

**Universidad de las Ciencias Informáticas**

**Facultad 3**



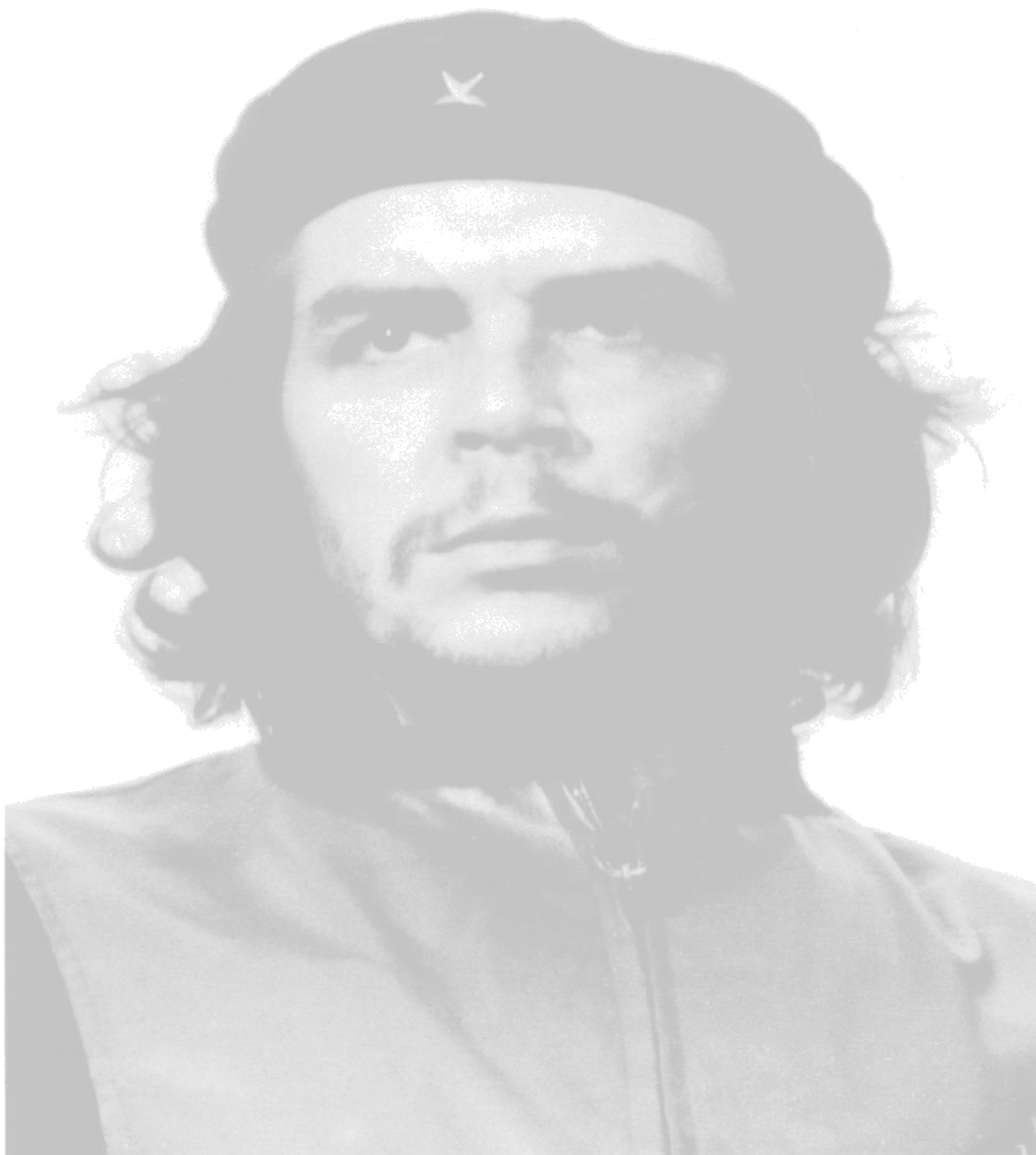
**Título: Propuesta de nuevos artefactos para el expediente de proyecto de la UCI en el área Gestión de la Configuración.**

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

**Autor:** Carlos Yanko Pozo Carralero.

**Tutor:** Ing. Susana Gonzalez Espinosa.

Junio 2009



*“Seamos Realistas y hagamos lo imposible”*

*Ernesto Guevara*

## DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y le otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Carlos Yanko Pozo Carralero

Ing. Susana Gonzalez Espinosa

---

Firma del Autor

---

Firma del Tutor

**DATOS DE CONTACTO**

Ing. Susana Gonzalez Espinosa.

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Categoría docente: Adiestrado.

[sespinosa@uci.cu](mailto:sespinosa@uci.cu)

**Agradezco de todo corazón:**

- ❖ A mi madre todo mi agradecimiento, por su constante dedicación hacia mi persona, por su amor, por su apoyo en los momentos difíciles, por su ejemplo.
- ❖ A mis hermanos Héctor, Daymarelis y Yilemnis por su incondicional apoyo y amor hacia mi, en parte les debo todo esto que he logrado, sin ellos me hubiese sido casi imposible.
- ❖ A mi padre que aunque no está conmigo desde los 4 años se que hubiese estado muy orgulloso de mi graduación.
- ❖ A mis abuelos Paula, Oscar y Carlos que aunque ya no están conmigo siempre los recuerdo.
- ❖ A mi tutora Susana por su dedicación y comprensión, por su ayuda y por su paciencia, por ser la mejor tutora del mundo.
- ❖ A mi ex padrastro Reymundo por ser el padre que no tuve, por su ejemplo.
- ❖ A mi padrastro Ramoncito por aceptarme como un hijo más suyo.
- ❖ A Yadira, Yurismara, Lilian y Dailiana por ser las mejores amigas que he tenido en toda mi vida y estar ahí cuando las he necesitado.
- ❖ A todos mis compañeros de la universidad que hemos compartido los buenos y malos momentos.

- ❖ A mi maestro de la primaria Alcibiades que siempre confió en mí como estudiante y persona.

**Dedico la presente tesis:**

- ✓ A mi madre que es lo más grande que tengo en este mundo y quien es la dueña absoluta de todos mis logros y de mi vida.
- ✓ A mi padre que ya no está conmigo y que aunque sé que le hubiese gustado que hubiese estudiado música, igualmente hubiese estado orgulloso de mí si me pudiese ver en estos momentos.
- ✓ A mis hermanos que son parte inseparable de mi vida.
- ✓ A Fidel y la Revolución por darme la oportunidad de estudiar en esta universidad.
- ✓ A mis amigas inseparables Yadira, Yurismara, Lilian y Dailiana.

## **Resumen**

El presente trabajo se realizó con el propósito de confeccionar un conjunto de plantillas para el Expediente de Proyecto de la Universidad de las Ciencias Informáticas en el área Gestión de la Configuración que contribuyan a una documentación eficiente de la misma en los proyectos productivos.

Este trabajo se centra en definir los indicadores que deben contener el grupo de plantillas que se elaboraron con el objetivo de llevar un buen control de la documentación en el área de Gestión de Configuración del Expediente de Proyecto de la Universidad de las Ciencias Informáticas, apoyado en las actividades que definen los modelos, normas y estándares para la realización de la Gestión de la Configuración, se tuvo en cuenta para realizar la selección de los principales aspectos que deben incluirse en las plantillas, cada una de las actividades de la Gestión de la Configuración y la compatibilidad que deben tener las mismas con respecto a los artefactos ya existentes en el expediente de proyecto vigente.

Las plantillas elaboradas fueron sometidas al criterio de un grupo de especialistas con conocimientos en el tema y que han estado relacionados por su rol o por la actividad que desempeñan en la universidad con el tema, algunos relacionados con el rol Gestor de Configuración en proyectos productivos de la universidad. También se realizó un análisis de los resultados obtenidos de dichos criterios con el propósito de comprobar si la investigación cumplía con los resultados esperados.

## **Palabras claves**

Gestión de Configuración de Software.



**Índice**

AGRADECIMIENTOS .....	IV
DEDICATORIA .....	IVII
RESUMEN.....	IV
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1    Introducción.....	4
1.2    Situación de la industria del software.....	4
1.3    Proceso de desarrollo de software.....	5
1.4    Gestión de Configuración. Definiciones.....	6
1.4.1    Apreciaciones sobre la Gestión de Configuración de Software (GCS).....	7
1.4.2    Modelos, normas y estándares de calidad referidos a la GCS.....	8
1.4.2.1.    ISO (Organización Internacional de Normalización).....	8
1.4.2.2.    IEEE (Instituto de Ingenieros Eléctricos y Electrónicos).....	9
1.4.2.3.    CMMI (Modelo de Capacidad y Madurez Integrado).....	9
1.4.3    La GC en la metodología Rational Unified Process (RUP).....	11
1.4.4    Actividades definidas por normas, modelos y estándares para la GC.....	13
1.5    Expediente de Proyecto (EP).....	14
1.5.1    Expediente de proyecto definido por calidad UCI.....	15
1.5.2    Requerimientos de la documentación asociada a los proyectos de la UCI.....	16
1.5.3    Plantillas definidas para el expediente de proyecto.....	18
1.5.4    Gestión de Configuración en el expediente de proyecto de la UCI.....	19
1.6    Conclusiones.....	20
CAPÍTULO 2: PROPUESTA SOLUCIÓN.....	22
2.1.    Introducción.....	22
2.2.    Actividades de la Gestión de la Configuración.....	23
2.2.1.    Identificación de la configuración.....	23
2.2.1.1.    Línea Base.....	24
2.2.1.2.    Elementos de Configuración de Software.....	24
2.2.1.3.    Bibliotecas de software.....	26
2.2.2.    Control de Cambios.....	27
2.2.3.    Auditoría de la Configuración.....	27
2.2.4.    Informes de Estado.....	29
2.3.3.1    Registros.....	30
2.3.3.2    Informes.....	36
2.3.    Estructura de las diferentes plantillas de la GC.....	38
2.3.1.    Plantilla Listado de Elementos de Configuración.....	38
2.3.2.    Plantilla Informe de Estado de la Configuración.....	38
2.3.2.1.    Informe de estado de los cambios.....	40
2.3.2.2.    Inventario de bibliotecas, líneas base y releases.....	41
2.3.2.3.    Informe de incidencias.....	42
2.3.2.4.    Informe de modificaciones.....	42
2.3.2.5.    Informe de diferencias de versiones.....	42
2.3.2.6.    Informe de deficiencias o discrepancias.....	43
2.3.3.    Plantilla Auditoría de la Configuración.....	44
2.4.    Aspectos generales que se incluyen en las plantillas.....	45
2.5.    Conclusiones.....	45

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA.....	47
3.1. Introducción.....	47
3.2. Selección de los especialistas. ....	47
3.2.1. Cantidad de Especialistas.....	48
3.2.2. Listado de especialistas.....	48
3.2.3. Participación de los especialistas. ....	49
3.3. Elaboración de la encuesta.....	49
3.4. Resultados de la evaluación .....	50
3.5. Conclusiones.....	52
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES .....	55
BIBLIOGRAFÍA.....	56
ANEXOS.....	58

## Introducción

El siglo XXI se enfrenta a la creciente implantación de la sociedad del conocimiento. La era del conocimiento en que vivimos no solo está cambiando la sociedad en sí misma, sino que los nuevos modelos de negocios requieren la reformación de nuevos conceptos. La ingeniería de software no es una excepción y por ello requiere no solo una actualización de conceptos, sino también una comprensión y una formulación del nuevo conocimiento existente en torno a las nuevas innovaciones y teorías de dicha disciplina (Pressman, 1998).

Actualmente el proceso de transformación y perfeccionamiento de las tecnologías de la información y la comunicación (TIC) ha traído consigo que se incremente cada día, la lista de países productores de software, donde la calidad de estos es imprescindible para la competencia y la inserción en el mercado. En este aspecto nuestro país está dando un paso de avance, existen empresas cubanas dedicadas a la producción y comercialización de software como son Softel, Desoft y Segurmática. Con el mismo objetivo se ha definido como estrategia, la creación de la UCI<sup>1</sup> como centro de investigación y desarrollo de software. En la UCI se desarrollan diversos proyectos de software donde se hace necesario aplicar apropiadamente metodologías, procedimientos y estándares que permitan desarrollar un software con calidad.

Para ello la UCI creó la Dirección de Calidad de Software que ha realizado diversas funciones entre las que se encuentran la propuesta de diferentes políticas, estrategias y herramientas con el objetivo de que el proceso de calidad en los proyectos productivos sea efectivo. A pesar de los esfuerzos realizados por la Dirección de Calidad de Software, todavía deben seguir trabajando para lograr certificarse en el nivel 2 de CMMI<sup>2</sup> y de esta forma mejorar los procesos que se llevan a cabo en la producción de software y esto a su vez le abrirá puertas en el mercado del software.

En la UCI, la Dirección de Calidad de Software definió que no era posible organizar la documentación de los proyectos sin definir un esquema genérico, que norme como usar y evaluar la documentación según el tipo de proyecto. Para lograr esta organización en los proyectos de desarrollo de software en la UCI se decidió definir una estructura para el

---

<sup>1</sup> Universidad de las Ciencias Informáticas

<sup>2</sup> Capability Maturity Model Integration

expediente de proyecto y un grupo de plantillas, esta estructura seleccionada debía tener una cierta compatibilidad con el modelo CMMI, pues es el modelo seleccionado para mejorar los procesos de desarrollo de software en la UCI. El expediente se ha ido construyendo basándose en las actividades que se deben realizar en las áreas de procesos definidas por CMMI.

