemente la contabilidad de estados (ofreciendo información acerca de los cambios a todos aquellos que necesiten conocerlos). (ASENSIO 2007).

1.9.2.1 IBM Rational ClearCase

IBM Rational ClearCase ofrece una gestión fiable, ampliable y flexible de los activos de software para equipos de desarrollo de gran tamaño y tamaño medio. (IBM 2007)

Dentro de las actividades que realiza se encuentran:

- Proporciona una gestión del ciclo de vida y control de los activos de desarrollo de software. Con un control integrado de versiones, una gestión del espacio de trabajo automatizado, un soporte de desarrollo en paralelo y una gestión de línea base.
- Se integra con los entornos de desarrollo (IDE) líder, incluido Rational Application Developer, WebSphere Studio, Microsoft Visual Studio .NET y la infraestructura Eclipse de código abierto agiliza aún más el desarrollo.
- Permite el acceso prácticamente a todos los sitios debido a interfaces locales, remotas y web
- Se integra sin fisuras con Rational ClearQuest para obtener una solución completa de GCS.

Después de haber observado todo lo planteado anteriormente cabe señalar que una de las dificultades de esta herramienta es el elevado costo que posee en el mercado además su nivel de aprendizaje no es nada fácil, por lo que se requiere de tiempo y esfuerzo para aprenderla. Esta herramienta cuesta alrededor de 4311.00 dólares sin los impuestos incluidos.

1.9.2.2 Rational ClearCase Change Management Solution

IBM Rational ClearCase Change Management Solution proporciona una gestión integrada de configuración del software para equipos de desarrollo de gran tamaño y de tamaño medio. (IBM 2007 a).

Dentro de las actividades que realiza se encuentran:

- Automatización y refuerzo del proceso de desarrollo para obtener una mejor eficacia y capacidad de respuesta.
- Gestión integrada de activos de software, gestión de flujos de trabajo y rastreo de defectos y cambios que mejoran la colaboración y modernización del ciclo de vida de aplicaciones.
- Desarrollo paralelo, tiempos reducidos de los ciclos de build/release y una mayor reutilización del software que ayudan a aumentar la productividad.
- Proporciona un seguimiento de comprobación de cambios que hace que sea más fácil la conformidad de normativas.
- Soporta Linux (Intel y zSeries), Windows, Unix y z/OS que permiten una gestión coordinada de los cambios de software entre entornos de mainframe y distribuidos.

Al igual que Racional ClearCase posee un elevado costo, alrededor de 6887.00 dólares (IBM 2007 a).

1.9.2.3 IBM Rational ClearQuest

IBM Rational ClearQuest proporciona un seguimiento flexible de defectos y cambios en toda la empresa. (IBM 2007 b)

Entre las actividades que realiza están:

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

Esta herramienta tiene como una de sus principales ventajas su integración con el resto de las herramientas de la Suite de Racional (MARTÍNEZ and SOCA 2006) al igual que las herramientas anteriores posee un elevado costo alrededor de 4737.00 dólares

1.9.2.4 Microsoft Visual SourceSafe. VSS

Visual SourceSafe (VSS) es el sistema de control de versiones y administración de código fuente que provee la familia de productos Microsoft Visual Studio. Un sistema de control de versiones permite, básicamente, administrar los programas fuentes dentro de un grupo de desarrollo, manteniendo toda la historia de las modificaciones de cada componente, controlando el acceso a estos de cada desarrollador. (MSDN 2007) MS Visual SourceSafe permite:

- Recuperar versiones anteriores.
- Cumple la función de base de datos central, donde se registra qué usuario tiene un programa, form, reporte, etc. tomado para su modificación y no permite (salvo que se especifique) a los otros usuarios que lo modifiquen hasta que quien lo tenga tomado lo libere.
- Sirve para administrar proyectos conteniendo distintos tipo de archivos como son archivos de texto, archivos gráficos, archivos binarios, etc.
- Compartir un archivo entre distintos proyectos, el cual será una versión única y cuando se modifique algo para un proyecto determinado, el cambio se propagará sobre todos los que compartan el archivo.
- Cuenta con las funcionalidades básicas de todo gestor de versiones (última versión, “check In”, “check out”).

MS Visual SourceSafe no presenta un coste elevado, sin embargo por ser un producto que pertenece a Microsoft que es una empresa estadounidense no puede ser comercializado con Cuba.(MARTÍNEZ and SOCA 2006)

1.9.2.5 Subversion

Subversion es un controlador de versiones empleado en la administración de archivos utilizados en el desarrollo de software. No es una herramienta propietaria, es decir, está clasificada como una herramienta libre “open-source.” Está pensado para que varias personas puedan trabajar en un mismo proyecto sin pisarse unos a otros el trabajo, además posee historial de revisiones, tanto por seguridad como para poder recuperar una versión determinada de nuestro código, también permite gestionar versiones de la aplicación, derivaciones y demás. En la actualidad es usado ampliamente por los desarrolladores mantener el registro de todo el trabajo y los cambios en la implementación de un proyecto (construcción de un software) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren en él.

Existen varias interfaces a Subversion, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo. La interfaz más popular en este sistema operativo es el TortoiseSVN, que se integra con el explorador de Windows (WIKIPEDIA 2007)

1.10 La Gestión de Configuración de Software en el mundo

La GCS en el mundo actualmente es un tema que generalmente está siendo tratado en cursos, maestrías y postgrados. Estos cursos no tienen una larga duración pero sí abarcan una gran cantidad de contenido y temas como son:

En el área de Calidad del proceso de Software se abordan temáticas relacionadas con el entorno de desarrollo del Software, el proceso de desarrollo del software, la calidad del proceso de software y los restos y oportunidades que se presentan en la Industria del software. (RIVERA 2003).

Con respecto al área de Gestión de Configuración generalmente primero dan una introducción a este tema donde la abordan de forma más específica y se dan muchos

conceptos básicos y se realiza un estudio por lo general de las herramientas que más se utilizan en cuanto a la GCS.

Después de haber hecho un estudio más detallado respecto a los puntos anteriores se entra al estudio del tema referente a la organización de la Configuración de Software guiándose generalmente por estándares y metodologías que ya son conocidas en el mercado del software. (CASALLAS 2004).

Cuando se concluyen estos temas, abordan entonces temas referidos a las disciplinas y áreas complementarias de la GCS que abarcan dentro de ellas las verificaciones y validaciones que se realizan, la gestión del conocimiento en el proceso de desarrollo de software y los modelos internacionales de calidad para el aseguramiento de la calidad de software. (NAVARRO 2004).

Una vez terminado este estudio se pudo observar como la GCS es tratada en muchas partes del mundo, sin embargo pienso que este es un tema de mucha importancia y cabe señalar que se tiene la tendencia de enseñarla en postgrados, sin embargo esta práctica no es muy factible para nuestra universidad ya que la UCI está directamente vinculada con la producción y si se prepara a un estudiante que desde que se inicia forma parte activa de un proyecto de software y no se le transmite todo el conocimiento necesario referente a la GCS y su relación con la calidad, sino que solo se le enseñan partes muy pequeñas dentro de tanto contenido esto a la larga influirá negativamente tanto en el producto terminado como en lo que respecta a la organización y documentación de un proyecto productivo.

1.11 Conclusiones

A lo largo de este capítulo hemos podido observar cómo se desenvuelve la gestión de configuración de software en el mundo así como su importancia y el significado que tiene en el proceso de desarrollo del software. Por el valor que tiene es una disciplina que se

encuentra dentro de los estándares internacionales y modelos como son ISO, SPICE, CMMI entre otros y es por eso que se dice que es esencial para el éxito de un proyecto de software.

CAPÍTULO 2 CARACTERIZACIÓN Y DISEÑO DEL PROCESO DE GCS A APLICAR.

En este capítulo se caracterizará el proceso de GCS dentro de la Universidad a través de una encuesta realizada diferentes líderes y estudiantes y se dará a conocer una propuesta de una buena práctica de GCS así como una propuesta de capacitación del personal.

2.1 Caracterización del proceso de GCS en la UCI

Con el objetivo de ver cuál es el estado actual de la Gestión de Configuración dentro de la Universidad se realizó la siguiente encuesta a líderes y estudiantes para una población total de 200 encuestados:

1. **¿Conoce usted las actividades de la Gestión de Configuración de Software (GCS)?**

   En esta pregunta solo un 47% de los encuestados contestó positivamente, sin embargo predominó el otro 53% que planteaba que no las conocía o que no sabía siquiera de que le estaban preguntando.

2. **¿En su proyecto se identifican cuáles son elementos de configuración?**

   En esta pregunta solo un 38% contestó positivamente y el otro 62% contestó que no se realizaba identificación alguna.

3. **¿En su proyecto se realiza control de versiones?**

   En esta pregunta el 60% de los encuestados contestó que sí se realizaba control de versiones.

4. **¿En su proyecto se realizan auditorías de la configuración?**

   Aquí el 41% de los encuestados contestó negativamente y el otro 42 % contestó que no se realizaban dichas auditorías porque no conocían exactamente en qué consistía.

5. **¿En su proyecto se lleva un procedimiento para realizar el control de cambios?**

   En esta pregunta un 74% de los encuestados respondió que no se realizaba procedimiento alguno para controlar los cambios, sin embargo el otro 26% contestó acertadamente a esta pregunta.

6. **¿En su proyecto se informan los cambios al personal del proyecto?**

   Aquí el 73% de los encuestados respondió que los cambios que se realizaban sí eran informados.

7. **Diga qué herramientas se utilizan en su proyecto para el control de versiones.**

   En esta pregunta predominó que en nuestra universidad se utiliza el Subversion como herramienta para controlar las versiones, sin embargo en dependencia del tipo de proyecto que sea no se utiliza la misma herramienta cliente.

8. **¿En su proyecto se definen las líneas bases?**

   Aquí un 84% de los encuestados respondió que no a esta pregunta.

9. **¿En su proyecto se capacita al personal en el sentido de la GCS?**

   En esta pregunta el 78% de los encuestados respondió negativamente, el otro 22% contestó que no era una capacitación muy buena pero se les enseñaba algo.

10. **¿En su proyecto se utilizan correctamente las bibliotecas?**

    En esta pregunta el 66% de los encuestados plantea que no se utilizan correctamente las bibliotecas.

11. ¿En su proyecto se controla el estado del proyecto?