La UCI está trabajando para certificarse en el nivel 2 de CMMI, para lo cual debe cubrir casi todas las áreas de procesos de las 4 categorías de este nivel. Actualmente en el expediente de proyecto faltan por cubrir actividades el área de proceso Gestión de la Configuración, lo que trae como consecuencia que no se documente correctamente la información relacionada con la Gestión de la Configuración del Software, por lo que se impide la estandarización y retroalimentación tan importante y necesarias planteadas en los objetivos de la creación del expediente de proyecto en la UCI.

### **Problema científico**

La información existente actualmente en el expediente de proyecto en el área Gestión de la Configuración es insuficiente para un control eficiente de la misma.

### **Objeto de estudio**

Gestión de la Calidad del Software.

### **Objetivo general**

Identificar los indicadores que debe tener el expediente de proyecto en el área Gestión de la Configuración para garantizar un control eficiente de la misma.

### **Campo de acción**

Gestión de la Configuración del Software.

### **Hipótesis**

Si se identifican los indicadores que debe tener el expediente de proyecto en el área Gestión de la Configuración de la UCI entonces se garantizará estandarizar y llevar un control de la documentación relacionada con esta área en los proyectos productivos.

## **Tareas de la investigación**

Para darle cumplimiento al objetivo general se plantean las siguientes tareas de la investigación:

1. Elaborar el marco teórico de la Gestión de Configuración y del expediente de proyecto de la UCI.
2. Caracterizar y evaluar las actividades que se llevan a cabo en el área Gestión de Configuración de Software, adaptando modelos, normas y estándares estudiados al expediente definido en la UCI.
3. Identificar los indicadores que debe tener el expediente de proyecto en el área Gestión de la Configuración, para garantizar un control eficiente de la misma.
4. Elaborar las plantillas que se integrarán al expediente de proyecto de la UCI.
5. Validar los indicadores que componen cada una de las plantillas elaboradas.

El trabajo consta de los siguientes capítulos:

El capítulo 1 se centra en el fundamento teórico; donde se abordan aspectos importantes sobre Gestión de la Configuración del Software y el Expediente de Proyecto de la UCI, se define el concepto de Gestión de Configuración de Software así como las actividades fundamentales que se deben desempeñar en la misma. Se realiza un estudio de las actividades que incluyen modelos, estándares y normas de calidad con el fin de comprobar la necesidad de la Gestión de la Configuración en el desarrollo de software. Se tratan además, de forma concisa, las principales características del actual expediente de proyecto de la UCI.

El capítulo 2 se centra en describir las actividades que se realizan dentro de la Gestión de Configuración efectuándose una selección de los aspectos más importantes de estas actividades y a partir de los mismos, se propone la solución para la problemática planteada.

En el capítulo 3 se validan los indicadores de las plantillas elaboradas para implantar en el expediente de proyecto de la UCI.

## Capítulo 1: Fundamentación Teórica.

### 1.1 Introducción

En el presente capítulo se describe de forma general el estado actual de la Gestión de Configuración de Software (GCS) de aquí en lo adelante, así como el proceso de construcción del actual expediente de proyecto en la UCI. Este capítulo es la base teórica para comprender el trabajo que se desarrolla y en él se tratan temáticas como:

- ✓ Conceptos sobre GCS.
- ✓ Modelos, normas y estándares de calidad referidos a la GCS.
- ✓ Expediente de proyecto de la UCI.

### 1.2 Situación de la industria del software.

La tendencia actual en la industria del software lleva a la construcción de sistemas cada vez más grandes y complejos. Por tal motivo, la presencia de un proceso bien definido y bien gestionado es la diferencia fundamental entre proyectos altamente productivos y otros que fracasan. En el mundo se han seguido dos tendencias fundamentales para lograr una buena calidad en el software, una es la seguida por las oficinas internacionales de estandarización como ISO <sup>3</sup>y la IEEE <sup>4</sup>y la otra la referida principalmente al desarrollo de software como lo son CMMI y SPICE<sup>5</sup>. Estas tendencias cada una por su parte, han arribado a conclusiones sobre qué es lo que realmente se necesita hacer para construir un software eficientemente y ambas en su conjunto han llegado a consensos sobre qué procesos son los que se necesitan para desarrollar un software con calidad. Resulta importante decir, que obtener éxito en la industria del software no es camino fácil pues en la actualidad existe una gran competencia en el mercado mundial del software. Existen actualmente muchas deficiencias e insatisfacciones por parte de los clientes en este aspecto. En varias ocasiones el software no es entregado en el tiempo establecido, o no satisface completamente las exigencias demandadas por los usuarios. Al profundizar en las raíces o causas de tales insatisfacciones surgen de manera reiterada la aplicación inadecuada de las disciplinas de Ingeniería de Software, la no utilización de los roles y procesos apropiados para el desarrollo de las

---

<sup>3</sup> International Organization for Standardization

<sup>4</sup> Institute of Electrical and Electronics Engineers

<sup>5</sup> Software Process Improvement and Capability Determination

tareas de la empresa de software y la no utilización de modelos de calidad en ellas (Estrada, 2003).

Dentro de estas disciplinas de Ingeniería se encuentra la GCS como uno de los procesos claves en el desarrollo de un producto informático (Navarro, 2006) y que actualmente su incorrecta aplicación constituye una de las principales causas de fracaso en grandes y millonarios proyectos en el mercado mundial.

### **1.3 Proceso de desarrollo de software.**

Se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad". (JACOBSON 1998).

"Es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". (JACOBSON 1998).

El éxito de un proyecto y por consiguiente de manera sostenida el de una empresa de software depende de la correcta ejecución de cuatro tipos de funciones según (Antonio, 2001):

- ✓ La Gestión del Proyecto.
- ✓ El Desarrollo Técnico.
- ✓ El Sistema de Calidad, que incluye las actividades de Validación.
- ✓ El Sistema de Gestión de Configuración (GC), de aquí en lo adelante.

A partir del proceso de ingeniería que se realiza en un software durante todo su ciclo de desarrollo, se genera información de diferente naturaleza y que se pueden clasificar según (Pressman, 2005) en:

- ✓ Programas de computadora (tanto en forma de código fuente como ejecutable).
- ✓ Documentos que describen los programas de computadora. (tanto técnico como de usuario).
- ✓ Datos (contenidos en el programa o externos a él).

Es decir, los elementos que conforman toda la información que se genera dentro del proceso de ingeniería de software se denomina colectivamente configuración del software.

### **1.4 Gestión de Configuración. Definiciones.**

Según (Rancán, 2003) la GCS es la rama de la Ingeniería de Software encargada de controlar la evolución de todos los elementos de configuración existentes en el desarrollo, involucrando un conjunto de técnicas para gestionar con eficiencia las modificaciones que se realicen sobre estos elementos a lo largo de su ciclo de vida.

En el diccionario de términos IEEE esta disciplina es conceptualizada, “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 producto, almacenando el estado de los elementos de configuración y de las solicitudes de cambio, así como verificando la completitud con respecto a los requerimientos especificados (Proceso de Control de Cambios Guiado por la Arquitectura del Software)”, fue definida en los años 70 e implantada en los años 90.

Una definición en términos generales sobre GCS es la que nos brinda (Antonio 2001): “... disciplina, cuya misión es controlar la evolución de un sistema software”

Aunque parezca algo simple la definición anterior, la misma resulta muy abarcadora, lo cuál se verá más adelante al analizar cada uno de los elementos que componen la GCS.

Para Babich (Babich, 1986): “El arte de coordinar el desarrollo de software para minimizar la confusión se denomina GC. La GC es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.”

En (Rational, 2003) se dice que la GC “describe la estructura del producto e identifica los elementos que lo constituyen y que son tratados como entidades que pueden ser puestas bajo control de versiones en el proceso de GC. La GC tiene que ver con la definición de la configuración, así como la construcción, el etiquetado y recolección de versiones de los artefactos”.

(Brown 1998) El propósito fundamental de la GC es establecer y mantener la integridad y el control en los productos software a lo largo del ciclo de vida del proyecto.



Para algunos autores la GC es la que administra y controla el contenido, el cambio o el estado de la información compartida en un proyecto.

De todas las definiciones abordadas se ha considerado la más completa para la presente investigación y a su vez más concisa y abarcadora la de Ivar Jacobson, Martín Griss y Patrick Jonson en el libro “Software Reuse: Architecture, Process, and Organizations for Business Success” en la que se expresa que (Jacobson, 2000):

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

### **1.4.1 Apreciaciones sobre la Gestión de Configuración de Software.**

A continuación se brindan una serie de valoraciones de disímiles autores que permiten apreciar la importancia que le concede cada uno de ellos a la GC dentro del proceso de desarrollo de software.

Watts Humphrey (Humphrey, 1989), en el libro, Administrando el Proceso de Software plantea que en el desarrollo de software, los problemas más frustrantes, son causa de un pobre proceso de GC. Y resultan frustrantes porque toman tiempo para solucionarlos, ocurren generalmente en momentos críticos y su ocurrencia es totalmente innecesaria.

Pressman le atribuye en (Pressman, 1997) a la GC el término en inglés “umbrella activity”, al revisar el uso del adjetivo “umbrella” se encontró que es empleado para actividades abarcadoras, globales, incluyentes, rectoras.

No debe dejarse de mencionar la Ley Fundamental de la Gestión de Configuración [cmBook, 1998], donde se define el papel de la GC de Software en el proceso de desarrollo.

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

Para concluir podemos hacer referencia en (Rational, 2004), a una frase escrita por Tom Milligan asesor de IBM Rational Software, donde plantea que: “La Gestión de Configuración de Software es el héroe no nombrado en el proceso de desarrollo de software”.

### **1.4.2 Modelos, normas y estándares de calidad referidos a la GCS.**

El proceso del ciclo de vida del software elegido para un proyecto de software en particular y las herramientas elegidas para la implementación de dicho software, afectan el diseño e implementación de los procesos de la GCS. Las “mejores prácticas”, como se reflejan en los estándares de la ingeniería del software publicados por varias organizaciones de estándares, se pueden usar como consejo para el diseño e implementación de un proceso de la GCS en las organizaciones productoras de software.

Para la Gestión de Calidad de Software a nivel mundial se han seguido principalmente dos tendencias: la primera a seguir son las reglas implantadas por las oficinas internacionales de estandarización para los productos y servicios a través de las normas ISO y la IEEE, y la segunda a seguir, las creadas específicamente para el mundo del software como CMMI y SPICE.

#### **1.4.2.1. ISO (Organización Internacional de Normalización).**

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios como de aplicaciones (ISO, 2005).

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

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

- ✓ Sistema de calidad.
- ✓ Especificación de los requisitos del comprador.
- ✓ Planificación del desarrollo.
- ✓ Planificación de la calidad.
- ✓ Pruebas y validaciones.

- ✓ Gestión de Configuración de Software.
- ✓ Control de documentos.
- ✓ Mediciones.

El hecho de que una norma de calidad tan reconocida a nivel mundial incluya dentro de sus requerimientos, el de efectuar las actividades de GCS, es totalmente consistente con la importancia que le otorgan los diversos autores analizados anteriormente.

#### **1.4.2.2. IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)**

Al hablar de los estándares para la Calidad de Software es prácticamente imposible dejar de mencionar el estándar IEEE 730-1998 para el Plan de Aseguramiento de la Calidad de Software (IEEE, 1998). Este aborda aspectos como la administración, la documentación, el control de código, etc. Dentro de la documentación le brinda gran importancia al Plan para la Gestión de Configuración de Software reflejado en el estándar IEEE 828-1998 (IEEE, 1998).

Dicho plan trata todo lo referente a la asignación de las responsabilidades, la identificación de las actividades que se realizarán durante todo el proceso, la identificación de la configuración, el reconocimiento de los elementos de configuración, el control de la configuración, el acceso a las bibliotecas, la aprobación o desaprobación de un cambio y la implementación del cambio de ser aprobado (todas estas incluidas dentro de las tareas de la GC).

#### **1.4.2.3. CMMI (Modelo de Capacidad y Madurez Integrado).**

Al inicio de la década de los 90, surgieron una serie de iniciativas para aplicar las mejores prácticas en el ámbito del desarrollo del software que se plasmaron en la realización y publicación de una serie de modelos y normas para el sector.

Uno de estos modelos fue CMM desarrollado por el SEI <sup>6</sup>en el año 1991.

CMM para software o SW-CMM<sup>7</sup> no es más que un grupo de lineamientos para el desarrollo de software (Gartner, 2001). Una descripción de los escenarios por los cuales las organizaciones de

---

<sup>6</sup> Software Engineering Institute

<sup>7</sup> Software Capability Maturity Model

software evolucionan, cómo ellos definen, implementan, miden, controlan y desarrollan sus procesos de software (CMM, 1993).

Después del lanzamiento del modelo SW-CMM se desarrollaron otros modelos para otras disciplinas. Pronto las empresas se encontrarían que resultaba muy difícil integrar todos estos modelos, solucionar las lagunas detectadas, resolver las inconsistencias y aclarar las diferentes terminologías (Ema, 2002).

Para dar solución a estos problemas surge en diciembre del 2001 CMMI (CMMI, 2005), el cual es un enfoque para la mejora de procesos que brinda a las organizaciones los elementos básicos sobre procesos efectivos. Puede ser empleado para guiar la mejora de procesos a lo largo de un proyecto, división u organización. CMMI ayuda a integrar funciones organizacionales tradicionalmente separadas, otorga metas y prioridades en la mejora de procesos, brinda además una guía para los procesos específicos de calidad y un punto de referencia para la valoración de los procesos actuales (CMMI, 2005). Cuenta al igual que CMM con cinco niveles, aunque algunos cambian de nombre (Ema, 2002).

Cuando se habla de la representación escalonada que es la recomendada para alcanzar una certificación de CMMI, para poder escalar a un nivel superior es necesario haber cumplido todos los aspectos que plantea el nivel actual, por lo que es posible incluso, cumplir con aspectos del nivel 2 y del nivel 3 simultáneamente y, aún así, mantenerse en un nivel 1.

Para obtener Nivel 2 de CMMI (Gestionado) es necesario realizar las siguientes tareas (Ema, 2002):

- ✓ Gestión de los requisitos del producto y del proyecto.
- ✓ Planificación de los proyectos.
- ✓ Seguimiento y control de los proyectos.
- ✓ Gestión de acuerdos con los proveedores de productos y servicios.
- ✓ Selección y supervisión de los proveedores.
- ✓ Medición y análisis.
- ✓ Aseguramiento de la calidad del producto y del proceso.

- ✓ Gestión de Configuración de Software.

Vuelve a ponerse de manifiesto como tiene un lugar básico la GCS, pues de los cinco niveles que plantea el modelo, desde el número 2 hay que garantizar las actividades de GC para poder considerar que se cumple con la especificación del mismo.

Hasta este momento puede arribarse a la siguiente conclusión parcial. Aunque aún no han sido expuestas todas las actividades concretas que establece la GC, a partir del análisis bibliográfico efectuado, una situación queda claramente establecida, se valora que la misma posee una elevada importancia dentro del proceso de desarrollo de software por autores de gran prestigio, y consecuentemente es considerada en los principales modelos, normas y estándares de calidad de software a nivel mundial.

### 1.4.3 La GC en la metodología Rational Unified Process (RUP).

RUP es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Los autores de RUP destacan que el proceso de software propuesto tiene tres características fundamentales: está dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental. RUP ha agrupado las actividades en grupos lógicos definiendo 9 flujos de trabajo y en su ciclo de vida cuenta con cuatro fases: inicio, elaboración, construcción y transición, que guían cada uno de los flujos, como muestra la figura 1:

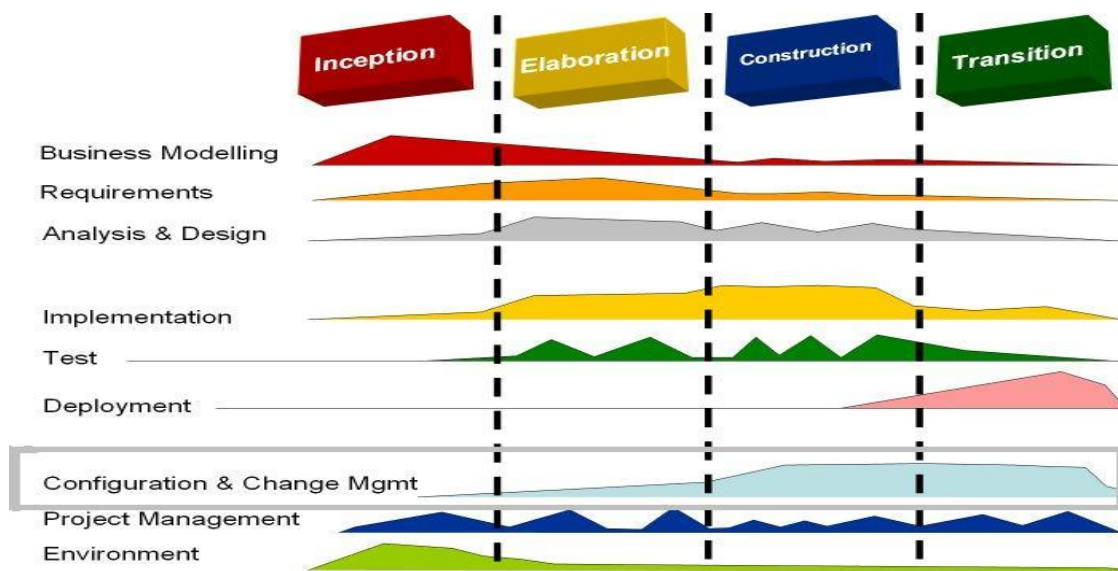


Figura 1: Rational Unified Process (RUP)

Entre los 9 flujos de trabajo se encuentran:

1. Modelado del negocio.
2. Requerimientos.
3. Análisis y Diseño.
4. Implementación.
5. Prueba.
6. Despliegue.
7. Configuración y gestión de cambios.
8. Gestión de proyecto.
9. Ambiente.

El concepto de hito en RUP es un concepto que provee de elementos tangibles (artefactos), que permiten decidir si el trabajo realizado hasta el momento es el adecuado o no, y si se han superado las necesidades de cada fase definida en la metodología.

El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no sólo debido a acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso de desarrollo, especialmente importantes por su posible impacto son los cambios en los requisitos. Por otra parte, otro gran desafío que debe abordarse es la construcción de software con la participación de múltiples desarrolladores, posiblemente distribuidos geográficamente, trabajando a la vez en un release, y quizás en distintas plataformas. La ausencia de disciplina rápidamente conduciría al caos. La configuración y gestión de cambios es la disciplina de RUP encargada de este aspecto.

La configuración y gestión de cambios como se puede apreciar en la Figura 1, se mantiene durante casi todo el proceso de RUP, desde el inicio del desarrollo del software hasta su final, teniendo su mayor peso a partir de la fase de construcción y hasta la fase de transición, lo que corrobora la importancia esencial de esta disciplina en el desarrollo de software.

La gestión del cambio, es una de las actividades básicas dentro de la GC y en la disciplina RUP se ocupa de tres áreas específicas:

- ✓ Gestión de la configuración.
- ✓ Gestión de solicitud de cambio.
- ✓ Estado de gestión y medición.

#### **1.4.4 Actividades definidas por normas, modelos y estándares para la GC.**

La ISO sugiere para esta área los siguientes procesos (ISO, 1995) (Villamil, 1999) (Burrows, 1999):

- ✓ Identificación de la Configuración.
- ✓ Control de Cambios a la Configuración.
- ✓ Informe del Estado de la Configuración.
- ✓ Auditoría de la Configuración.

Los procesos asociados a la GCS que aparecen en el estándar de la IEEE se dice que son suficientes para conseguir sus objetivos principales, (Antonio, 2001) (IEEE, 1990) (IEEE, 1987). Estos procesos son los siguientes:

- ✓ Identificación de la Configuración.
- ✓ Control de Cambios en la Configuración.
- ✓ Generación de Informes de Estado.
- ✓ Auditoría de la Configuración.

Por otro lado y con igual propósito en CMMI, para la GC, se proponen las prácticas claves siguientes (Paulk, 1993) (Paulk, 1999a):

- ✓ Planificación de las actividades de Gestión de Configuración.
- ✓ Identificación de los ECS.

- ✓ Control de cambios a los ECS.
- ✓ Informar a los grupos e individuos involucrados de los cambios a los ECS.
- ✓ Auditoría de la Configuración.

Puede apreciarse que existe consenso entre los autores en cuanto a las actividades a incluir dentro de la GCS.

### **1.5 Expediente de Proyecto (EP).**

En la actualidad es evidente que la información se ha convertido en uno de los activos principales de las organizaciones, representando en muchos casos su principal elemento estratégico para la realización de sus objetivos, y como soporte de su actividad. No hay que olvidar que las tecnologías de la información alcanzan desde el conjunto de datos, hasta los elementos humanos, procedimientos de trabajo y de tecnología que de forma coordinada y alineada a una estrategia institucional, proporcionan soporte a la operación, a la toma de decisiones y al servicio de los clientes de una empresa, por lo que evidentemente representan un factor crítico para cualquier organización.

Luego de realizarse un análisis de la documentación relacionada con el EP de la UCI se arribó a la conclusión de que el EP es la herramienta que agrupa y organiza todos los artefactos que se generan durante el desarrollo de software en un proyecto productivo.

La documentación asociada a los proyectos de software y sistemas debe cumplir con algunos requisitos como son:

- ✓ Servir como medio de comunicación entre los miembros del equipo.
- ✓ Servir de repositorio de información que pueda ser utilizado por los ingenieros de sistemas.
- ✓ Proveer información para el control de los planes, cronogramas e hitos en el proceso de desarrollo de software.
- ✓ Definir quién hace y cómo hace las actividades específicas del desarrollo.



Satisfacer estos requerimientos requiere de un grupo de documentos que no son solamente las documentaciones técnicas, asociadas al producto. No es posible organizar la documentación de los proyectos sin definir un esquema para la organización que norme como usar y evaluar la documentación según el tipo de proyecto.

### **1.5.1 Expediente de proyecto definido por calidad UCI**

La UCI se ha convertido, en muy poco tiempo, en centro de referencia para la Industria Cubana del Software. En apenas 5 años sorprendería ver el número de proyectos que han nacido y crecido en la misma. El modelo productivo que caracteriza la producción en la UCI es otro elemento distintivo. La vinculación *Formación - Producción - Investigación* debe tenerse en cuenta en todo modelo, procedimiento o plantilla que se defina. Todo esto ha impuesto un reto en cuanto a la organización de la producción con la máxima de que la cantidad no puede afectar la calidad. Apoyados en la definición de calidad de Pressman (Pressman, 2005), “concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y de las características implícitas que se espera de todo software desarrollado profesionalmente” y en aras de lograr la calidad tanto en los procesos como en los productos se trabajó en la estandarización de los procesos de desarrollo de software y la documentación que en estos se genera.

En cualquier caso sería evidente la necesidad de querer documentar suficientemente el producto y su proceso de producción. Por tanto, desde que un proyecto se inicia debe abrirse un expediente que recoja toda la documentación necesaria para garantizar la calidad del proceso de producción del software como del producto en sí.

En la UCI no existía una homogeneidad en la documentación que se construía por parte de los proyectos, lo que dificultaba la retroalimentación que pudiera obtenerse de la misma, así como su reutilización por parte de otros proyectos. Como parte de los mecanismos que permiten mejorar la calidad de los procesos de desarrollo de software, se encuentran, estandarizar la documentación de todos los proyectos que se realizan, facilitar la formación y adiestramiento de los equipos de desarrollo en el uso de modelos y estándares propios, para elevar la calidad del producto entregado al cliente, se diseñó un EP que debía implantarse en cada uno de los proyectos que se estuviesen desarrollando en la universidad. Este esquema de expediente y grupo de plantillas definidas, tiene como objetivo influir en la estandarización de la documentación y la creación de una cultura de calidad en la organización. Estas plantillas para documentar toda la información de

un proyecto no son una obligación usarlas pero si es necesario que estén ahí para cuando un proyecto las necesite y de esta manera estandarizar su uso en la UCI logrando un vocabulario común entre proyectos.

El EP definido por calidad UCI como forma de organización y estandarización de los documentos o artefactos generados en los proyectos productivos fue liberado inicialmente sin realizarse ningún estudio de caso, con el transcurso del tiempo se han ido corrigiendo errores en las diferentes plantillas y se han agregado algunas que inicialmente no se encontraban, el mismo ya se encuentra en su versión 2.0 con nuevos artefactos incluidos para otras metodologías de desarrollo de software como XP <sup>8</sup>y SCRUM.

### **1.5.2 Requerimientos de la documentación asociada a los proyectos de la UCI.**

En una organización de software inmadura el proceso de desarrollo de software es generalmente improvisado, no existen planes rigurosos, sus actividades se enfocan en resolver las crisis que se presentan, carecen de bases objetivas para evaluar la calidad de los productos o para resolver los problemas que surgen. Por lo contrario cuando la organización alcanza cierto grado de madurez posee una gran habilidad para administrar el proceso de desarrollo y mantenimiento del software, se hacen pruebas y análisis del costo-beneficio para mejorar el proceso, el administrador monitorea la calidad del producto y la satisfacción del cliente, se llevan registros y todos los integrantes están vinculados en el proceso de desarrollo.

La UCI, como prospecto de la nueva universidad cubana tiene un modelo compuesto por tres procesos fundamentales: *Formación - Producción - Investigación*, y este modelo fue uno de los retos de este expediente, lograr sincronizar estos tres procesos. En esto tuvo gran impacto el área de Entrenamiento Organizacional de CMMI. Esta es un área de proceso de Nivel 3 de CMMI, pero a pesar de que este expediente está basado en el Nivel 2 de dicho modelo, se tuvieron en cuenta algunos elementos dentro del Plan de Capacitación.

Para lograr esta organización en los proyectos de desarrollo de software en la UCI se definió una estructura para el EP y un grupo de plantillas para el mismo. Para esto se analizaron varios modelos y estándares que norman la documentación de software como (CMMI) (CMMI, 2002) en su versión 1.2, las normas ISO y los estándares de la IEEE (IEEE, 1990) (IEEE, 1998).

---

<sup>8</sup> Extreme Programming

Otro aspecto a tener en cuenta es la compatibilidad que tiene dicho expediente con el modelo CMMI v1.2 (CMMI, 2002), modelo seleccionado para los procesos de desarrollo de software en la universidad.

Una característica presente en todos los estándares y normas revisados es su no adherencia a una metodología de software específica, por lo que suelen quedar poco específicas las secciones que abordan las áreas de ingeniería. En el expediente se toma como referencia el Proceso Unificado de Rational (RUP) (RUP, 2003) dada su adaptabilidad a casi cualquier proceso de desarrollo que se quiera utilizar y adaptable a una gran variedad de tipos de proyectos aunque se han incluido plantillas para otras metodologías como XP o SCRUM.