En esta pregunta el 73% de los encuestados contestó positivamente

12. ¿En su proyecto se sigue un procedimiento para realizar la solicitud de cambio?

En esta pregunta predominaron las respuestas negativas ya que el 81% de los encuestados plantea que solamente realizan reuniones y si es necesario realizar un cambio lo notifican en la misma y se pasa directamente a realizarlo, por lo que se puede observar que los pasos que se deben seguir para esto (Fig. 2.1) no se están cumpliendo.

13. ¿En su proyecto se almacena dicha solicitud de cambio?

En esta pregunta la gran mayoría respondió que la solicitud no se archivaba en papel por lo que no era posible almacenarla. Solamente contestaron positivamente un 17% de los encuestados.

14. ¿En su proyecto se revisa la solicitud de cambio?

Esta pregunta como depende de los antes mencionado al igual que arriba sólo un 17% contestó positivamente.

15. ¿En su proyecto se realizan informes de estado?

En esta pregunta el 73% contestó que no se realizaban informes.

16. ¿En su proyecto existe un comité de control de cambio?

En esta pregunta el 87% de los encuestados contestó que no existía un comité de control como tal en la mayoría de las ocasiones era el líder el que autorizaba o no a realizar dicho cambio.

En el anexo 4 se muestra una gráfica con las preguntas con valores positivos y negativos, donde se evidencian mejor estos resultados.

2.2 Propuesta de una Práctica de Gestión de Configuración de Software.

Una vez terminado el estudio sobre la aplicación e importancia de la GCS se evidenció que no existe un procedimiento definido para la realización de esta actividad, es por eso que tomando como base los diferentes estándares que se estudiaron se realiza la siguiente propuesta de una buena práctica a seguir, además siempre se deberá trabajar con los documentos y plantillas que fueron establecidos en la universidad. La propuesta se apoya además en el modelo de madurez de las capacidades CMMI para el desarrollo de los productos y los servicios que consta de las mejores prácticas que se ocupan del desarrollo y el mantenimiento de las actividades que cubren el ciclo de vida del producto desde la fase de inicio hasta el mantenimiento.

En la fig. 2.1 se muestran cuales son los flujos de trabajo establecidos y las actividades a realizar para dar cumplimiento a los mismos.

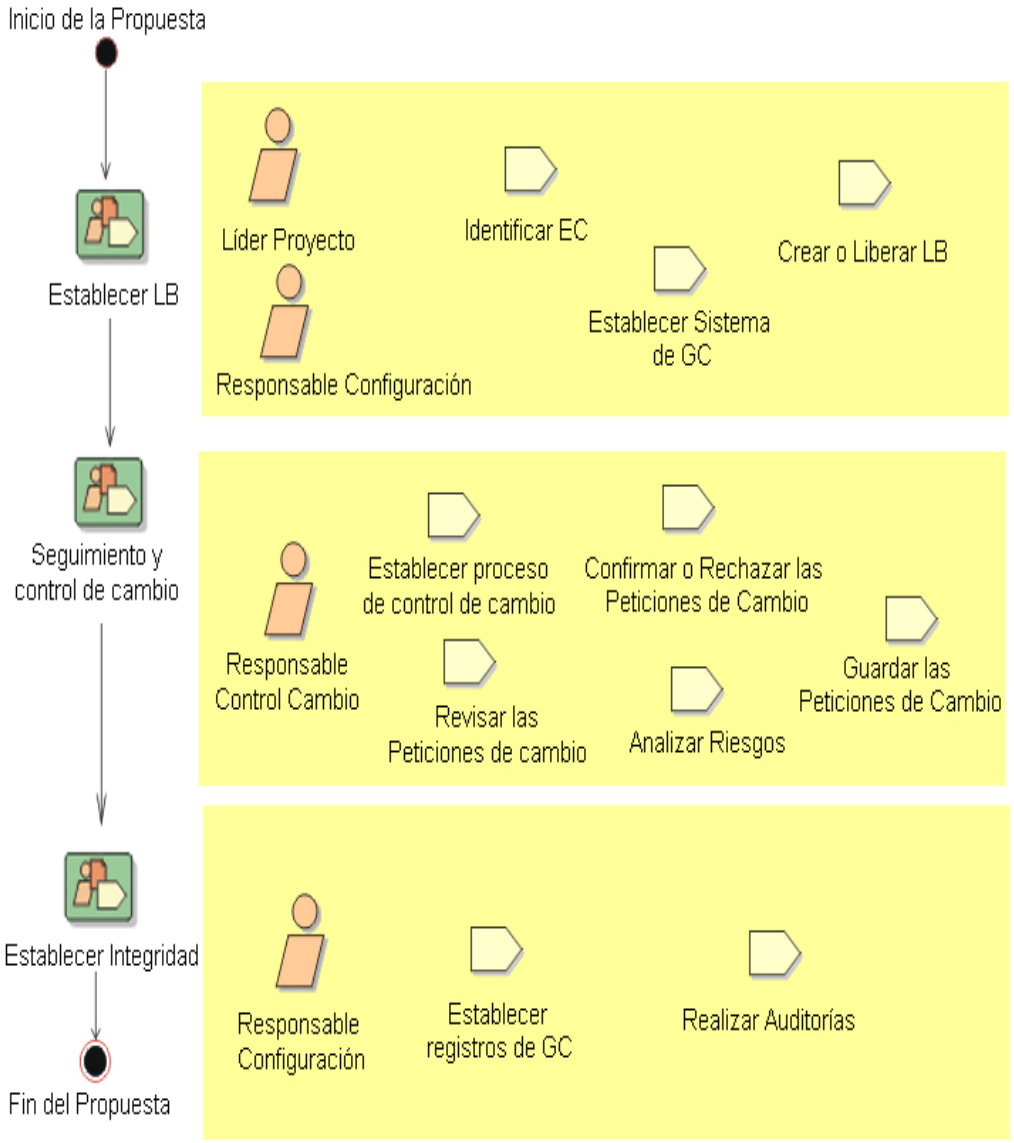


Fig.2.1 Propuesta para realizar la Gestión de Configuración

A continuación se dará una explicación de cada unos de los flujos de trabajo en los que se basa la propuesta y las actividades que lo componen.

2.2.1 Flujo de trabajo: Establecer líneas bases

Las líneas bases van a ser una instantánea de cómo se encuentra la configuración en un momento determinado del ciclo de vida del software y además pueden servir como base para el desarrollo futuro. En la fig.2.2 se observan todas las actividades que se desarrollarán dentro de este flujo.

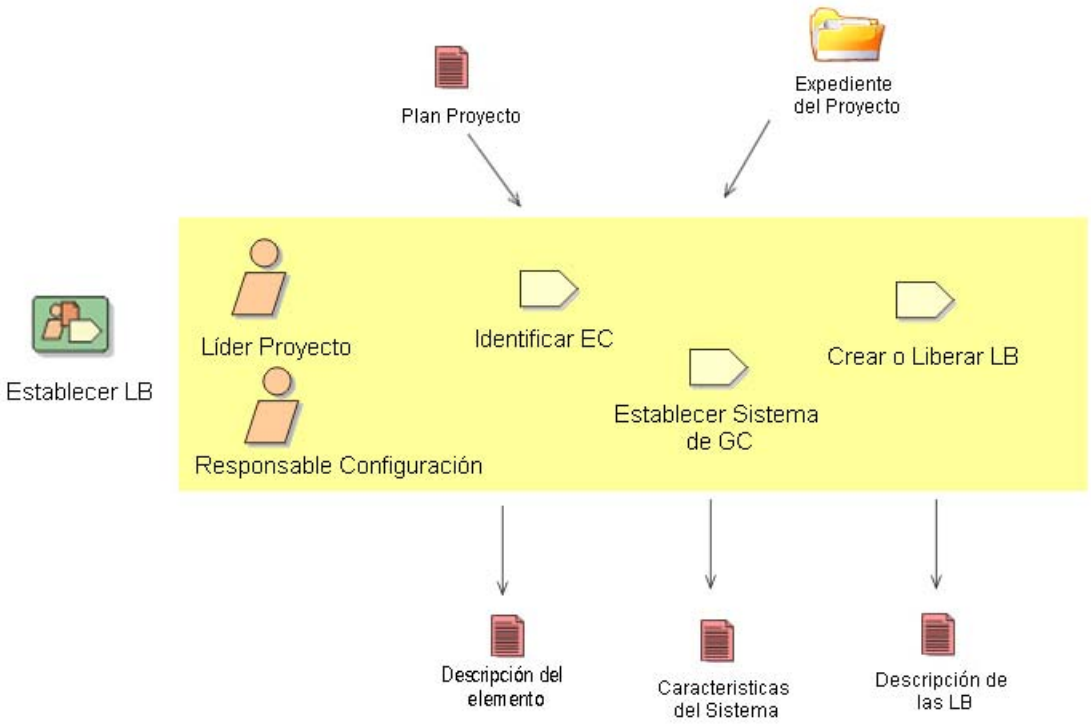


Fig. 2.2 : FT Establecimiento de las Líneas Bases

2.2.1.1 Identificar los elementos de configuración que se desean poner bajo GC

Un elemento de configuración es la unidad básica de control dentro de la GCS, es por eso que antes de comenzar primeramente se debe tener bien claro todo lo que se desee controlar ya sea documentación, ficheros de especificaciones, modelos, ficheros fuentes, scripts de instalación entre otros, es decir, todo lo que resulte de interés para el proyecto se adiciona y posteriormente se reconoce como un elemento de configuración de software y el mismo puede ser objeto de actividades como el control de cambios y el control de versiones.

En la figura anterior se observa un documento que hace referencia a la descripción del elemento, el mismo tendrá en su contenido los puntos que a continuación aparecen.

2.2.1.1.1 Asignarle un identificador único a cada elemento

Cada elemento debe ser único y accesible en algún momento determinado por lo que su identificación se debe realizar de forma atómica es decir no deben existir dos elementos iguales. Se puede seguir el siguiente esquema:

< Número del elemento>

< Nombre del elemento>

< Tipo de elemento> (documento, programa, elemento físico)

< Quién lo hizo> (Va a ser la persona responsable de ese elemento)

< Número de versión a que pertenece el elemento>

< Fecha en que se crea el elemento>

< Si es significativo> (si se va a poner bajo GC)

2.2.1.1.2 Realizar una descripción del elemento

Esta descripción se realizará en un documento que va a contener las características más importantes del elemento en cuestión es decir, quién lo hizo, tipo de elemento (si es un documento, un programa, un elemento físico, etc.) lenguaje de programación, etc.

2.2.1.1.3 Especificar cuando un elemento va a ser puesto bajo GC

Se debe tener en cuenta el estado en que se encuentra el ciclo de vida del proyecto, además, se debe decidir cuándo el elemento estará listo para ser probado y cuál será el grado de control que se desea se tenga sobre el mismo, asimismo el costo y las limitaciones que pueda traer consigo y se deben tener en cuenta los requerimientos del usuario.

2.2.1.1.4 Definir la persona responsable por el elemento

Esta persona es la que responderá por el elemento en cuestión en caso de existir algún problema con el mismo o sólo a modo de información.

2.2.1.2 Establecer un Sistema de Gestión de Configuración

Un sistema de gestión de configuración no es más que la utilización de una herramienta como apoyo para realizar las actividades propias de la gestión de configuración.

En epígrafes anteriores vimos que existe una gran diversidad de herramientas sin embargo es difícil encontrar una herramienta específica que sea capaz de realizar todas las actividades de la Gestión de Configuración es por eso que se propone que se emplee como herramienta para el control de versiones “Subversion” no solo por ser una

herramienta de software libre sino que cumple con las especificaciones que se darán posteriormente.

En la figura 2 se observa un documento que hace referencia a las Características del Sistema, el mismo tendrá en su contenido los puntos que a continuación aparecen.

2.2.1.2.1 Establecer mecanismos para gestionar niveles de control múltiples

Los niveles de control se seleccionaran en dependencia de los objetivos de cada proyecto, los riesgos que puedan existir y los recursos del mismo. Estos niveles de control pueden variar en relación al ciclo de vida del proyecto, tipo de sistema que se esté desarrollando los requerimientos específicos del proyecto.

Los niveles de control pueden ir desde control informal que simplemente sigue los cambios producidos cuando un elemento de configuración está siendo desarrollado, hasta un control formal mediante las líneas bases que solo pueden ser cambiadas como parte de un proceso formal de gestión de configuración.

Mediante el empleo de la herramienta anteriormente mencionada es posible almacenar y recuperar los elementos de configuración. Además nos permite compartir y transferir los elementos entre los diferentes niveles de control que se establezcan en el proyecto. Igualmente las versiones archivadas de los elementos pueden ser recuperadas en caso de existir algún daño.

Subversion es un sistema de “open/source” (código abierto), que se encarga de manejar archivos y directorios, los cuales son colocados en un repositorio central. El repositorio es un servidor común, solo que recuerda los cambios que alguna vez fueron realizados en los archivos y directorios. Esto permite recuperar las versiones más viejas de los datos o revisar el historial de todos los cambios que ocurrieron con un determinado elemento. Con respecto a esto, muchas personas piensan en un sistema de control de versiones como una “máquina del tiempo”.

Muchos sistemas controladores de versiones sólo dan permiso para cambiar un archivo a la vez emplean el sistema bloquea- modifica- desbloquea es decir, si existe una persona Y realizando un cambio en un elemento X este no podrá ser accedido para realizar otro cambio por otra persona Z hasta que la persona Y no termine de realizar sus modificaciones y finalmente haya liberado el elemento. La persona Z solamente tendrá permisos para leer el archivo y esperar hasta que se terminen los cambios y sea liberado el elemento.

El problema de este tipo de sistema es que es un poco restrictivo y a menudo se convierte en un bloqueo de caminos para usuarios.

Para dar solución a este problema y una de las características principales por lo que proponemos la herramienta es que Subversion utiliza un sistema de copia- modifica- sube solución como alternativa a la hora de bloquear un archivo. Con este modelo cada integrante del proyecto puede realizar una copia de los archivos, trabajar paralelamente modificando sus copias privadas modificarlos independientemente unos de otros y posteriormente subir las soluciones.

2.2.1.3 Crear o liberar las líneas bases

Antes de crear o liberar las líneas bases se deberá obtener autorización del Comité de Control de Cambios. Además todos los elementos de configuración que estén contenidos dentro de la misma se deberán documentar y no se debe permitir el acceso a las actualizaciones que ocurran de una línea base.
Se pueden crear diferentes tipos de líneas bases:

Para uso interno:
Una línea base es un conjunto de especificaciones o producto de trabajo que ha sido formalmente revisado y se ha aprobado, que sirve como base para promocionar el

desarrollo, y solo puede ser cambiada mediante procedimientos formales de control de cambio. Una línea base representa la asignación de un identificador para un elemento de configuración o una colección de los mismos y las entidades asociadas. Como un producto evoluciona varias líneas bases pueden ser usadas para controlar su desarrollo y prueba.

Para sistemas de ingeniería

Un conjunto común de líneas bases incluye los requisitos nivelados en sistema, los requisitos del diseño del nivel de elemento de sistema, y la definición del producto al final del desarrollo / comienzo de producción. Estas son típicamente llamadas como “línea base funcional”, “línea base ubicada” y la “línea base del producto”.

Para Software de ingeniería

Una línea base de software puede ser un grupo de requisitos, diseño, archivos de código fuente y el código ejecutable asociado, puede construir archivos, y documentación del usuario (las entidades asociadas) que ha sido asignada a un identificador único.

2.2.2 Realizar un seguimiento y control de cambios

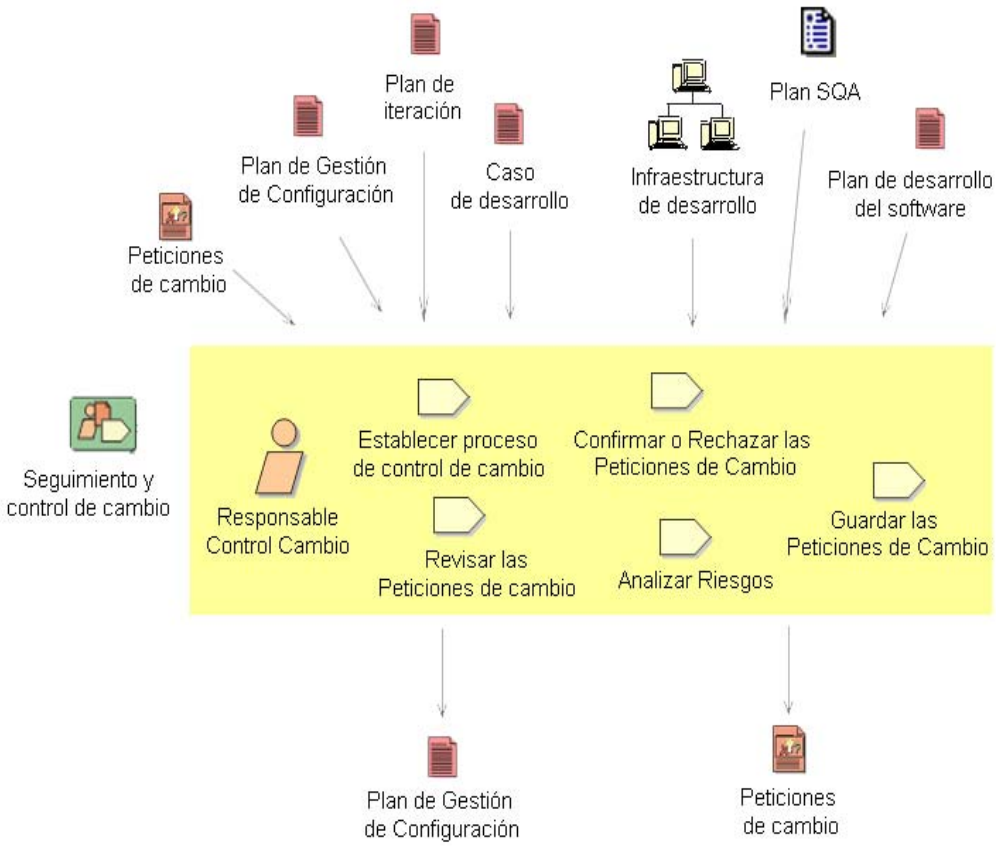


Fig. 2.3: FT Seguimiento y Control de Cambios

Las actividades que se mencionan anteriormente se encuentran implícitas dentro de lo siguiente:

2.2.2.1 Seguimiento de los cambios

Se debe seguir todas las peticiones de cambio que ocurran. Estas peticiones no sólo se ocupan de los cambios nuevos, sino también de las fallas y defectos que se puedan encontrar. Estas peticiones deben ser analizadas para determinar qué impacto pueda traer consigo el cambio, ya sea tanto en el producto de trabajo como tal, en el presupuesto o en las tareas que ya han sido asignadas. Además se deben guardar en una base de datos del mismo nombre. Al analizar el impacto se debe fijar una propuesta y todas las peticiones que serán puestas en la próxima línea base deben ser revisadas con los stakeholders.

2.2.2.2 Control de los cambios

El control se va a mantener sobre la línea base y va a incluir un seguimiento a cada elemento de configuración, aprobando una configuración nueva si es necesario y manteniendo actualizada la línea base. Es bueno aclarar que este control se va a mantener a todo lo largo del ciclo de vida del producto y se deben obtener los permisos necesarios antes de realizar algún cambio. Además se debe chequear dentro y fuera del sistema de gestión de configuración los elementos para la incorporación de los cambios de manera que se mantenga la integridad de dichos elementos. Asimismo se debe asegurar que los cambios no han causado efectos no intencionados en las líneas bases (ej. Asegurar que los cambios no han comprometido la integridad y la seguridad del sistema) al terminar todo esto se deben guardar los cambios que se realizaron en cada elemento y las razones que se tuvieron para dicho objetivo.

2.2.2.3 Artefactos de entrada y salida

2.2.2.3.1 Plan de Gestión de Configuración

El plan de gestión de configuración describe todas las configuraciones y actividades de gestión de control de cambio que se realizarán durante todo el ciclo de vida del producto. Detalla la planificación de actividades, la asignación de responsabilidades, recursos requeridos, incluyendo personal, herramientas, etc.

El propósito del plan de gestión de configuración es definir o referenciar los pasos y las actividades que describen como se debe realizar la configuración y el control de los cambios durante el desarrollo del producto software. El plan debe escribirse al principio de la fase de elaboración, y necesita ser archivado de forma tal que esté disponible para las actividades de mantenimiento de postdesarrollo particularmente para la guía en dónde ciertos activos del software podrían guardarse.