Al adaptar el expediente a dicha metodología las principales diferencias se encontraron en las áreas de procesos (AP) de ingeniería, específicamente en las actividades de análisis, diseño e implementación. Otra área fue la referente a la arquitectura, específicamente la arquitectura del software y la arquitectura de información. El resto de las AP son perfectamente compatibles por lo que se tomó la decisión que dejarlas tal y como estaban propuestas dentro del expediente.

CMMI divide sus AP en 4 categorías de las cuales fueron escogidas 3 para el expediente:

- ✓ Ingeniería.
- ✓ Soporte.
- ✓ Gestión de proyecto.
- ✓ Gestión de procesos.

Se escogieron las categorías Ingeniería, Soporte y Gestión de proyecto solamente pues la categoría Gestión de procesos es una categoría que comienza a jugar su papel a partir del nivel 3 de CMMI. Estas clasificaciones fueron las utilizadas para agrupar las plantillas en el EP y se agregó una cuarta categoría para la documentación legal que, dada su naturaleza y lo delicado de la misma no se consideró conveniente mezclar con documentación técnica. El contenido de esta categoría legal no fue definido por la Dirección de Calidad de Software (DCS), de aquí en lo adelante, de la UCI sino por especialistas en esta área.

El modelo de CMMI esencialmente sigue el concepto de “Di lo que haces, haz lo que dices, Pruébalo”. Esto significa que las organizaciones que desean alcanzar el nivel 2 de CMMI deben

documentar varios procedimientos y métodos de Ingeniería de Software, ejecutarlos y ser capaces de probar que están conformes con los métodos y procedimientos mediante registros de calidad y artefactos del proyecto.

La definición e institucionalización del expediente se realizó siguiendo el modelo escalonado definido CMMI por cada uno de sus niveles de madurez iniciando por el nivel 2 incluyendo algunas exigencias del nivel 3.

La estructura completa del expediente se encuentra en estos momentos como se muestra en la figura 2:



**Figura 2: Estructura del actual Expediente de Proyecto**

### **1.5.3 Plantillas definidas para el expediente de proyecto**

Para cada área tenida en cuenta en el EP se definieron plantillas que permiten estandarizar la documentación de los proyectos de la UCI. La selección de las plantillas se realizó a partir de la revisión de las propuestas por la IEEE (IEEE, 1990).

(Ver Anexo 1 – Plantillas del Expediente de Proyecto).

Todas estas plantillas seleccionadas para conformar el expediente están adaptadas al entorno de la universidad. Utilizan la terminología manejada por los proyectos, se describen los epígrafes y algunos nombres se encuentran adaptados a los artefactos manejados por los proyectos.

Como en la universidad se construyen otros artefactos que no se encuentran formalmente en las normas y estándares internacionales, en este expediente, luego de realizarse una revisión en los proyectos se llegó a un consenso para estos artefactos, fácil de asimilar por todos los proyectos.

Ejemplos de estos documentos son:

Plantilla DCS Evaluación de Áreas de la Organización.

Plantilla DCS Arquitectura de Información.

Plantilla DCS Informe del Levantamiento de Información para la Arquitectura de Información.

Plantilla DCS – Presupuesto.

### **1.5.4 Gestión de Configuración en el expediente de proyecto de la UCI.**

Vale decirse que para planificar un proceso de GCS para un proyecto se necesita comprender el contexto de la organización y la relación entre los distintos elementos de la misma. La GCS interacciona con otras actividades o elementos de la organización.

Para prevenir confusión acerca de quién debe realizar tareas de la GCS determinadas, se deben identificar claramente las organizaciones involucradas en el proceso de la GCS. Responsabilidades específicas para una actividad o tarea de la GCS también deben ser asignadas a organizaciones o personas, bien por título o por elemento de la organización. Se debe identificar también la autoridad general y las vías de información de la GCS, aunque se podría realizar como parte de la fase de planificación de la garantía de la calidad o al nivel de la gestión de proyectos. (SWEEBOK, 2005).

La planificación de un proceso de la GCS para un proyecto dado debería ser consistente con el contexto de la organización, las restricciones que sean aplicables, los consejos comúnmente aceptados y la naturaleza del proyecto (por ejemplo, tamaño, lo crítico que sea). Los resultados de la planificación de actividades que se registran en un Plan de GCS.

Planificar, para llevar a cabo la GC ayuda a identificar las necesidades de personal y herramientas que se requieren para realizar las actividades de la GC. Aborda cuestiones de planificación estableciendo las secuencias de tareas de la GC necesarias e identifica sus relaciones con los planes de proyecto y los hitos establecidos en la fase de planificación del proyecto. También se especifican las necesidades de formación requeridas para la implementación de los planes y formación de personal. (SWEEBOK, 2005).

Los resultados de la planificación de la GCS para un proyecto dado se reflejan en un Plan de Gestión de la Configuración del Software, que es un “documento vivo” que sirve como referencia para los procesos de la GCS. El documento se mantiene (o sea, se actualiza y aprueba) según vaya siendo necesario durante el ciclo de vida del software. Al implementar un Plan de Gestión de la Configuración del Software normalmente es necesario desarrollar un conjunto de procedimientos subordinados más detallados, que definirán la forma en que se realizan requerimientos específicos durante las actividades del día a día. (SWEEBOK, 2005).

Con este objetivo en el expediente se agregó una plantilla denominada Plan de GCS la cual define todo el proceso de GC a seguir en los proyectos productivos de la universidad.

Otras plantillas ya existentes en esta área del expediente son las relacionadas con las solicitudes de cambios y los pedidos de cambios, las cuales controlan, como su nombre lo indican actividades que están relacionadas con el control de cambios dentro de la GC.

### **1.6 Conclusiones**

Como se ha visto a lo largo del presente capítulo la GCS juega un papel importante en el proceso de desarrollo de software. Es una disciplina que se encuentra presente en las normas, estándares y modelos de calidad más utilizados a nivel mundial como CMMI, IEEE e ISO. Personalidades de gran renombre en el mundo del software como Humphrey y Pressman han situado a la GCS como una disciplina esencial para el éxito de un proyecto software.

Se ha podido apreciar como el EP vigente en la universidad ayuda a crear una cultura de calidad en la misma, creando un vocabulario común entre proyectos, el mismo no obstruye el proceso de *Formación Producción Investigación* establecido en la UCI por lo que ha constituido un reto adaptar el EP de proyecto a estas condiciones de la nueva universidad cubana.

No existe actualmente un estándar para organizar la información de una institución o empresa de software, cada una por su parte debe definir como documentar los artefactos necesarios generados en un proyecto productivo en ejecución.

## Capítulo 2: Propuesta solución.

### 2.1. Introducción

En este capítulo se escogerán y describirán los indicadores que formarán parte de las plantillas para dar solución a la problemática planteada , se escogieron los procesos que definen CMMI, IEEE e ISO, y además las sugerencias que ofrece la autora Angélica de Antonio en su libro "La Gestión de la Configuración del Software"(2001). Se tuvo en cuenta además las actividades definidas en la plantilla Plan de GC existente en el EP, la misma define una guía para las actividades a realizarse de GC en los proyectos productivos de la universidad. Estas actividades son:

- ✓ Identificación de la Configuración.
- ✓ Control de Cambios.
- ✓ Informes de Estado.
- ✓ Auditorías de la Configuración.

De estas cuatro actividades, la relacionada con el control de los cambios ya fue cubierta en el EP y por ello la investigación se enfocará en las otras tres actividades, aunque no por ello se dejará de mencionar en este capítulo.

Las restricciones y consejos para el proceso de la GC pueden venir de diferentes fuentes. Las normas y procedimientos definidos a nivel corporativo o de la organización pueden tener influencia o prescribir el diseño e implementación de los procesos de la GC en un determinado proyecto. Además, el contrato entre el proveedor y el cliente podría contener estipulaciones que afecten los procesos de la GC. Por ejemplo, podría ser necesaria una Auditoría de la Configuración o podría ser necesario poner ciertos elementos bajo el control de la GC. Cuerpos de regulación externos podrían imponer restricciones a productos de software que se vayan a desarrollar cuando estos puedan afectar potencialmente a la seguridad pública.



## 2.2. Actividades de la Gestión de la Configuración

### 2.2.1. Identificación de la configuración

Varios autores llegan a un consenso que una de las tareas básicas que forma parte de la GCS es la Identificación de la Configuración de Software (Antonio, 2001) (NASA, 1995) (IEEE, 1990) (Febles, 2004), (Rational, 2003), (Gervás, 2002), (IEEE, 1987), llegando a catalogarla como la actividad más importante para el proceso de GCS (Febles, 2004) (Pressman, 1997).

Una definición acerca de la Identificación de la Configuración es la que se nos brinda en (NASA, 1995) donde se dice que la Identificación de la Configuración es el proceso de identificar cada Línea Base a ser establecida durante el ciclo de vida del proyecto y describir los ECS y su documentación, que formarán parte de cada una de las Líneas Base. Una vez que han sido seleccionados los ECS y sus componentes, se debe establecer una forma de identificar y acceder a estos.

La actividad de identificación de la configuración engloba varias actividades:

- ✓ Establecimiento de una jerarquía preliminar del producto software.
- ✓ Selección de los elementos de configuración.
- ✓ Definición de las relaciones en la configuración.
- ✓ Definición de un esquema de identificación.
- ✓ Definición y establecimiento de líneas base.
- ✓ Definición y establecimiento de bibliotecas de software.

Se puede llegar a un consenso general en cuanto a la definición de Identificación de la Configuración de Software en las definiciones expuestas anteriormente, se puede observar que la Identificación de la Configuración de Software consiste en identificar y asignar nombres significativos y consistentes a todos los elementos que forman parte del producto software, hacerlos únicos y que puedan ser accedidos por los involucrados en el proceso de desarrollo de software.

### **2.2.1.1. Línea Base**

La IEEE define una Línea Base como una especificación o producto que ha sido revisado formalmente y se ha aceptado, que más tarde servirá como punto de partida para el desarrollo, y que solo podrá ser cambiado a partir de un proceso formal de control de cambio (IEEE, 1990).

(Pressman, 1997) La Línea Base es un concepto en la GCS que nos ayuda a controlar los cambios sin impedir que se realicen cambios que estén justificados. El mismo autor define que en el contexto de la Ingeniería de Software, una Línea Base es un hito, un hecho memorable en el desarrollo de un software que es señalado por la aprobación de uno o más elementos de configuración, donde la aprobación de los mismos es obtenida a través de una revisión técnica formal (Pressman, 1997).

Varios autores (Febles, 2004), (Racional, 2003), (Antonio, 2001), (Pressman, 1998) abordan un concepto básico dentro de la Identificación de la Configuración de Software y en la GCS en general, el cual lo denominan Línea Base.

Habitualmente las líneas base se ubican al finalizar determinadas fases del proceso de desarrollo, buscando la obtención de dos objetivos (Rancán, 2003):

1. Identificar los resultados de las tareas realizadas durante la fase.
2. Asegurar que se ha completado la fase, contando con elementos consolidados para iniciar la fase siguiente.

Las maneras más generales de instaurar las líneas bases pueden ser de dos formas (Antonio, 2001):

Físicamente: Etiquetando cada ECS y almacenándolos en un Archivo o Biblioteca de Proyecto.

Lógicamente: Publicando un documento de identificación de la configuración, que hace referencia al estado actual del producto en dicho punto del proceso de desarrollo.

### **2.2.1.2. Elementos de Configuración de Software**

Como resultado de la aplicación del proceso de desarrollo, se obtienen y emplean a lo largo del ciclo de vida, un conjunto de elementos que pueden ser clasificados en ejecutables, fuentes, ficheros de datos, documentación, ayudas, entre otros.

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).

Durante el proceso de desarrollo, el número de los ECS al igual que las relaciones que se establecen entre ellos crece a medida que va transitando el proyecto por el ciclo de vida, provocando confusión entre los involucrados en la realización del proyecto (Pressman, 1998).

Según (Antonio ,2001) se pueden considerar como ECS los siguientes componentes:

1. La especificación del sistema.
2. El plan del proyecto software.
3. La especificación de requisitos software.
4. Un prototipo, ejecutable o en papel.
5. El diseño preliminar.
6. El diseño detallado.
7. El código fuente.
8. Programas ejecutables.
9. El manual de usuario.
10. El manual de operación e instalación.
11. El plan de pruebas.
12. Los casos de prueba ejecutados y los resultados registrados.
13. Los estándares y procedimientos de ingeniería de software utilizados.
14. Los informes de problemas.
15. Las peticiones de mantenimiento.
16. Los productos hardware y software utilizados durante el desarrollo.

17. La documentación y manuales de los productos hardware y software utilizados durante el desarrollo.

18. Diseños de base de datos.

19. Contenido de base de datos.

A medida que progresa el proceso de desarrollo, el número de ECS crece rápidamente. La Especificación del Sistema da lugar al Plan del Proyecto y a la Especificación de Requisitos Software. A su vez estos dan lugar a otros documentos. Si simplemente cada ECS produjera otros ECS, no habría prácticamente confusión. El problema aparece cuando entra en juego la variable CAMBIO.

La selección de un número adecuado de ECS es muy importante. El tener demasiados puede provocar un número excesivamente elevado de especificaciones y documentos que al final resulta inmanejable. Sin embargo, el tener pocos ECS puede hacer que no se tenga la suficiente visibilidad sobre el producto.

### **2.2.1.3. Bibliotecas de software**

Otra de las actividades que forma parte de la Identificación de la Configuración de Software es el establecimiento y actualización de las Bibliotecas Software.