2.2.2.3.2 Caso de desarrollo

El caso de desarrollo describe el proceso de desarrollo, que se ha decidido seguir en el proyecto.

El propósito del caso de desarrollo es capturar el proceso adaptado para el proyecto individual. Sirve como un clasificador para el proceso de desarrollo arreglado para un proyecto o una organización.

Es creado a comienzos de la fase de inicio y se actualiza durante todo el proyecto. Al principio sólo se creará una pequeña versión del caso de desarrollo y a medida que avance el proyecto este se irá incrementando hasta cubrir la mayoría de las disciplinas en cada iteración.

2.2.2.3.3 Infraestructura de desarrollo

La infraestructura de desarrollo incluye hardware y software (computadoras y sistemas operativos) en las cuales las herramientas corren. Además incluye hardware y software usado para interconectar computadoras y usuarios.

Una infraestructura estándar de desarrollo existe para posibilitar que el esfuerzo de desarrollo tenga lugar.

La infraestructura de desarrollo es necesaria para soportar el auge del proyecto por lo que debe ser establecida en las etapas tempranas del ciclo de vida del mismo.

2.2.2.3.4 Plan de desarrollo del software

El plan de desarrollo de software es un artefacto exhaustivo y compuesto que reúne toda la información requerida para dirigir el proyecto. Adjunta varios artefactos desarrollados durante la fase de inicio y mantenido durante todo el proyecto.

Se divide en tres partes fundamentales:

Visión del proyecto: con propósito, objetivos y entregables.

Organización del proyecto: posee estructura, roles y responsabilidades.

Administración del proceso: donde se hacen estimaciones del proyecto, plan de proyecto y plan de iteraciones.

Es una guía para la administración del proyecto y sus actividades y en el mismo se definen las actividades que se asignan a responsables que deben dar cumplimiento en el término de fechas establecidas dentro de un cronograma.

El propósito del plan de desarrollo de software es recoger toda la información necesaria para controlar el proyecto. Describe el enfoque para el desarrollo del software y es el plan del más alto nivel generado, es usado por los líderes y directores para dirigir el esfuerzo

de desarrollo. Desarrollado durante la fase de inicio y es actualizado en cada acontecimiento importante que ocurra.

2.2.2.3.5 Peticiones de cambio

Los cambios para artefactos de desarrollo son propuestos a través de solicitudes de cambio. Las solicitudes de cambio son usadas para documentar y estar al día con los defectos, las peticiones de mejora y cualquier otro tipo de solicitud para un cambio en el producto. La ventaja que tienen las peticiones de cambio es que proveen un registro de las decisiones y aseguran que el impacto del cambio sea comprendido en el proyecto, debido a su proceso de valoración.

La necesidad para el cambio es inherente a desarrollar un sistema de software cuando se desarrolla durante su creación inicial y cuando es usado posteriormente y mantenido día a día funcionando en un ambiente vivo. Generar y dirigir los pedidos de cambio asegura que los cambios para un sistema son hechos de una manera controlada con el propósito de que su efecto sobre el sistema pueda ser pronosticado. Los pedidos de cambios pueden ser usados para documentar y seguir toda clase de pedidos de cambios en el sistema, además incluye peticiones de mejora y los defectos. Las peticiones de mejora incluyen las futuras características para incluir el producto.

Los defectos son informes de las anomalías o los fracasos en un producto de trabajo repartido. Los defectos incluyen tales cosas como las omisiones y los fallos encontrados o síntomas de defectos (fracasos) que tenían que ser aislados y corregidos dentro del software durante las primeras etapas del ciclo de vida. Los defectos también podrían incluir las desviaciones de lo que puede ser esperado razonablemente del comportamiento de software.

2.2.2.3.6 Plan de iteración

Es un grupo de secuencias ordenadas de actividades y tareas. Contiene dependencia de tareas para cada iteración, es un plan muy detallado.

Sirve para que el director del proyecto o líder pueda planear las tareas y actividades en cada iteración, los recursos necesarios y dar seguimiento al progreso del proyecto contra tiempo. Para el equipo de desarrollo le sirve como guía para entender que es lo que necesitan hacer, cuando necesitan hacerlo y que otras actividades son dependientes.

2.2.2.3.7 Plan de aseguramiento de la calidad

El plan de aseguramiento de calidad ofrece una visión clara de como el producto, sus artefactos y la calidad del proceso deben ser asegurados. Contiene revisiones y planes de auditorías y establece referencias para un número de otros artefactos desarrollados durante la fase de inicio. Se mantiene durante todo el proyecto.

Tiene como propósito suministrar un solo punto de referencia sobre el tema de calidad para el proyecto. Es un artefacto orientado a procesos y resalta los elementos de RUP que contribuyen para el logro de objetivos de calidad. Este artefacto es desarrollado durante la fase de inicio y es actualizado en cada hito.

2.2.3 Establecer la integridad

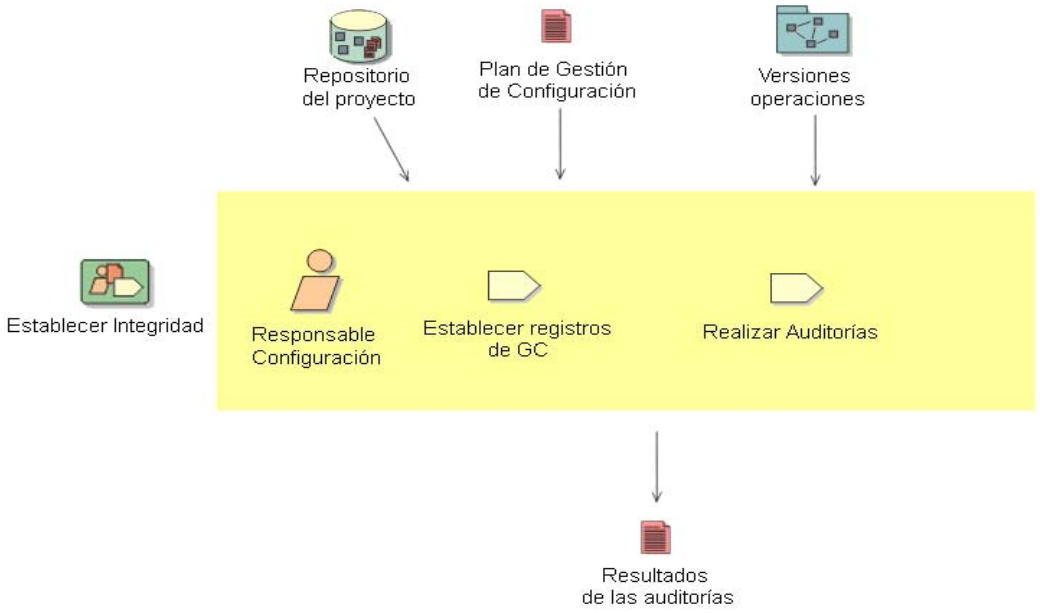


Fig. 2.4: FT Establecimiento de la integridad

2.2.3.1 Establecer registros de gestión de configuración

Esta actividad se ejecutará con el objetivo de establecer y mantener la integridad de las líneas bases, por lo que deben realizarse diferentes revisiones históricas a los elementos de configuración, hacer cambios de los registros, efectuar copias de las peticiones de cambio, llevar el estado de los elementos de configuración y conocer las diferencias que existen entre las líneas bases.

Para esto se deben registrar las acciones de gestión de configuración con suficiente detalle, para que el contenido y el estado de cada elemento de configuración sea conocido y las versiones anteriores puedan ser recuperadas. Se debe de asegurar que los stakeholders tengan acceso y conocimiento del estado de los elementos de configuración, por ejemplo: proporcionar permiso de acceso a los usuarios finales autorizados, hacer que las copias de las líneas bases estén disponibles a los usuarios finales autorizados.

Además se debe especificar siempre cual es la última versión de las líneas bases, asimismo identificar la versión de los elementos de configuración que constituyen una línea base en particular, se deben describir las diferencias entre dos líneas bases sucesivas y revisar el estado y el historial (cambios y otras acciones) de cada elemento de configuración si es necesario.

2.2.3.2 Realizar auditorías de la configuración

Una auditoría sirve como guía para asegurarnos que una colección de elementos de configuración que conforman una línea base, siguen un estándar especificado o un requerimiento.

Existen diferentes tipos de auditorías: auditorías funcionales, físicas y de gestión de configuración.

Las auditorías funcionales están dirigidas a verificar que las características funcionales probadas de un elemento de configuración han logrado los requisitos especificados en su documentación funcional de línea base y que su documentación de apoyo y operacional es satisfactoria y completa.

Las auditorías físicas de configuración están dirigidas a verificar que los elementos de configuración conforman la documentación técnica que los define.

Las auditorías de gestión de configuración están dirigidas a confirmar que los archivos de gestión de configuración y los elementos de configuración están completos y son consistentes y precisos.

Para mantener la integridad de las líneas bases se debe primeramente permitir el acceso de forma autorizada a la misma, además se debe tener bien claro si los archivos de gestión de configuración identifican correctamente a los elementos de configuración, también se debe confirmar la integridad y corrección de los elementos en el sistema de gestión de configuración y revisar la estructura y la integridad que estos tengan dentro del mismo sistema.

La integridad y la exactitud del contenido se basan en los requisitos indicados en el plan y la disposición de peticiones de cambio aprobadas.

Se debe confirmar además la conformidad con las normas aplicables de gestión de configuración y los procedimientos, y seguir los elementos de acción de la auditoría para el cierre.

2.2.3.3 Artefactos de entrada y salida

2.2.3.3.1 Repositorio del proyecto

El repositorio del proyecto almacena todas las versiones de los archivos del proyecto y los directorios. También almacena todos los datos derivados y metadatos asociados con los archivos y directorios.

El repositorio del proyecto almacena todos los ficheros y directorios que están bajo la herramienta de control de versiones, es un recurso global que necesita ser accedido por la mayoría de los integrantes del proyecto. En dependencia del tamaño del proyecto será el

tamaño del repositorio. Asimismo el número de archivos del repositorio dependerá del tamaño de la máquina donde se encuentre alojado este.

Este artefacto es desarrollado a comienzos del ciclo de vida del proyecto y mantenido hasta el fin del mismo.