Las Bibliotecas de Software son una colección controlada de software y/o de la documentación relacionada al mismo, diseñada para ayudar en el uso, el desarrollo y el mantenimiento del software (Antonio, 2001), (SWEEBOK 2004).

En éstas se almacenan las copias oficiales de los ECS que se encuentran bajo control de cambio. Según (SWEEBOK, 2004) el número de bibliotecas a definir corresponde con el nivel de madurez de los ECS que son colocados en las mismas.

En (Antonio, 2001) se definen un grupo básico de bibliotecas, a ser establecidas:

- ✓ Biblioteca de Trabajo.
- ✓ Biblioteca de Integración.
- ✓ Biblioteca de Soporte al Proyecto.

- ✓ Biblioteca de Producción.
- ✓ Biblioteca Maestra.
- ✓ Repositorio de Software.

### **2.2.2. Control de Cambios**

Según enuncia la 1ra. Ley de la Ingeniería de Bersoff (Bersoff, 1980), “No importa en que momento del ciclo de vida se encuentra el sistema, este va a cambiar y el deseo de cambio persistirá a lo largo de todo el ciclo de vida del proyecto. Pressman añade además que los cambios son un hecho inevitable durante el desarrollo de un software (Pressman, 1998). Es un hecho que los cambios durante el transcurso de un proyecto son inevitables. Ocurren, fundamentalmente, porque los involucrados en los proyectos, con el tiempo, incrementan sus conocimientos y aumentan o varían las necesidades o requisitos (Febles, 2000a) (Febles, 2000b) (Febles, 2000c) (Febles, 2002).

Es la actividad de GC más importante y su objetivo es proporcionar un mecanismo riguroso para controlar los cambios, partiendo de la base de que los cambios se van a producir. Normalmente combina procedimientos humanos y el uso de herramientas automáticas. (Angélica de Antonio, 2001).

Para ello se definió en el EP vigente en la UCI, 2 plantillas para el control de los cambios, la plantilla de Pedido de cambio y la de Solicitud de cambio, las mismas cubren las actividades que se llevan a cabo dentro de la GC para el control de los cambios.

### **2.2.3. Auditoría de la Configuración**

A pesar de que históricamente la auditoría ha sido siempre de cuentas (o financiera), la informática ha pasado a tomar un papel relevante en esta actividad, primero sirviendo como soporte automatizado para su realización, y después, precisamente para verificar el funcionamiento correcto, eficaz y eficiente de la informática.

Esta es otra de las actividades principales en las que se enfocará la presente investigación pues será una actividad candidata a incluir en una de las plantillas a desarrollar en el presente capítulo.

Una auditoría es una verificación independiente de un trabajo o del resultado de un trabajo o grupo de trabajos para evaluar su conformidad respecto a especificaciones, estándares, acuerdos contractuales u otros criterios. La Auditoría a la Configuración es la forma de comprobar que efectivamente el producto que se está construyendo es lo que pretende ser. (Antonio, 2001).

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

Se pueden diferenciar 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 GC 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 se han realizado correctamente.
- ✓ Auditorías: Se realizan en determinados momentos del proceso de desarrollo de software y su objetivo es examinar el producto en su conjunto.

Las revisiones se deben realizar de forma continua, durante todo el proceso de desarrollo, y no sólo al finalizar éste, cuando los problemas ya no tienen solución.

La tarea de revisión implica tres tipos de funciones:

- ✓ Verificar que la configuración actual del software se corresponde con lo que era en fases anteriores. Debe haber correspondencia y trazabilidad entre los elementos de configuración que aparecen en una línea base y los que aparecen en las líneas base que la preceden y que la siguen. La verificación se realiza con respecto a la línea base precedente.
- ✓ Validar que la configuración actual del software satisface la función que se esperaba del producto en cada hito del proceso de desarrollo. La validación se realiza con respecto a los requisitos del sistema.

- ✓ Valorar si una determinada línea base, teniendo en cuenta los resultados de la verificación y validación, y otro tipo de comprobaciones, se debe considerar aceptable o no.

En cuanto a las auditorías, se suelen distinguir dos tipos de Auditorías de la Configuración (Antonio ,2001):

Auditoría Funcional: Cuyo objetivo es comprobar que se han completado todos los test necesarios para el ECS auditado, y que, teniendo en cuenta los resultados obtenidos en los test, se puede afirmar que el ECS satisface los requisitos que se impusieron sobre él.

Auditoría Física: Cuyo objetivo es verificar la adecuación, completitud y precisión de la documentación que constituye las líneas base de diseño y de producto. Se trata de asegurar que representa el software que se ha codificado y probado. Tras la auditoría física se establece la línea base de Producto. Tiene lugar inmediatamente después de haberse superado la auditoría Funcional.

Y aún se puede considerar un tercer tipo de auditoría:

Revisión Formal de Certificación: Cuyo objetivo es certificar que el ECS se comporta correctamente una vez que éste se encuentra en su entorno operativo.

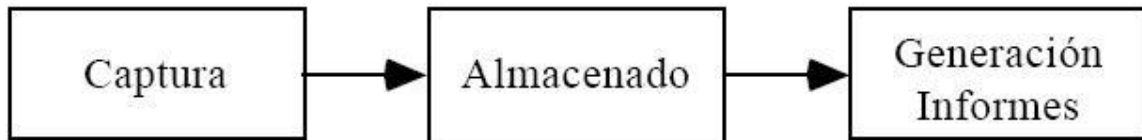
### **2.2.4. Informes de Estado**

Es la última de las cuatro tareas básicas de GC y no por ello la menos importante y será una de las tareas en las que se enfocará la investigación pues resulta un objetivo importante dentro de esta. A veces también denominada Contabilidad de Estado. El objetivo es mantener a los usuarios, a los gestores y a los desarrolladores al tanto del estado de la configuración y su evolución. En definitiva, pretende dar respuesta a la pregunta “¿Qué ocurrió?”, y también a la pregunta “¿Cuándo ocurrió?”. Ayuda también a mejorar los problemas de comunicación entre los participantes en un proyecto. (Antonio, 2001).

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

- ✓ Captura de la información.

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



¿Por qué es importante la generación de informes de estados?

- ✓ La generación de informes de estados es una tarea de suma importancia pues ayuda a que se mantenga la continuidad del proyecto; se trata de permitir que el proyecto siga adelante cuando, por ejemplo, el jefe de proyecto deja la empresa.
- ✓ Evita la duplicidad del trabajo. Si no se guarda información acerca de lo que ya se ha hecho, se puede estar repitiendo el trabajo ya hecho.
- ✓ Evita caer en los mismos errores una y otra vez.
- ✓ Posibilita que se repita aquello que salió bien.
- ✓ Puede ayudar a encontrar las causas de un fallo.

Así pues, los productos de esta actividad pertenecen fundamentalmente a dos categorías:

- ✓ Registros.
- ✓ Informes.

### **2.3.3.1 Registros**

Algunos ejemplos de los tipos de registros que pueden mantenerse en un proyecto productivo son los siguientes (Antonio ,2001):

- a) Registro de elementos de configuración: Conteniendo toda la información relativa a los diferentes elementos de configuración.



Es necesario decidir cuál es el método que se va a utilizar para identificar de forma unívoca cada ECS. Dicho en otras palabras, es necesario establecer un esquema de identificación que permita etiquetar cada uno de los ECS:

- ✓ Número o código del ECS.
- ✓ Nombre del ECS.
- ✓ Descripción del ECS.
- ✓ Autor del ECS.
- ✓ Fecha de creación.
- ✓ Identificación del proyecto al que pertenece el ECS.
- ✓ Identificación de la línea base a la que pertenece el ECS.
- ✓ Identificación de la fase en la que se creó.
- ✓ Tipo de ECS (documento, programa, elemento físico (cintas, discos, etc...)).
- ✓ Localización.
- ✓ Número de versión.
- ✓ Fecha de la versión.

b) Registro de líneas base: Conteniendo toda la información relativa a cada línea base:

- ✓ Nombre.
- ✓ Fecha de establecimiento.
- ✓ Elementos de configuración que la componen.

c) Registro de solicitudes de cambios: El tipo de información que se suele mantener acerca de cada solicitud de cambio es la recogida a través del formulario de solicitud de cambio, incluyendo:

- ✓ Código de solicitud.
  - ✓ Información acerca del solicitante.
  - ✓ Descripción del cambio.
  - ✓ La documentación que apoya la petición de cambio, por ejemplo, una referencia a un informe de incidencia.
  - ✓ La resolución o disposición acerca del cambio (aprobado, aplazado, denegado, entre otros).
- d) Registro de cambios: El tipo de información que se suele mantener acerca de cada cambio es la recogida a través del Informe de Cambio, la Orden de Cambio, el proceso de Gestión de Problemas, etc. Puede contener información acerca de:
1. Solicitud del cambio a la que corresponde.
  2. Evaluación del cambio:
    - ✓ Coste
    - ✓ Esfuerzo
    - ✓ Tiempo
    - ✓ Soluciones alternativas
  3. Plan de implementación del cambio.
  4. Restricciones y criterios de revisión
  5. Impacto sobre la configuración:
    - ✓ Líneas base afectadas.
    - ✓ Elementos de configuración afectados
    - ✓ Versiones afectadas

6. Historia del cambio: Puesto que un cambio es algo que evoluciona, es necesario mantener una historia de cada cambio. En cuanto a los datos que se deben mantener en la historia de un cambio, se pueden considerar:
- ✓ Fecha de la solicitud de cambio.
  - ✓ Fecha de aprobación del cambio.
  - ✓ Fecha de rechazo del cambio.
  - ✓ Fecha de cancelación del cambio.
  - ✓ Fecha de implementación del cambio.
  - ✓ Fecha de cierre del cambio.
- e) Registro de incidencias: El tipo de información que se suele mantener acerca de cada incidencia es la recogida a través del Informe de Incidencia, del tipo:
1. Información del incidente.
  2. Resultado de la Incidencia:
    - ✓ Disposición del CCC.
    - ✓ Número de la solicitud de cambio a la que dio lugar.
    - ✓ Número de formulario de seguimiento de documentación.
    - ✓ Número de notificación de cambio asignada.
  3. Historia:
    - ✓ Fecha de la incidencia.
    - ✓ Fecha de resolución acerca de la incidencia.
    - ✓ Fecha de cierre de la incidencia.
- f) Registro de modificaciones del código: Puede contener información del tipo:

- ✓ Número de identificación de la modificación.
- ✓ Descripción de la modificación.
- ✓ Número de notificación de cambio a la que corresponde (si es aplicable).
- ✓ Número de solicitud de cambio a la que corresponde (si es aplicable).
- ✓ Nombre de los módulos afectados.
- ✓ Versiones modificadas.
- ✓ Persona responsable de la modificación.
- ✓ Fecha de inicio.
- ✓ Fecha de terminación.

g) Registro de modificaciones sobre bases de datos:

- ✓ Número de identificación de la modificación.
- ✓ Descripción de la modificación.
- ✓ Número de notificación de cambio a la que corresponde.
- ✓ Número de solicitud de cambio a la que corresponde.
- ✓ Base de datos modificada.
- ✓ Número de versión modificada.
- ✓ Registros modificados.
- ✓ Persona responsable de la modificación.
- ✓ Fecha de inicio.
- ✓ Fecha de terminación.

h) Registro de modificaciones sobre documentación:

- ✓ Número de identificación de la modificación.
  - ✓ Descripción de la modificación.
  - ✓ Número de formulario de seguimiento de documentación al que corresponde.
  - ✓ Documento modificado.
  - ✓ Número de versión modificada.
  - ✓ Persona responsable de la modificación.
  - ✓ Fecha de inicio.
  - ✓ Fecha de terminación.
- i) Registro de releases y variantes: Su objetivo es describir la composición y estado de una versión liberada del producto:
- ✓ Código de release o variante.
  - ✓ Fecha de liberación.
  - ✓ Elementos de configuración que la componen.
  - ✓ Versión de los elementos de configuración que la componen.
  - ✓ Medio en el que se encuentran.
  - ✓ Diferencias con la release anterior.
  - ✓ Cambios incorporados.
  - ✓ Cambios pendientes.
- j) Registro de instalaciones: Su objetivo es mantener información acerca de todos los lugares en los que se ha instalado un producto software. Puede contener información del tipo:
- ✓ Identificación del producto.

- ✓ Lugar en el que se ha instalado.
- ✓ Fecha de la instalación.
- ✓ Release instalada.

k) Actas de las reuniones del Comité de Control de Cambios:

1. Fecha.
2. Lista de miembros asistentes.
3. Propósito de la reunión.
4. Acciones del CCC:
  - ✓ ECS etiquetados.
  - ✓ Líneas base revisadas/cambiadas.
  - ✓ Disposición de solicitudes de cambio, informes de incidencias.
  - ✓ Notificaciones de cambio, formularios de seguimiento de documentación, informes de cambios, etc.
  - ✓ Líneas base aprobadas.
5. Resultados de las auditorías:
  - ✓ Deficiencias detectadas.
  - ✓ Plan de resolución de las deficiencias encontradas.
  - ✓ Recomendaciones.

### **2.3.3.2 Informes**

En cuanto a los informes, podemos distinguir dos tipos (Antonio, 2001):

- ✓ Planificados.

- ✓ Bajo demanda.

Algunos ejemplos de los tipos de informes que se pueden generar son:

- a) Informe de estado de los cambios: Es un resumen del estado en que se encuentran todas las solicitudes de cambio registradas durante un determinado período de tiempo.
- b) Inventario de elementos de configuración: Para ofrecer visibilidad sobre el contenido de las bibliotecas de proyecto.
- c) Informe de incidencias: Es un resumen del estado en que se encuentran todas las incidencias originadas durante un determinado período de tiempo y las acciones a las que han dado lugar.
- d) Informe de modificaciones: Es un resumen de las modificaciones que se han efectuado en el producto software durante un determinado período de tiempo.
- e) Informe de diferencias entre versiones: Resumen de las diferencias entre las sucesivas versiones de un elemento de configuración.