2.2.3.3.2 Versiones operacionales

Una versión operacional es una parte del sistema que posee un subconjunto de las capacidades que tendrá el producto final. Posee uno o más elementos de implementación (a menudo ejecutables), cada uno construido de otros elementos usualmente por un proceso de recopilación y eslabonamiento de código fuente.

2.2.3.3.3 Plan de gestión de configuración de software.

Este artefacto está especificado en el flujo de trabajo anterior.

2.2.3.3.4 Resultado de las auditorías

Estos resultados van a identificar a las líneas bases, cualquier artefacto requerido faltante y los requerimientos incompletamente probados o ausentes.

El propósito de estas auditorías es reportar ya sea la realización del desarrollo del software como que los artefactos requeridos estén físicamente presentes. (RATIONAL 2003b)

2.3 Propuesta de capacitación al personal.

Después de analizar los resultados obtenidos en la encuesta anteriormente aplicada se llegó a la conclusión de que el personal de nuestra institución no se encuentra

correctamente capacitado, esta afirmación es debido a que un 78% del total de encuestados planteó que no existe adiestramiento alguno. Como inicialmente se explica la GCS es una disciplina de control dentro de los proyectos, sin embargo es una de las actividades claves para que una organización de desarrollo de software pueda alcanzar el Nivel de Madurez 2 establecido por el SEI, pero al mismo tiempo influye en la garantía de calidad de un producto de software.

Para dar solución al problema anterior se propone capacitar al personal tanto de los proyectos como los no vinculados aún, mediante un Curso de Introducción a la Gestión de Configuración de Software.

2.3.1 Duración:

La duración específica del curso, aún no está definida pero según los resultados que arrojó la encuesta aplicada se propone que se dedique por temas los siguientes valores, debido a que es en estos temas donde el personal que recibirá el curso tiene mayores dificultades.

El tema 1 se le debe dedicar un 15% del total de horas que se le asignen al curso y al tema 3 un 25% y debido a que la mayor importancia se centra en el tema 2 que se le debe dedicar el 60% de las mismas por ser en este tema donde precisamente se encuentra los epígrafes que formaron parte de la encuesta y donde radican las mayores dificultades.

2.3.2 Objetivo general:

Reflexionar acerca de los retos y prácticas alrededor de la calidad de procesos del software tomando como referencia los principios y procesos definidos en los modelos internacionales de calidad de software.

2.3.3 Objetivos Específicos:

- Describir el Proceso de Gestión de Configuración de Software.
- Describir los artefactos que se generan como resultado de dicho proceso.
- Identificar cada uno de los roles que intervienen en la Gestión de Configuración, y las responsabilidades que tienen asignadas.
- Mostrar algunas herramientas utilizadas para automatizar el proceso.

2.3.4 Objetivos Educativos:

- Crear una conciencia de trabajo en equipo.
- Mostrar la necesidad de realizar un proceso de desarrollo de software con calidad.
- Inculcar la ética y valores que debe tener el Ingeniero de Software durante su trabajo.

2.3.5 Metodología

El programa se desarrollará combinando distintas estrategias metodológicas como exposición de seminarios, desarrollo de talleres y presentación y discusión de casos de estudio.

2.3.6 Temas:

Tema 1: Calidad del Proceso Software

- El entorno de desarrollo de software
    - Enfoques de la Calidad de Software
    - El contexto de la organización de software

    - Estándares de Calidad de Software
- El proceso de desarrollo de software
    - El Proceso Software
    - Modelos de Ciclo de Vida del Software
    - Factores relacionados con la calidad del proceso software
    - Modelos de Mejora de procesos software
    - Mejores Prácticas
- La Calidad del proceso software
    - El concepto de calidad
    - Antecedentes de la Calidad de Software
    - La Calidad de Software
    - Calidad a nivel de la organización y a nivel de los proyectos software
    - Modelo de evaluación de la calidad del producto software
- Retos y oportunidades de la industria de software
    - Productividad y competitividad de la industria de software
    - La industria de software cubana: retos y oportunidades.

Tema 2: Gestión de Configuración.

- Introducción a la gestión de la configuración de software
    - Conceptos generales y fundamentos de la Gestión de la Configuración bajo la perspectiva de la Ingeniería de Software
    - Conceptos básicos de Gestión de la Configuración del Software
    - El proceso de Gestión de Configuración del Software

- Identificación de objetos o artefactos en la Configuración del Software
- Control de versiones
- Control de cambios
- Control de defectos
- Auditoría de la configuración
- Informes de estado
- Herramientas de gestión de configuración
- Práctica / Taller: Formular a partir de un diagnóstico empresarial, la problemática que gira alrededor de Gestión de la Configuración de la empresa evaluada

- Organización de la Configuración del software
  - Estándares de Configuración del Software
  - Guías Prácticas
  - Gestión de la Configuración del software según: CMMI, ISO9000, ISO/SPICE, IEEE
  - La Organización
  - El responsable de la gestión de configuración
  - El comité de control de cambios
  - El bibliotecario
  - Metodología para adoptar un Sistema de Gestión de Configuración
  - Práctica/Taller: Formular alternativas de solución de la problemática planteada en el diagnóstico realizado, seleccionando y aplicando alguna de

las alternativas y buenas prácticas propuestas. En lo posible, se busca usar alguna herramienta de las que se propongan

- Disciplinas y Áreas complementarias a la Gestión de la Configuración
    - Conceptos generales y fundamentos de las disciplinas complementarias
    - Gestión del Entorno
    - Gestión de Pruebas
    - Alternativas de Gestión propuestas (CMMI, ISO 9000, IEEE)
    - Práctica / Taller: complementos de la solución que viene evolucionando a partir de robustecerla con los nuevos elementos propuestos en este capítulo.

Tema 3: Modelos Internacionales de Calidad para Aseguramiento de la Calidad de Software

- Conceptos Básicos de CMMI
- Áreas de Proceso de Soporte
- Verificación
- Validación
- Administración de la Configuración
- Aseguramiento de la Calidad de Procesos y Productos
- Medición y Análisis
- Gestión de Proveedores
- Análisis de Decisiones y Resolución

2.3.7 Bibliografía para el curso:

"El Proceso Unificado de Desarrollo de Software". Iver Jacobson, Grady Booch y James Rumbwgh. Addison-Wesley. 2000.

Pressman, Roger, "Ingeniería de software, un enfoque práctico", 2001.

De Antonio, Angélica, *"La Gestión de la Configuración del Software"*.

"Ingeniería de Software: Un enfoque práctico". Roger Pressman. Mc Graw-Hill/interamericana de España. De España. 2002.

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

CMMI (2005). The Capability Maturity Model Integrated. USA, Carnegie Mellon Software Engineering Institute.

IEEE Guide to Software Configuration Management, American National Standards Institute, 1987, Std. 1042-1987

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

IEEE Standard for Software Quality Assurance Plans". Std. 730-1998, IEEE Computer Society, 1998.

ISO 10007 Quality management – Guidelines for configuration management, ISO, 1995

2.4 Conclusiones

Con la realización de este capítulo se le ha dado cumplimiento a dos de los objetivos específicos que se plantearon en un principio. Se realizó el diseño de la propuesta para aplicar en los proyectos de la universidad y como método de enseñanza a la hora de aplicar la Gestión de Configuración de Software se decidió elaborar un curso que sirviera como capacitación al personal de la universidad.

CAPÍTULO 3 EVALUACIÓN DE LA PROPUESTA

Para la validación y aceptación del proceso propuesto en el Capítulo 2, se utilizó como herramienta la selección de un grupo de expertos a los que se les preguntó su opinión sobre las cuestiones referidas al proceso en evaluación y el empleo de técnicas propuestas en el Método Delphi. Las estimaciones de los expertos se realizaron de forma anónima.

El método Delphi, cuyo nombre se inspira en el antiguo oráculo de Delphos, fue ideado originalmente a comienzos de los años 50 en el seno del Centro de Investigación estadounidense RAND Corporation por Olaf Helmer y Theodore J. Gordon, como un instrumento para realizar predicciones sobre un caso de catástrofe nuclear. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro.(MONOGRAFIAS 2005)

Es considerado uno de los métodos subjetivos de pronóstico más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de expertos en el tema tratado. Otros autores definen la técnica Delphi como un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo. El mismo permite rebasar el marco de las condicionantes actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva.

"... el Delphi es la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas".

De manera resumida los pasos que se llevaron a cabo para garantizar la calidad de los resultados fueron los siguientes:

CAPITULO III

3.1 Proceso de selección de los expertos.

Antes de comenzar con la selección se debe obtener la cantidad de expertos, para esto se consideró que todos los participantes tuvieran conocimiento en las áreas relacionadas a la solución del problema propuesto.

Estás áreas son:

- RUP (Rational Unified Process)
- PSP (Personal Software Process)
- Calidad y Gestión del Software
- Modelo de mejora de procesos CMMI
- Método de evaluación de procesos SCAMPI
- Normas y estándares internacionales de calidad

Esta etapa es muy importante en cuanto a que el término de "experto" es ambiguo. Con independencia de sus títulos, su función o su nivel jerárquico, el experto fue elegido por su capacidad de ofrecer tanto valoraciones como recomendaciones conclusivas de un problema en cuestión y poseer conocimientos sobre el tema consultado. Las opiniones de los expertos fueron recogidas por vía electrónica y de forma anónima y así se obtuvo la opinión real de cada experto y no la opinión más o menos falseada por un proceso de grupo (se trató de eliminar el efecto de los líderes). Este panel de expertos se conformó con un grupo de especialistas en los temas de Calidad y Gestión de Software de la UCI y de la empresa Softel adjunta a ella.

La confiabilidad de la valoración emitida por este grupo depende, en primer lugar, del número de expertos que lo integre, en segundo lugar, de la estructura del mismo por especialidades y por último, de las características de los propios expertos. Para la descripción de los expertos, desde el punto de vista de la valoración de la calidad de la solución del problema, se pueden mencionar las siguientes características esenciales:

competencia, creatividad, disposición a participar en la encuesta, conformismo, capacidad de análisis y de pensamiento, espíritu colectivista y autocrítico.