En cualquier caso, al comienzo de cada proyecto será necesario decidir qué tipo de registros se van a mantener, qué tipo de informes se van a generar y para quién.

Los informes generados pueden ser usados por varios elementos de la organización o del proyecto, incluyendo el equipo de desarrollo, el equipo de mantenimiento, la gestión del proyecto y las actividades de calidad de software. Los informes pueden tener la forma de respuestas inmediatas a preguntas específicas o ser informes prediseñados producidos periódicamente. Alguna de la información producida por las actividades de contabilidad del estado durante el curso del ciclo de vida podrían acabar siendo registros de la garantía de la calidad. Además de informar del estado actual de la configuración, la información obtenida puede usarse como base para varias mediciones útiles para la gestión, desarrollo y GCS. Un ejemplo podría ser el número de cambios pedidos por ECS y el tiempo medio necesario para implementar una petición de cambio. (SWEEBOK, 2005).

### **2.3. Estructura de las diferentes plantillas de la GC.**

#### **2.3.1. Plantilla Listado de Elementos de Configuración.**

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

1. Número o código del ECS.
2. Nombre del ECS.
3. Identificación de la línea base a la que pertenece el ECS.
4. Identificación de la fase en la que se creó.
5. Tipo de ECS (documento, programa, elemento físico (cintas, discos, etc....)).
6. Localización.

Se recomienda que el esquema de identificación deba permitir diferenciar entre las diferentes versiones de un mismo ECS, puesto que los ECS van a evolucionar a lo largo del ciclo de vida. Por ello se recomienda añadir a los anteriores campos el siguiente:

1. Número de versión.

Toda esta información puede ir codificada y agrupada en un único código de identificación o puede ser referenciada como partes separadas.

Otra recomendación obtenida al indagar en algunos proyectos productivos de la facultad fue la de incluir un campo para poner el nombre o el identificador de aquellos ECS que dependen del ECS en cuestión, esto ayudará a identificar la importancia de un ECS en el momento que exista la necesidad de realizarse un cambio sobre este, se tendrá la visibilidad sobre los demás ECS que podrían verse afectados ante un cambio

#### **2.3.2. Plantilla Informe de Estado de la Configuración.**

Como ya se vio anteriormente, los informes de estado según (Antonio, 2001) consisten en informar acerca del estado de los componentes de un producto y de las solicitudes de cambio, recogiendo estadísticas acerca de la evolución del producto.



Además según la propia autora, los productos de esta actividad suelen ser de 2 categorías:

- ✓ Registros.
- ✓ Informes.

Entre los registros que propone la autora en su libro como ya se vieron anteriormente y además se describieron detalladamente se encuentran:

- ✓ Registro de elementos de configuración.
- ✓ Registros de líneas base.
- ✓ Registro de solicitudes de cambio.
- ✓ Registro de cambios.
- ✓ Registro de incidencias.
- ✓ Registro de modificaciones del código.
- ✓ Registro de modificaciones sobre base de datos.
- ✓ Registro de modificaciones.
- ✓ Registro de releases y variantes.
- ✓ Registro de Instalaciones.
- ✓ Actas de las Comisiones del CCC.

Algunas preguntas que se podrían hacer antes de continuar con la investigación:

¿De donde proviene la información necesaria para realizar un Informe de Estado?

Básicamente, la información proviene de las otras tres actividades de GCS y de los registros que se genera en cada una de ellas.

¿Qué información generalmente es necesario capturar?

La cantidad y tipo de información a capturar depende de las características del proyecto, de su tamaño y complejidad. Sin embargo, al comenzar un proyecto nunca se sabe qué información puede llegar a ser útil y cuál no. Por eso, es preferible guardar la mayor información posible. En cualquier caso, el Plan de GCS deberá indicar la información mínima a capturar. Una vez se ha decidido qué información capturar hay que decidir dónde almacenarla. Lo mejor suele ser almacenarla en una base de datos y utilizar herramientas automatizadas para gestionarla. La información contenida en esta base de datos se podrá utilizar para obtener información para generar los informes pertinentes y para facilitar otras tareas como:

- ✓ El análisis postmortem del proyecto.
- ✓ La realización de estimaciones para proyectos futuros.

### **2.3.2.1. Informe de estado de los cambios**

Es un resumen del estado en que se encuentran los cambios y las solicitudes de los mismos en un período de tiempo determinado por lo que valdría tener los siguientes indicadores en el mismo:

- ✓ Tipo de Informe (Planificado o Bajo Demanda).
- ✓ Período de tiempo que se analiza (fecha inicio - fecha fin).
- ✓ Cantidad de solicitudes de cambio realizadas.
- ✓ Cantidad de solicitudes de cambio aplazadas.
- ✓ Cantidad de solicitudes de cambio denegadas.
- ✓ Coste total aproximado de los cambios realizados.
- ✓ Tiempo total aproximado empleado en completar los cambios (en días).
- ✓ Cantidad de líneas bases afectadas.
- ✓ Cantidad de elementos de configuración afectados.
- ✓ Cantidad de versiones afectadas.
- ✓ Cantidad de módulos afectados.

- ✓ Cantidad de veces que un cambio ha sido realizado sobre el código.

Una vez que se tienen todos estos datos contables los integrantes del proyecto, los directivos y todas aquellas personas interesadas podrán tener una visión general del estado del mismo hasta ese período que se analiza y ayudará por consiguiente a la toma de decisiones.

### **2.3.2.2. Inventario de bibliotecas, líneas base y releases.**

Este inventario serviría para ofrecer una visibilidad sobre el contenido de las bibliotecas del proyecto y puede contener indicadores como:

- ✓ Tipo de Informe (Planificado o Bajo Demanda).
- ✓ Período de tiempo que se analiza (fecha inicio - fecha fin).
- ✓ Bibliotecas creadas.
- ✓ Cantidad de elementos de configuración.
- ✓ Cantidad de líneas base.
- ✓ Cantidad de releases.

Vale señalarse que muchas organizaciones también ponen los productos hardware y software utilizados durante el desarrollo bajo control de configuración además de la documentación y los manuales de estos productos. Como los mismos son necesarios para desarrollar el software, deberán estar disponibles cuando se realicen cambios sobre su configuración. Aunque no es normal que haya problemas, es posible que una nueva versión de una herramienta produzca resultados diferentes que la versión usada (un compilador, por ejemplo). Por lo que para la presente investigación también se propone a los productos hardware y software como posibles elementos de configuración y deberán ser tomados en consideración en el momento de realizar una contabilidad sobre ellos. En cualquier caso para cada proyecto concreto se deberá considerar al inicio del mismo que se va a considerar como un elemento de configuración.

### **2.3.2.3. Informe de incidencias**

Es un resumen del estado en que se encuentran todas las incidencias originadas durante un determinado período de tiempo y las acciones a las que han dado lugar y puede contener aspectos tales como:

- ✓ Tipo de Informe (Planificado o Bajo Demanda).
- ✓ Período de tiempo que se analiza (fecha inicio - fecha fin).
- ✓ Cantidad de incidencias detectadas.
- ✓ Cantidad de incidencias a las que se le ha dado solución.
- ✓ Cantidad de solicitudes de cambio a las que han dado lugar las incidencias.
- ✓ Resumen sobre las afectaciones globales de las incidencias sobre el producto.

### **2.3.2.4. Informe de modificaciones**

Es un resumen de las modificaciones que se han efectuado en el producto software durante un determinado período de tiempo y puede contener aspectos tales como:

- ✓ Tipo de Informe (Planificado o Bajo Demanda).
- ✓ Período de tiempo que se analiza (fecha inicio - fecha fin).
- ✓ Cantidad de modificaciones realizadas.
- ✓ Lista de elementos de configuración sobre los que se han realizado modificaciones.
- ✓ Resumen sobre las afectaciones globales a las que han dado lugar estas modificaciones.

### **2.3.2.5. Informe de diferencias de versiones**

Es un resumen de las diferencias entre las versiones y puede contener aspectos tales como:

- ✓ Tipo de Informe (Planificado o Bajo Demanda).

- ✓ Período de tiempo que se analiza (fecha inicio - fecha fin).
- ✓ Cantidad de versiones realizadas hasta la fecha.
- ✓ Resumen del estado en que se encuentra la versión más actual y principales diferencias de las versiones anteriores.

Existen herramientas de control de versiones que se encargan de la identificación de las distintas versiones de cada elemento, de mantener información acerca de las relaciones que hay entre ellas y de su almacenamiento en un repositorio.

### **2.3.2.6. Informe de deficiencias o discrepancias.**

Los productos generados durante el proceso de desarrollo se agrupan, al llegar determinados hitos, en una “línea base pendiente de aprobación”. Seguidamente a esto tiene lugar la revisión de fase. Si en la revisión se encuentran deficiencias en los ECS que componen la línea base, se generan los correspondientes informes y se entregan al CCC, el cual recomienda ciertos cambios sobre la configuración. Este puede contener indicadores tales como:

- ✓ Elementos de Configuración con deficiencias.
- ✓ Resumen de las deficiencias encontradas.
- ✓ Recomendaciones del CCC.

Hasta este punto, luego de haberse analizado y descrito los posibles epígrafes de la plantilla propuesta se puede arribar a las siguientes conclusiones parciales:

Todas las propuestas de informes tienen dos características comunes (tipo de informe y período de tiempo) por lo que pudieran constituir posibles campos generales de la plantilla.

Los Informes de Estado de la Configuración facilitan el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo. Así mismo, permite controlar el sistema como producto global a lo largo de su desarrollo, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de mejora de la organización.

No se podría hablar de una GC completa si no se incluye la generación de Informes de Estado.

### **2.3.3. Plantilla Auditoría de la Configuración.**

Luego de la realización de las revisiones como una de las posibles actividades a realizar propuestas por la autora dentro de la actividad de Auditoría de la Configuración nos quedaría entonces como última actividad las auditorías.

Se pueden definir algunos campos que podría contener la plantilla que se propone, estos pueden ser:

- ✓ Fecha de la auditoría.
- ✓ Nombre del producto que se va a auditar.
- ✓ Nombre(s) de los auditores.

Según la autora Angélica de Antonio la auditoría debe ser realizada por personal ajeno al equipo de desarrollo para lograr mantener la objetividad de la misma, por lo que cabría entonces definir un campo donde iría el nombre del organismo o dirección que va a realizar la auditoría (ej.: Dirección de Calidad UCI).

- ✓ Organismo o Dirección que realizará la auditoría.
- ✓ ¿Que se va a auditar?
- ✓ Propósito de la auditoría.
- ✓ Tipo de auditoría (Funcional o Física).
- ✓ Nombre del responsable de la creación de la línea base o ECS a auditar.
- ✓ Resumen del resultado de la auditoría.

Se recomienda la realización de una revisión cada vez que se culmina una fase o se haga un cambio y una auditoría al finalizar el desarrollo, aunque en el contrato con el cliente se puede definir la realización de alguna auditoría en determinado momento del desarrollo.

Cabe realizar una observación antes de terminar. Una característica de los estándares y normas es su no adherencia a ninguna metodología de desarrollo de software específica, por lo que a veces suelen quedar poco detalladas algunas secciones que en ellas se abordan o son muy generales. La GCS debe adaptarse a la metodología que guía el desarrollo de software escogido para el proyecto que se desarrolla y en todo caso para la organización, en este caso la metodología escogida por la DCS para el EP vigente en la universidad: RUP, aunque se han añadido algunas plantillas para otras metodologías que son menos usadas en los proyectos productivos de la universidad como XP y SCRUM.

### **2.4. Aspectos generales que se incluyen en las plantillas.**

No existen estándares para la forma de organizar la información, cada organización tiene características propias en cuanto al tipo de proyecto que desarrolla, nombres de artefactos, forma de nombrar a los proyectos, etc. Entre los posibles datos a incluir en el informe general y como forma de lograr su compatibilidad con las demás plantillas del EP vigente en la universidad se pudieran incluir:

- ✓ Nombre del Proyecto (Generalmente coincide con el nombre del producto).
- ✓ Versión (Versión actual en la que se encuentra el proyecto).
- ✓ Reglas de confidencialidad (Normas que no deben incumplirse).
- ✓ Clasificación (Confidencial o no).

### **2.5. Conclusiones**

En este capítulo se llevó a cabo la construcción de las plantillas para el EP de la universidad. Para ello se tuvieron en cuenta todas las actividades definidas por modelos, normas y estándares para realizar la GCS en un proyecto. Se describieron los indicadores de cada plantilla siempre con la máxima de mantener cierta compatibilidad con las demás plantillas existentes en el EP en cuanto a su estructura. Se describió además como toda la información relacionada con estas plantillas se obtiene de los registros y control que se mantienen acerca de las diferentes actividades que se realizan dentro de la GCS. Cabe señalar que estos registros son una propuesta de la autora Angélica de Antonio, revisados en su libro “La Gestión de la Configuración”, el mismo fue revisado para generar las plantillas ya existentes en esta área del expediente, por lo que se creyó conveniente usar ese mismo punto de vista para la presente investigación y así de alguna manera

mantener cierta compatibilidad entre las plantillas de esta área, todo esto sin dejar de tener en cuenta otros criterios como los propuestos en CMMI v.12.