Para la selección de los expertos se consideraron las siguientes etapas:

3.1.1 Confección del listado de expertos

Para la confección del listado de expertos se realizó un análisis de la posibilidad real de participación de cada candidato debido a que todos son miembros de la universidad o profesores adjuntos de la CUJAE y el ICID, poseen más de un año de experiencia en el proceso productivo, asimismo vinculación activa a la docencia, la mayoría a la asignatura Ingeniería de Software, proyectos productivos y a la vez participación en eventos científicos y otras actividades.

Todos los posibles candidatos tenían noción general de todos los temas abordados en las áreas de conocimiento que abarca el proceso propuesto. En total se seleccionaron siete expertos debido a que todos poseen más de tres años de experiencia promedio en los temas de calidad. En el anexo 2 se muestra una breve caracterización de cada experto.

3.1.2 Aprobación del experto para su participación

Cuando estuvieron seleccionados los expertos que cumplían con los requisitos para la realización de la evaluación del proceso se procedió a invitar a cada uno de forma personal o por vía electrónica, para que participara en la valoración. Se le explicó el objetivo de la realización de la encuesta y en qué consiste el método. Con esto se pretende conseguir la obtención de previsiones fiables.

3.2 Elaboración y lanzamiento de los cuestionarios

Una vez que fue recibida la respuesta positiva sobre la participación de los expertos, se procedió a la elaboración del cuestionario el cual se preparó de manera que facilitara la respuesta por parte de los consultados. Se tuvieron en cuenta las probabilidades de aplicación de los elementos a los cuales el proceso intenta dar solución; además los elementos que existen dentro de la organización en la actualidad y que podrían imposibilitar la ejecución en toda su magnitud del proceso. La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario.

En principio, la encuesta fue de preguntas abiertas, a manera de enfoque investigativo del problema, y en los cuestionarios subsiguientes se concentraron las preguntas en la evaluación del proceso, cerrando el entorno de las respuestas a los puntos básicos del problema tratado.

La inclusión de preguntas que determinaran la capacidad de valoración del tema por parte del experto, constituyó un elemento importante para derivar posteriormente la encuesta hacia los expertos, cuya opinión pudo ser significativa. En todos los casos se requirió cumplir con un lapso determinado para dar las respuestas o hacer las preguntas.

Naturalmente el cuestionario fue acompañado por una nota de presentación que precisó las finalidades, así como las condiciones prácticas del desarrollo de la encuesta (plazo de respuesta, garantía de anonimato). Además, en cada cuestión, se le propuso al experto que evaluara su propio nivel de competencia. En el anexo 3 aparece el cuestionario final realizado a los expertos.

CAPITULO III

3.3 Desarrollo práctico y explotación de resultados

Una vez aplicadas las encuestas y su posterior análisis se llegó a la conclusión unánime de los expertos de la importancia que tiene la implementación de los procesos de gestión de configuración de software en el proceso productivo de la UCI evitando de esta manera que no se pierda ninguna información relevante de un proyecto durante el desarrollo del mismo.

La razón de la afirmación anterior se debe a que en nuestra universidad no existe un único tipo de proyecto, sino que hay una gran diversidad y la mayoría de ellos consta de gran tamaño, todo esto implica que se requiera de un mayor control en los mismos ya que podría fácilmente perderse todo el proyecto sino se tiene implementado correctamente un sistema de GCS que satisfaga esas necesidades además se ha demostrado que con el tiempo ciertas personas logran hacerse imprescindibles por ser los que más conocen del tema en cuestión y son los únicos capaces de controlar el proceso y de darse el caso de la ausencia de alguna de estas personas, puede generarse un caos dentro del proyecto, la integración de los componentes se haría muy ineficiente y se cometerían muchos errores asociados a no tener un control real del software en desarrollo lo que puede traer como consecuencia que se incumplan los plazos establecidos en el plan del proyecto.

La correcta implementación de la solución propuesta conlleva ineludiblemente al impulso en el desarrollo de los equipos de trabajo pues la aplicación de las actividades de la gestión de configuración del software comprende una mayor participación y responsabilidad para los desarrolladores ya que la calidad de los procesos es algo que interesa de algún modo a todos los que de una forma u otra participan en la producción del software. Al mismo tiempo, inhibe la mezcla o pérdidas de artefactos de cualquier tipo en el desarrollo de ramas o versiones. Otra de las atribuciones por las cuales conviene poner en práctica los procesos de GCS es que siempre podemos contar con una versión o instancia del producto estable para demostraciones, independientemente de la continuidad del trabajo y la modificación del código en los distintos componentes. Todo esto, por

supuesto, debe realizarse aplicando las normas y estándares específicos establecidos para los procesos de desarrollo.

En esta encuesta se indagó principalmente sobre la necesidad de la implementación pronta de un sistema de GCS y las razones según las cuales consideraban los expertos de la premura de esta situación, especialmente en nuestra Universidad además de las razones por las cuales esto debe hacerse, asimismo como las consecuencias de su omisión. Otra cuestión lo fueron las actividades fundamentales que se deben llevar a cabo dentro de un proceso de Gestión de Configuración de software, su importancia, así como una pequeña evaluación de las herramientas utilizadas para este proceso.

En las respuestas hubo discrepancias entre los expertos a la hora de definir cuáles eran las principales actividades a implementar en la gestión de configuración de software. Sin embargo se pudo apreciar que muchos coincidían con la propuesta que estaba siendo objeto de evaluación por parte de los mismos.

En la tabla 1 se muestran las actividades definidas por los encuestados. En ella se pueden apreciar las coincidencias que tuvieron los expertos en sus criterios, marcados con el número 1, en caso de que no fuera citada por este se llenaba la casilla correspondiente con el valor 0.

Actividades propuestas por los expertos	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7	Total
Identificar elementos de configuración	1	1	1	1	0	1	1	6
Establecer sistema de GC	1	1	1	1	1	1	1	7
Crear líneas bases	1	1	1	1	0	1	0	5
Control de cambios	1	1	1	1	1	1	1	7
Control de versiones	1	1	1	1	1	0	1	6
Realizar auditorías	1	0	0	1	1	0	1	4
Admón. de entregas (release management)	1	0	0	0	1	0	0	2

Tabla 1: Actividades propuestas por los expertos

En la Fig. 3.5 se muestra la totalización de las coincidencias que tuvieron los expertos en las actividades de gestión de configuración de software.

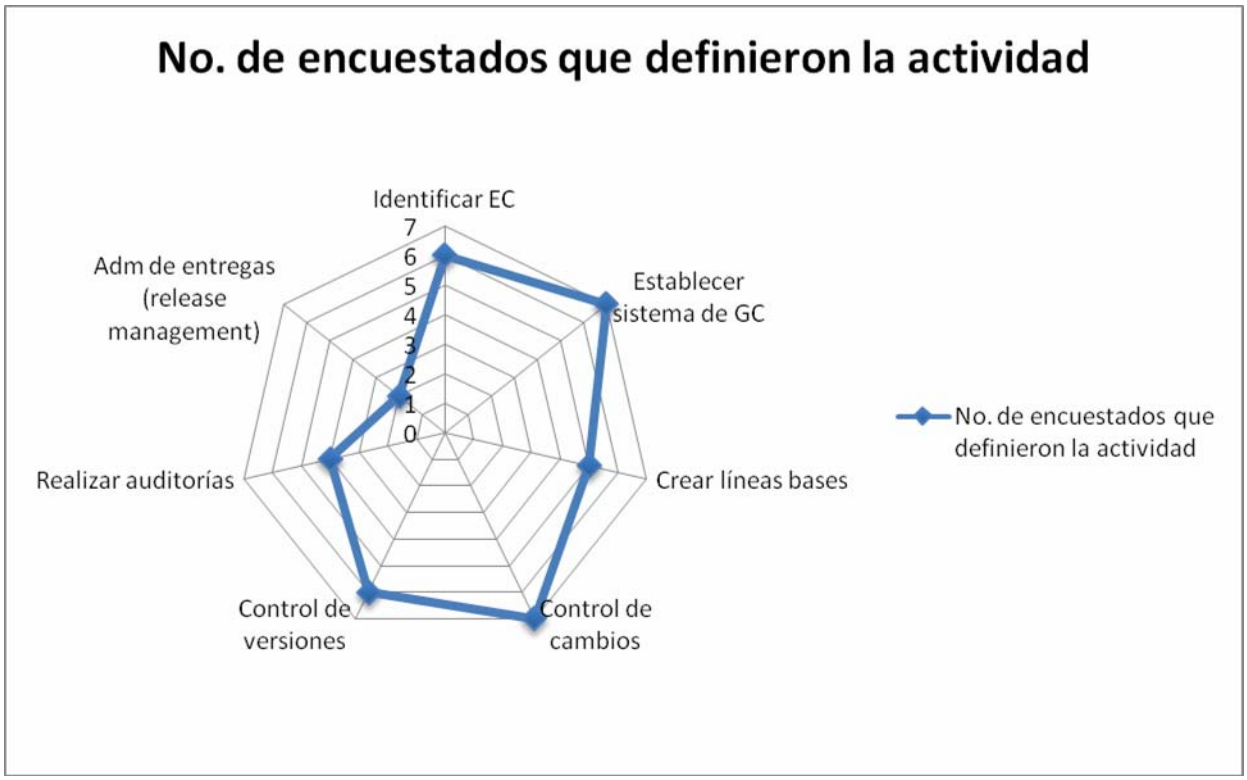


Fig. 3.5: No. de encuestados que definieron la actividad

En la tabla. 2 y la fig.3.6 que corresponde a la misma se muestra el nivel de importancia promedio dado por los expertos para cada una de las actividades.