### **Capítulo 3: Validación de la propuesta.**

#### **3.1. Introducción**

Para validar la propuesta anterior mediante su aplicación a los proyectos productivos de la Universidad es necesario un conjunto de datos que no están disponibles debido al poco tiempo con que se contaba para el desarrollo de la investigación. Además los proyectos productivos debían encontrarse en determinadas fases de desarrollo y en el caso particular de la plantilla de Auditoría de la Configuración, la auditoría debe ser realizada por personal calificado y con un alto grado de conocimiento acerca del tema. En este caso, se determinó que sería conveniente la aplicación de métodos que permitieran obtener información de personas con conocimiento en el tema que se está investigando. La validación del procedimiento propuesto se realizó a través de criterios de especialistas.

En el presente capítulo se hará la descripción de los pasos utilizados en la selección de los especialistas, la elaboración de la encuesta y los resultados obtenidos.

#### **3.2. Selección de los especialistas.**

Se entiende por un especialista a una persona calificada en una materia, que posea habilidad en una actividad, capaz de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones al respecto. Es muy importante contar con especialistas de alta calificación técnica y que tengan conocimientos del área a evaluar, debido a que este hecho determina el valor de sus respuestas.

A pesar de no contar con una gran cantidad de especialistas de muchos años de experiencia en la UCI, debido a que nuestro país está inmerso hoy en la producción de software pero no era así hace 5 años atrás, se realizó un gran esfuerzo para encontrar aquellos que han tenido mayor relación desempeñándose en el tema.

Para continuar con este proceso se definieron tres etapas de peso:

- ✓ Determinar la cantidad de especialistas.
- ✓ Conformar el listado de los especialistas.
- ✓ Confirmar la participación de los especialistas.

Se debe tener en cuenta que ninguno de los especialistas conoce las identidades y respuestas de los otros que componen el grupo, para lograr así que las respuestas de uno no influyan en la de ninguno de los otros y en caso de ser erróneas, no habrá pérdida de su prestigio.

### **3.2.1. Cantidad de Especialistas.**

No existe una norma generalizada para la selección de la cantidad óptima de especialistas que deben configurar el panel. Los investigadores de Rand Corporation (Landetta, 1999) indican que este número debería estar comprendido entre un mínimo de siete y un máximo de treinta.

Se seleccionó una cantidad de 7 especialistas partiendo del anterior criterio y teniendo en cuenta la deficiencia de especialistas que hay en la universidad relacionada con el tema que se investiga.

### **3.2.2. Listado de especialistas.**

Para confeccionar el listado de especialistas se tuvo en cuenta su relación con el tema, su participación en cursos y posgrados sobre GC o su desempeño en roles relacionados con el tema en proyectos productivos, muy importante esta última consideración pues su criterio influiría en los aspectos relacionados con las plantillas propuestas, ya que han interactuado con los artefactos del EP existente en la UCI.

La selección correcta de los especialistas propicia un mayor porcentaje de obtener resultados más claros y una opinión grupal con un alto grado de consenso.

Existe una serie de cualidades propias de estos especialistas seleccionados, que se tuvieron en cuenta en la presente investigación para la confección del listado. A continuación se relacionan:

- ✓ Seriedad.
- ✓ Honestidad.
- ✓ Sinceridad.
- ✓ Responsabilidad.
- ✓ Creatividad.
- ✓ Capacidad de análisis.

### **3.2.3. Participación de los especialistas.**

Una vez conformado el listado, se invitó personalmente a cada especialista elegido para participar en la evaluación. Se les explicó en que consistía la encuesta en general, se les envió la propuesta de solución con anterioridad para que tuvieran tiempo de analizarla, se les explicó el objetivo de la realización de la encuesta, así como el plazo de entrega. Luego de confirmada su participación, se definió el listado final de los especialistas. Culminando de esta forma el proceso de selección, logrando la participación de los 7 especialistas escogidos.

### **3.3. Elaboración de la encuesta.**

Las encuestas se llevan a cabo de manera anónima.

Para la elaboración de las mismas se tuvieron en cuenta los objetivos que debía cumplir la propuesta de solución.

A los especialistas se les facilitó la posibilidad de emitir su criterio y de modificar aspectos que ellos consideraran necesarios y que pudieron ser obviados en las encuestas, además de presentar su opinión a favor o en contra de la propuesta, también se les permitió realizar preguntas que les pudieran surgir al estudiar el documento presentado.

La encuesta establece una serie de preguntas que permiten visualizar la posibilidad real de aplicar la propuesta en los proyectos productivos de la Universidad.

Se estableció una correlación entre las preguntas y los indicadores que se tuvieron en cuenta para comprobar la utilidad, efectividad y completitud de la propuesta.

Para comprobar la utilidad de la propuesta se efectuaron las preguntas 1, 2, 3, 4.

Para comprobar la efectividad de la propuesta se confeccionó la pregunta 5.

Para comprobar la completitud de la propuesta se diseñaron las preguntas 6 y 7.

### 3.4. Resultados de la evaluación

A continuación se describe el análisis estadístico descriptivo realizado sobre los resultados de la encuesta realizada a los especialistas. Se analizaron estos resultados, dividiendo los mismos a partir de la correlación de las preguntas con los indicadores establecidos para la elaboración de la encuesta.

#### Utilidad de la Propuesta

Para determinar la utilidad de la propuesta se evaluaron las preguntas 1, 2, 3 y 4. Del total de especialistas el 85.7 % considera que las plantillas propuestas cubren las necesidades de documentación de la GC.

La casi totalidad de los especialistas considera además que con la realización de un estudio más profundo podrían añadirse nuevos elementos a las plantillas pues existen proyectos en la universidad con otras características que podrían incluirse, ya que estos manejan otros artefactos para la GC.

En la pregunta 2 el 100 % de los especialistas está de acuerdo que con la propuesta se logrará estandarizar la documentación relacionada con la GC y se obtendrá luego una retroalimentación de las mismas.

El especialista 1 considera además que las plantillas son de vital importancia para los proyectos que comienzan para poder realizar un trabajo más eficiente y conciso.

Sobre la pregunta 3 acerca de que si los Gestores de Configuración lograrán mantener un buen control de la documentación una vez que comiencen a usar las plantillas, el 85.7 % de los especialistas considera que si se logrará, el especialista 2 aborda que con lo que reflejan las plantillas es suficiente para un comienzo, y que es necesario usar estas plantillas y ver los resultados que arrojan para incrementar nuevos elementos, si lo requiere.

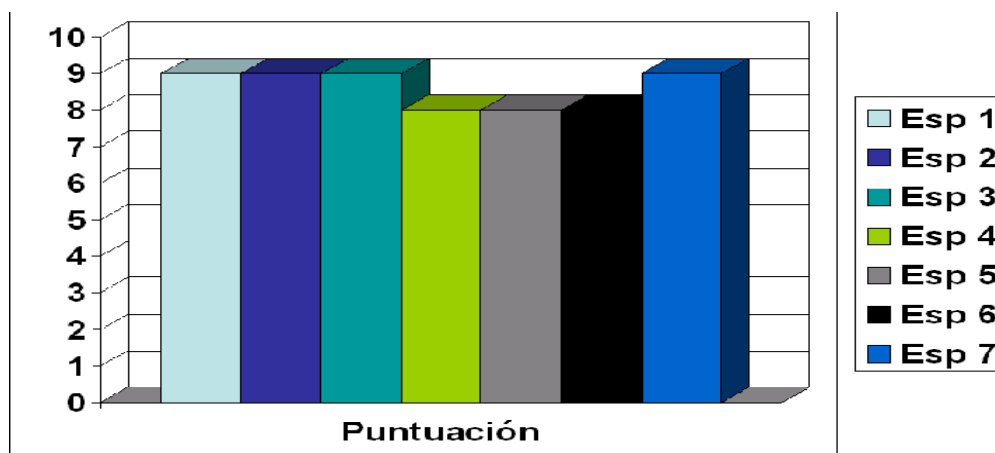
Del total de especialistas el 85.7 % considera que las plantillas cubren las actividades de GC que proponen las normas y estándares más importantes a nivel mundial, aunque el especialista 1 considera que existen otras actividades y elementos en los proyectos productivos que no se encuentran en estos modelos y estándares debido a las características propias de la universidad pero que por el momento la propuesta es un gran paso de avance. El especialista 4 considera que no es bueno ser absolutos pues existen otras propuestas a tener en cuenta.

### Efectividad de la Propuesta

Para comprobar la efectividad de la propuesta se elaboró la pregunta 5 dándole a los especialistas a escoger en un rango entre 1 y 10 (1 el mínimo y 10 el máximo) a partir de 4 criterios acerca de las plantillas, los mismos fueron: satisfacción a las necesidades de los proyectos productivos, adaptabilidad a los proyectos productivos, repercusión en los proyectos productivos y posibilidad de aplicación en los proyectos productivos.

Del total de especialistas, 4 dieron una evaluación de 9 puntos y 3 dieron una evaluación de 8 puntos, lo que se considera una buena puntuación para la propuesta, confirmando que la misma cumple con estos 4 criterios y por lo tanto tendrá la efectividad esperada en los proyectos productivos de la universidad una vez que los Gestores de Configuración o el personal capacitado para usarlas las aplique adecuadamente.

Todo lo anterior se muestra en la gráfica de la figura 3.



### Completitud de la Propuesta

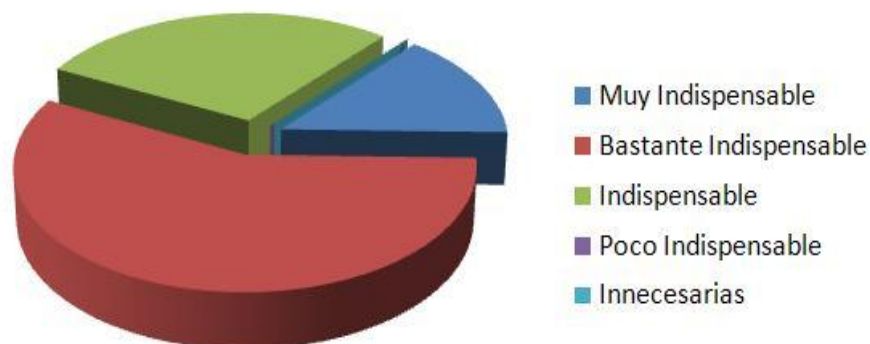
Para comprobar la completitud de la propuesta anterior se confeccionaron las preguntas 6 y 7 donde los especialistas debían escoger entre 5 criterios (Excelente, Muy buena, Buena, Regular, Mala) para la pregunta 6 y (Muy indispensables, Bastante indispensables,

Indispensables, Poco indispensables, Innecesarias.) para la pregunta 7, todo lo anterior, acerca de la estructura de las plantillas propuestas y de la indispensabilidad de las mismas para la universidad.

Del total de especialistas, al responder a la pregunta número 6, 2 de ellos consideran que las plantillas tienen una excelente estructura, 3 una muy buena estructura y 2 que tienen una buena estructura. En la siguiente figura se observa el resultado de forma gráfica.



Al responder a la pregunta 7, 1 especialista evalúa de muy indispensable las plantillas, 4 las evalúan de bastante indispensables y 2 las evalúan de indispensables. En la siguiente figura se muestra gráficamente el resultado de este análisis



Al analizar todo lo anterior se puede apreciar que existe un buen grado de aceptación de la propuesta entre los especialistas en cuanto a los indicadores establecidos para comprobar la efectividad de la propuesta del capítulo 2, por lo que se considera válida dicha propuesta.

### 3.5. Conclusiones

En este capítulo se sometió a criterio de especialistas las plantillas definidas en el capítulo 2. Para esto se siguieron una serie de actividades, para las que se tuvieron en cuenta las capacidades de

los especialistas seleccionados a la hora de emitir su criterio, la cantidad de los mismos y su aprobación de participar en las encuestas.

Para las encuestas se diseñaron una serie de preguntas con el propósito de comprobar los resultados de la investigación y el cumplimiento del objetivo planteado en la misma.

Luego de haberse aplicado las encuestas se realizó un análisis estadístico descriptivo de los resultados arrojados, demostrándose con los mismos el cumplimiento de la hipótesis planteada.

## CONCLUSIONES GENERALES

En el presente trabajo se definieron un conjunto de plantillas para el área de GCS del EP de la universidad, las cuales se consideran que mejoran el control de la documentación relacionada con las actividades de un proceso tan importante como lo es la GCS en los proyectos productivos.

Las plantillas fueron sometidas al criterio de un conjunto de especialistas.

Con la realización de esta investigación se pudo arribar a las siguientes conclusiones:

- ✓ El uso de estándares para la documentación que se genera en los proyectos productivos en las organizaciones le permite a las mismas entre otras cosas: facilitar la formación y adiestramiento de los equipos de proyecto en el uso de modelos y estándares propios, elevar la calidad del producto entregado al cliente, obtener retroalimentación de la documentación para su uso en futuros proyectos y como último crear una cultura de calidad.
- ✓ El aseguramiento de las actividades de la GC permite reducir los costos de los cambios y las posibilidades de fracaso de los proyectos productivos.
- ✓ Con las plantillas definidas se identificaron las actividades principales que deben guiar la GC para un desarrollo satisfactorio de los proyectos productivos de la universidad.
- ✓ El análisis de los resultados obtenidos comprobó la necesidad de implantar estándares en la Universidad en lo referente a la documentación que se genera en los proyectos productivos a partir de las actividades de GC.



## Recomendaciones

Aplicar las plantillas a la mayor cantidad de proyectos productivos de la UCI.