Actividades propuestas por los expertos	**Nivel de importancia promedio**
Identificar elementos de configuración	7,86
Establecer sistema de GC	8,57
Crear líneas bases	7,57
Control de cambios	8,86
Control de versiones	8,86
Realizar auditorías	6,29
Admón. de entregas (release management)	5,57

Tabla 2: Nivel de importancia promedio

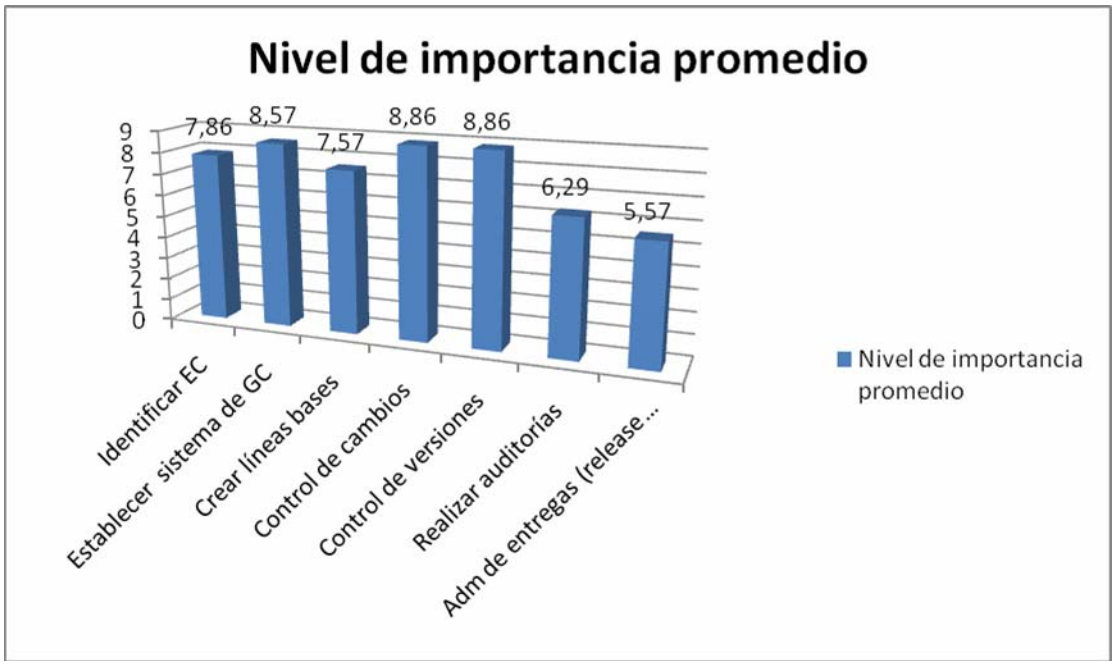


Fig.3.6: Gráfico del nivel de importancia promedio

Si observamos los valores del gráfico radial nos da una idea de la medida de importancia de cada una de las actividades evaluadas y si deben ser incluidas o no. Al realizar una comparación con el gráfico de nivel de importancia promedio que le conceden los expertos vemos como las actividades que mayor promedio tienen son en las cuales el mayor número de expertos coincidió e igualmente son las que están definidas ya sea tanto implícitamente o explícitamente en la propuesta que está siendo objeto de evaluación.

En sus comentarios los expertos identificaron como herramienta para llevar a cabo el control de versiones “Subversion” que es una herramienta probada a nivel mundial (coincidiendo con la herramienta propuesta) debido a que es un software libre, permite la integración con diferentes plataformas, es capaz de resolver conflictos y permite que varias personas puedan trabajar en el mismo proyecto.

En sus respuestas plantearon, además, la importancia de la utilización de estas herramientas debido a que posibilitan el control de los cambios y versiones de los elementos de configuración definidos en el proyecto, permite que el trabajo de los desarrollares se agilice y lo torna más seguro y eficiente.

Al responder a la pregunta que analizaba la efectividad de la aplicación del proceso de gestión de configuración de software que se propone, en los proyectos productivos de la UCI, todos los expertos estuvieron de acuerdo en que sería positiva, porque el proceso recoge todos los aspectos y actividades de la gestión de configuración, hace seguro el desarrollo de productos y versiones. Incorpora disciplina. Permite seguir los cambios realizados a cualquier componente a través del tiempo. Propone mejoras al proceso, de manera iterativa e incremental, así como representa una mejora en la comunicación entre los equipos, los desarrolladores y la organización.

El proceso, además comentaron, admite el uso de buenas prácticas de Ingeniería de Software y Administración de Proyectos a lo largo del proceso de desarrollo, por parte de todos los miembros del proyecto productivo. Esto último repercute directamente en el fortalecimiento del capital intelectual de la UCI y crea una retroalimentación necesaria y obligatoria que permite el desarrollo de una cultura de calidad.

Otro elemento que destacaron en sus comentarios a favor y en contra del proceso que estaba siendo evaluado por ellos fue la posibilidad de agilización del proceso en la práctica esto se debió a que muchas de las plantillas y documentos que tiene definida la Universidad se hacen un poco tediosas de llenar por su extensión y complejidad. Además generalizaron la opinión de que muchos de los líderes de proyecto aún no cuentan con una conciencia de calidad y tampoco con todo el conocimiento que lleva que se establezca esta disciplina de forma correcta asimismo sugieren que a la hora de realizar todo este proceso se necesita sistematicidad y no hacer las cosas por formulismo ni a posteriori sino que estas actividades se deben comenzar a realizar desde el principio del trabajo en el

proyecto y todos los involucrados deben mantener una conciencia de equipo y ayudarse unos a los otros.

CONCLUSIONES

Una vez concluida toda la investigación conjuntamente con la realización y evaluación de la propuesta para la aplicación y la enseñanza de la gestión de configuración de software en los proyectos productivos de la UCI se llegaron a las siguientes conclusiones:

- Todos los objetivos que inicialmente se trazaron fueron cumplidos cabalmente.
- Se logró caracterizar la gestión de configuración de software en la Universidad.
- Se realizó la propuesta para la aplicación de la gestión de configuración de software en la Universidad.
- Se obtuvo una evaluación positiva sobre aplicación de la misma.
- El curso que se propone de Introducción a la Gestión de Configuración para la capacitación del personal, cumple con los objetivos trazados en el Perfil Calidad del Software.
- La Gestión de Configuración de Software desempeña un papel importante en el proceso de desarrollo de software. Es una disciplina que está presente en normas y estándares utilizados a nivel mundial

RECOMENDACIONES

Para garantizar que el proceso propuesto se desempeñe correctamente se recomienda:

- Que se revisen las plantillas que tiene definida la Universidad para la elaboración de los planes u otros documentos con el fin de agilizar un poco más el proceso en su práctica.
- Que se continúe la búsqueda de un sistema de Gestión de Configuración que sea capaz de realizar todas las actividades propias del proceso.
- Que todo el personal involucrado con las actividades propias de la gestión esté capacitado para realizar estas tareas.
- Que el perfil docente- productivo se comience a aplicar cuanto antes.

BIBLIOGRAFIA CITADA

ANTONIO, A. D. LA GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE, 2001: 59.

APPLETON, B. and S. BERCZUK. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. 2002. 247 p. 0-000-000000-0

ASENSIO, R. M.-B.: *Conferencia sobre Herramientas CASE*. Departamento de Informatica, Murcia, 2007.

BERLACK, R. H. *Configuration Management.*, 1997. 89 p. 0-471-53049-2

BRADLEY, B. J. *Assurance of Software Quality*. CARNEGIE MELLON UNIVERSITY, S.-C.-.-. 1987.

CASALLAS, R. *Curso de Gestión de Configuración*, Universidad de Valencia, 2004.

CROSBY, P. B. *Quality is free: The art of Making Quality Certain*, 1979.

HUMPHREY, W. S. *Introduction to the Team Software Process (sm)*. ADDISON-WESLEY, 2000.

IBM. *IBM Rational ClearCase* 2007. [marzo, 2007]. Disponible en: http://www.ibm.com/software/awdtools/clearcase/

---. *IBM Rational ClearQuest*, 2007 b. [Disponible en: http://www.ibm.com/software/awdtools/clearquest

---. *Rational ClearCase Change Management Solution*, 2007 a. [marzo,2007]. Disponible en: http://www.ibm.com/software/awdtools/changemgmt

IEEE. *Guide to Software Configuration Management*, American National Standards Institute, 1987. Std. 1042-1987.

INFORMÁTICA, D. D. *"Censo del personal informático en Cuba".* SIME. , 1996.

ISO-10007 Quality Management - Guidelines For Configuration Management. ISO 10007,1995.

MARTÍNEZ, R. D. and E. G. SOCA. *ConfigCASE 3.0 HERRAMIENTA DE APOYO A LA GESTIÓN DE CONFIGURACIÓN. PROPUESTA ARQUITECTÓNICA.*, CUJAE, 2006. 135. p.

MONOGRAFIAS. 2005. [2007]. Disponible en: www.codesyntax.com/prospectiva/Metodo_delphi.pdf

MORENO, B. *Conferencia sobre el estado de la informática en Cuba*, Ministerio de la Informatica y las Comunicaciones. La Habana, 2002. p.

BIBLIOGRAFIA

MSDN. *Microsoft Visual SourceSafe*, Microsoft Corporation, 2007.

NAVARRO, A. . Conferencia sobre Gestion de Configuracion de Software, Valencia, España, 2004. 47 p.

PIATINI, M. G. C.-M., VILLALON; CERVERA, JOAQUÍN; FERNÁNDEZ, LUIS. *Análisis y Diseño detallado de aplicaciones informáticas de gestión.*, 1996. 179 p. 84-7897—233-1

PRESSMAN, R. S. *Ingeniería de Software, un enfoque práctico*. 3 ed. Mc GrawHill, 1997. p.

---. *Ingenieria de software: un enfoque práctico*. 5 ed., 2001. 601 p.

RATIONAL. *Rational Unified Process*. IBM, 2003a.

RATIONAL , S. C. RUP Help, 2003b.

RIVERA, J. G. *Curso de Calidad de Software*, Universidad de los Andes, 2003.

SEI. *Computer-Aided Software Engineering (CASE) Environments*, Carnegie Mellon Software Engineering Institute, 2007.

WIKIPEDIA. *Wikipedia, la Enciclopedia Libre*, 2007. [abril, 2007]. Disponible en: http://es.wikipedia.org,

BIBLIOGRAFIA CONSULTADA

[Appleton, 2000] Appleton Brad. SCM definitions, 2000, Disponible en: http://www.enteract.com/~bradapp/scm-defs.html, mayo 2007

[Babich 1986] Babich, W, Software Configuration Management, Adison – Wesley, 1986

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

[CSIS-Limerick,2005] Burkley Grace, Donohoe Colm, Spellman Geraldine *Software Configuration Management.* Disponible en: http://www.csis.ul.ie/Modules/CS5122/SCM.htm, mayo del 2007-06-15

[CMM 2003] CMM, The Capability Maturity Model. USA, Carnegie Mellon Software Engineering Institute, 1993

[CMMI 2005] CMMI (2005). The Capability Maturity Model Integrated. USA, Carnegie Mellon Software Engineering Institute.

[cmBook, 1998] Brown Norm; W. Evans Michael; Gathmann-Hobbs Jeannine. *Litle Book of Configuration Manager.* Computer & Concepts Associates, 1998.

[Database 2006] Database Journal, Disponible en : http://www.databasejournal.com/features/mssql/article.php/3087841, marzo del 2007

[Franco, 2004] Franco, José Ángel, ConfigCASE v2.0 Herramienta para la Gestión de Configuración de Software. Trabajo de Diploma, La Habana, Cuba, CUJAE, 2004.

[Gartner 2001] Gartner, Describing the Capability Maturity Model, Gartner, Inc. 2001

[Gervás, 2002], Gervás Pablo, Estándares y Gestión de Configuración, Capítulo 1, UCM, 2002, p4.

[Jacobson, 2000], I. El Proceso Unificado de Desarrollo de Software, Addison Wesley Longman Inc, 2000

[SPICE 2005] SIPICE, 2005. Disponible en: http://www.sqi.gu.edu.au/spice/, abril 2007

GLOSARIO DE TÉRMINOS

Auditoría: Sirve para complementar la revisión técnica formal al comprobar características que generalmente no tiene en cuenta la revisión.

CMMI: Capability Maturity Model Integration: Integración del Modelo de Madurez de las Capacidades.

EC: Elemento de configuración

Estándar: Es aquello que se considera modelo.

GCS: Gestión de Configuración de Software.

GS: Gestión de Software.

IDE: Entorno Integrado de Desarrollo. Ambiente que agrupa variadas funcionalidades para la construcción, mantenimiento, depuración de programas, ejemplos de IDE son el VisualStudio.Net y el Eclipse.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos

IS: Ingeniería de Software

ISO: Organización Internacional de Normalización.

OCI: Orden de cambio de Ingeniería.

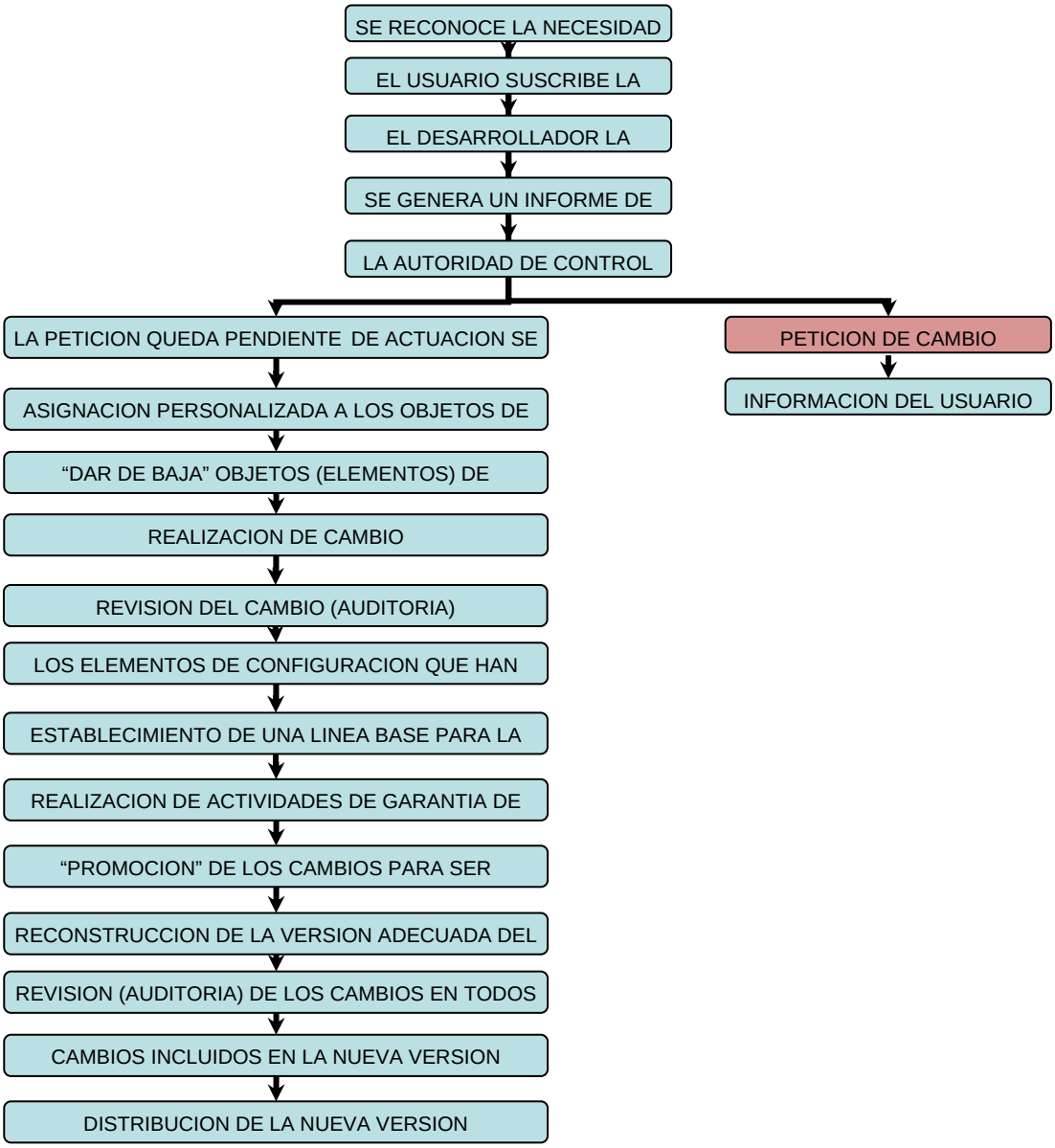
RUP: Rational Unified Process: Proceso Unificado de Desarrollo de Software de Rational.

SEI: Instituto de Ingeniería de Software

UML: Lenguaje Unificado para el Modelado. Empleado mayormente para la modelación visual de aplicaciones con enfoque orientado a objeto, pero con mecanismos de extensibilidad para emplearlo en otros contextos.

ANEXOS

ANEXO 1: PROCESO DE CONTROL DE CAMBIO



ANEXO 2: CARACTERIZACION DE LOS EXPERTOS

Experto 1

Máster en Ciencias Informáticas. Profesor de Ingeniería de Software de la CUJAE. Graduado hace 3 años. Posee publicaciones en informática y obtuvo mención en el pasado fórum nacional de ciencia y técnica.

Experto 2

Graduada de Lic. Matemática en 1972 en la Universidad de la Habana. Máster en Sistemas Digitales 1979. Investigador Auxiliar desde Mayo 1982. Profesor Titular Adjunto desde 1996. Trabaja desde 1973 en el Instituto Central de Investigación Digital (ICID). Desarrolló software base para las minicomputadoras cubanas, incluyendo compiladores. Amplia experiencia en el desarrollo de aplicaciones sobre MS-DOS, UNIX y Windows. Actualmente participa en el desarrollo de equipos médicos de alta tecnología. Trabaja sobre Linux. Tiene publicaciones nacionales e internacionales. Amplia participación en eventos científicos. Ha obtenidos variados premios y reconocimientos, entre ellos: Premio de plata en concurso Centro Regional Enseñanza Informática, España. Condición de Mención y de Destacado en Fórums Nacionales de Ciencia y Técnica. Premio Orlando Ramos por ponencia más técnica en evento TECBIOMED del ICID.

Experto 3

Graduado de Ing. Industrial en 1991 en la CUJAE. Máster en Informática Aplicada. Profesor adjunto desde 1995. Ha participado en eventos y fórum nacionales. Tiene certificado el curso de CMMI.

Experto 4

Graduado de Ing. Informática en el 2003. Profesor adjunto desde hace tres años. Tiene certificado el curso de CMMI. Ha participado en convenciones de informática. Es profesora de Ingeniería de Software de la UCI.

Experto 5

Graduado de Lic. Cibernética en el año 2001, lleva cinco años vinculada a la UCI, es profesora de Ingeniería de Software, tiene certificado el curso de CMMI. Ha participado en eventos nacionales, obtuvo mención en el pasado fórum de ciencia y técnica, tiene buena participación en los proyectos productivos.

Experto 6

Graduado de Ing. Informático. Lleva cuatro años vinculado a la UCI. Tiene avalado el curso de CMMI. Participo en diferentes eventos como fueron Evento Sepgla 2004, Informática 2007 Uciencia. Especialista en Calidad. Tiene experiencia en pruebas a diferentes software: Identidad, Digitalización, Multimedia, Intranet de PDVSA, entre otros.

Experto 7

Graduada de Ing. Informática, lleva tres años vinculada a la UCI, se desempeña como Jefa de Departamento de la Especialidad y asesora de calidad en su facultad. Profesora de Ingeniería de software y metodología de la Investigación. Participó en Uciencia obteniendo mención y en Informática 2007. Tiene certificado el curso de CMMI.

ANEXO 3: CUESTIONARIO PRESENTADO A LOS EXPERTOS

Usted ha sido seleccionado para evaluar el proceso de Gestión de Configuración de software en la UCI.

1) ¿Considera usted que es importante aplicar los procesos de Gestión de Configuración de Software en los proyectos de la UCI?

Si____ No___¿Porqué?_____

_____

2) ¿Cuáles considera usted que deban ser las actividades fundamentales que se deban llevar a cabo dentro de un proceso de Gestión de Configuración de software?

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____

3) Evalúe dichas actividades por el nivel de importancia que usted le concierne. (1-10)

Actividades	1	2	3	4	5	6	7	8	9	10
Nivel de importancia										

4) ¿Qué herramienta para llevar a cabo la GCS usted propondría? ¿Por qué?

_____

_____

_____

5) ¿Considera que la utilización de estas herramientas son importantes? ¿Por qué?

_____

_____

_____

6) ¿Qué considera usted que pasaría si no se aplicaran correctamente los procesos de GCS en los proyectos?

_____

_____

_____

7) ¿Cree usted que el proceso evaluado pueda tener una correcta aplicación en la UCI y cumpla con las actividades básicas de la GCS?

Si____ No___ Porqué?_____

_____

_____

8) Realice un comentario sobre el proceso que está siendo evaluado por usted (que le ve a favor y en contra)

_____

_____

_____

Le garantizamos total discreción y confidencialidad con todo lo que se escriba aquí al respecto.

ANEXO 4: CARACTERIZACIÓN DEL PROCESO DE GCS EN LA UCI