Previo a la aplicación de las mismas:

- ✓ Lograr que sean tenidas en cuenta por la DCS de la UCI a la hora de definir un proceso para documentar las actividades de GC en los proyectos productivos de la universidad.
- ✓ Realizar una capacitación de los Gestores de Configuración de los proyectos productivos de la universidad para el uso de las mismas.

Durante la ejecución:

- ✓ Realizar encuestas entre los Gestores de Configuración de los proyectos productivos de la Universidad para verificar si es necesario añadir nuevos elementos a las plantillas teniendo en cuenta las características de un proyecto productivo en específico.
- ✓ Llevar un proceso adecuado en el llenado de las plantillas por parte de los Gestores de Configuración de los proyectos productivos de la universidad.

Para ampliar la investigación:

- ✓ Realizar un estudio más profundo acerca de las características propias de los proyectos productivos de la Universidad para adecuarlas a las plantillas.
- ✓ Continuar la investigación para enriquecer el EP en esta área, con el objetivo de llevar un control mas riguroso de tan importante actividad como lo es la GC.

**Bibliografía**

(Antonio, A. d. 2001). *La Gestión de la Configuración del Software*.

(Babich W, 1986). *Software Configuration Management*. Addison-Wesley.

(Burrows, 1999) Burrows Clive. Configuration Management, coming of age in the year 2000, Crosstalk Magazine, <http://www.stsc.hill.af.mil/crosstalk/1999/03/burrows.asp>, diciembre del 2005.

(Estrada ,2003). Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. Ciudad de la Habana, Cuba.

(Ema E., 2002). La evolución desde la gestión convencional a la gestión moderna del software (CMM y CMMI), Novotec (Grupo Soluziona).

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

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

Ivar Jacobson, G. B. (2003). *Ayuda del Rational Unified Process*.

(Ivar Jacobson, 2000). *El Proceso Unificado de Desarrollo de Software*, Addison Wesley Longman Inc., 2000.

(ISO, 1995) ISO 10007 Quality management – Guidelines for configuration management, ISO, 1995 abr. 15.

Navarro, A. *Ingeniería del Software*.

Navarro, J. A. (Junio de 2006). Entorno Unificado para la Gestión de Configuración de Software. Ciudad de la Habana, Cuba.

Paulk, M.C. (1993) Capability Maturity Model for Software, Versión 1.1, Carnegie Mellon University, SEI-93-TR-024. 1993.

(Paulk, 1999a) Paulk M. C., Weber C.V., et. al., The Capability Maturity Model, Guidelines for Improving the Software Process, p. 180–191, CMU, SEI, 1999.

(Pressman, R. S. 2005). *Ingeniería del Software. Un Enfoque Práctico*.

(Pressman, 1997) Pressman Roger S., Ingeniería de Software, un enfoque práctico, ed. 3, Mc GrawHill, 1997.

(Pressman 1998) Pressman, R. Ingeniería de Software. Un enfoque práctico. Madrid, Mc Graw-Hill Interamericana de España S.A., 1998.

Rancán, C. J. (Julio de 2003). Gestión de Configuración de Productos Software en Etapa de Desarrollo.

(Astigarraga, 2005). *El Método Delphi*.

(Landeta, 1999) *El Método Delphi*. Barcelona: Ariel Practicum.

(Estrada, A. F. 2003). Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. Ciudad de la Habana, Cuba.

Navarro, J. A. (Junio de 2006). Entorno Unificado para la Gestión de Configuración de Software. Ciudad de la Habana, Cuba.

Delgado, Ramsés M. (Junio, 2006) (Villamil 1999), Villamil Gustavo. ISO 9000 aplicada al software. Tercera parte, Guías de la computación, 1999, p3.

## Anexos

### Anexo 1

#### Plantillas existentes en el EP definido por la DCS de la UCI.

Ingeniería	Gestión de Proyecto	Soporte	Legal
Plantilla DCS Diagrama de Proceso Nombre del Proceso v1.0	Plantilla DCS Plan Desarrollo de Software v1.0	Plantilla DCS Glosario de términos	Plantilla ALBET Acta de Aceptación
Plantilla DCS Especificación de Requisitos v1.0	Plantilla DCS Presupuesto v1.0.	Plantilla DCS Listas de chequeo.	Plantilla ALBET Acta de Entrega.
Plantilla DCS IDEFO Diagram Shapes	Plantilla DCS Lista De riesgos v1.0	Plantilla DCS No Conformidades (ampliada) v1.0	Plantilla ALBET Acta de inicio de proyecto.
Plantilla DCS Modelo de Casos de uso del sistema.	Plantilla DCS Plan Mitigación de Riesgos v1.0	Plantilla DCS No conformidades (reducida) v1.0	Plantilla ALBET Acta de Terminación de Proyecto.
Plantilla DCS Plan De gestión de requisitos v1.0	Plantilla DCS Ambiente de desarrollo v1.0	Plantilla DCS Plan aseguramiento de la calidad v1.0	Plantilla ALBET Carta
Plantilla DCS Especificación de Requisitos v1.0	Plantilla DCS Plan de capacitación.	Plantilla DCS Plan de mediciones v1.0	Plantilla ALBET Indefiniciones
Plantilla DCS Modelo del Negocio v1.0	Plantilla DCS Roles y responsabilidad v1.0	Plantilla DCS Respuestas a No Conformidades v1.0	Plantilla ALBET Informe Técnico
Plantilla DCS Modelo del Dominio v1.0	Plantilla DCS Documento Visión v1.0	Plantilla DCS Solicitud de cambio (reducida) v1.0	Plantilla ALBET Minuta de reuniones
Plantilla DCS - Arquitectura de Información v1.0	Plantilla DCS Diagnóstico v 1.0	Plantilla DCS Pedido de cambio v1.0.	Plantilla ALBET Proyectos Técnicos
Plantilla DCS Documento De Arquitectura de Software v1.0	Plantilla DCS Minuta de reunión v1.0	Plantilla DCS Plan Gestión de Configuración v1.0	
Plantilla DCS Informe del Levantamiento de Información para la			

---

Arquitectura de Información v1.0			
Plantilla DCS Modelo de Diseño v1.0			
Plantilla DCS Diseño casos de prueba v1.0			
Plantilla DCS Plan de pruebas v1.0			
Plantilla DCS Modelo de Despliegue v1.0			

## Anexo 2

## Datos de los Especialistas.

Exp	Vinculado a PP (Rol)	Cursos Sobre el tema.	Graduado de	Años de Graduado	Años vinculados a la UC	Eventos Científicos	Experiencia
Esp 1	Gestor de Config del proyecto SGF.	Ha impartido cursos sobre GC en la fac 3.	Ingeniero en Ciencias Informáticas.	1 año	6 años	1 publicación, Fórum de Ciencia y Técnica 2009	Gestor de Config. Proyecto Fiscalía 9 meses.
Esp 2	Planificadora del Proyecto Sinapsis Planificadora del Polo Gestión Gubernamental.	Ha impartido cursos sobre GC en la fac 3.	Ingeniero en Ciencias Informáticas	2 años	7 años	UCIENCIA 2008 Serie Científica 2009 Publicación en Ilustrados.com	2 años
Esp 3	Sistema de Gestión Penitenciaria (SIGEP).	Ha recibido curso sobre GC.	Ingeniero en Ciencias Informáticas	1 años	6 años	Informática 2007 UCIENCIA 2008	Gestor de Config. 10 meses
Esp 4	-	-	-	-	-	-	-
Esp. 5	Proyecto Aduanas.	Ha recibido Cursos Sobre CMMI y GC.	Ingeniero en Ciencias Informáticas.	1 año	6 años	No	2 años Proyecto Aduanas.
Esp. 6	Polo productivo Auditoría y Control.	Introducción a CMMI por el SEI.	Ingeniero Industrial.	4 años	4 años	No	1 año como asesor de calidad.
Esp. 7	Planificador, Laboratorio de Calidad de Software	Introducción a CMMI por el SEI.	Ingeniero Ciencias Informáticas	2 año	7 años	UCIENCIA 2008 FORUMS	2 años Laboratorio de Calidad de Software

**Anexo 3****Cuestionario para los expertos.**

- 1) ¿Considera usted que las plantillas propuestas cubren las necesidades de documentación para las actividades de Gestión de Configuración en los proyectos productivos de la UCI?

\_\_\_SI \_\_\_No ¿Por qué?

- 2) ¿Cree usted que con las plantillas propuestas se logrará estandarizar la documentación y obtenerse luego una buena retroalimentación relacionada con la Gestión de Configuración en los proyectos productivos de la UCI?

\_\_\_SI \_\_\_No ¿Por qué?

- 3) ¿Con la propuesta de plantillas realizadas, cree usted que los gestores de configuración lograrán mantener un buen control de la documentación donde se plasmen los datos necesarios de las actividades fundamentales de la gestión de configuración en los proyectos productivos de la UCI?

\_\_\_SI \_\_\_No ¿Por qué?

Si considera necesario que se podrían añadir nuevos elementos a las plantillas elaboradas, méncionelo y explique brevemente por qué lo considera así.

- 4) ¿Considera usted que las plantillas propuestas cubren todas las actividades establecidas por algunos de los modelos, normas y estándares más reconocidos en el mundo para la Gestión de Configuración?

\_\_\_SI \_\_\_No ¿Por qué?

- 5) En una escala del 1 al 10 confiera una evaluación a la propuesta según los siguientes criterios:

- Satisfacción a las necesidades de los proyectos productivos y al expediente de proyecto de la Universidad.
- Adaptabilidad a los proyectos productivos.
- Repercusión en los proyectos productivos.
- Posibilidad de aplicación.

\_\_\_Evaluación.

6) En que medida cree usted que las plantillas tienen una buena estructura.

\_\_\_Excelente \_\_\_Muy buena \_\_\_ Buena \_\_\_Regular \_\_\_Mala.

7) ¿Cree que las plantillas definidas son indispensables?

\_\_\_Muy indispensables \_\_\_ Bastante indispensables \_\_\_ indispensables \_\_\_Poco indispensables \_\_\_Innecesarias.



## Anexo 4

### Plantilla Listado de Elementos de Configuración.

#### Listado de Elementos de Configuración

#### 1 Introducción.

*[Incluye un resumen del documento].*

#### 1.1. Propósito.

*[Define el propósito del documento].*

#### 1.2. Listado de elementos de Configuración.

Código	Nombre	Línea Base	Fase de Creación	Tipo de Elemento	Localización	Versión	Elementos de configuración dependientes
<i>Número o código del elemento.</i>	<i>Nombre del elemento</i>	<i>Identificación de la línea base a la que pertenece el elemento (Si es que pertenece a alguna)</i>	<i>Fase del desarrollo en la que se creó.</i>	<i>Tipo de elemento (Documento, Programa, Elemento físico (cintas, discos, etc.))</i>	<i>Biblioteca en la que está ubicado (En caso de ser un elemento físico sería el lugar).</i>	<i>Número de versión.</i>	<i>Nombre o código de los elementos de configuración dependientes del elemento en cuestión.</i>

## Anexo 5

### Plantilla de Auditoría de la Configuración.

#### Auditoría a la Configuración

---

Fecha de la Auditoría.

*[Fecha].*

Nombre del producto.

*[Nombre del producto que se va a auditar].*

Nombre(s) de los auditores.

*[Nombre de los auditores].*

Dirección u organismo.

*[Dirección u organismo que realizará la auditoría].*

Propósito de la auditoría.

Tipo de auditoría

*[Funcional o física].*

Auditado.

*[Nombre del responsable de la línea base o elemento de configuración que se auditará].*

Resumen de la auditoría.

*[Resumen del resultado de la auditoría].*

## Anexo 6

## Plantilla Informe de Estado de la Configuración.

## Informe de Estado de la Configuración

## Índice de Contenidos

1.	<b>Introducción</b> .....	1
1.1	Propósito.....	1
1.2	Registros referenciados.....	1
1.3	Tipo de informe.....	1
1.4	Periodo que se analiza.....	1
2.	<b>Inventario de bibliotecas, líneas bases y releases</b> .....	1
2.1	Bibliotecas creadas.....	1
2.2	Elementos de configuración.....	1
2.3	Líneas bases.....	1
2.4	Releases.....	1
3.	<b>Informes de Estado de los cambios</b> .....	1
3.1	Solicitudes de cambios realizadas.....	1
3.2	Solicitudes de cambios aplazadas.....	1
3.3	Solicitudes de cambios denegadas.....	1
3.4	Coste total aproximado de los cambios realizados.....	1
3.5	Tiempo total empleado en completar los cambios realizados.....	2
3.6	Líneas bases afectadas.....	2
3.7	Elementos de configuración afectados.....	2
3.8	Versiones afectadas.....	2
3.9	Módulos afectados.....	2
3.10	Cambios realizados sobre el código.....	2
4.	<b>Informe de Incidencias</b> .....	2
4.1	Incidencias detectadas.....	2
4.2	Incidencias a las que se les ha dado solución.....	2
4.3	Solicitudes de cambio a las que han dado lugar las incidencias.....	2
4.4	Resumen sobre las afectaciones al producto originadas por las incidencias.....	2
5.	<b>Informe de Modificaciones</b> .....	2
5.1	Modificaciones realizadas.....	2
5.2	Lista de elementos de configuración sobre los que se han realizado modificaciones.....	2
5.3	Resumen sobre las afectaciones de las modificaciones.....	2
6.	<b>Informe de diferencias de versiones</b> .....	3
6.1	Versiones realizadas.....	3
6.2	Resumen del estado de las versiones.....	3
7.	<b>Informe de diferencias o discrepancias</b> .....	3
7.1	Elementos de configuración con deficiencias.....	3
7.2	Resumen de las deficiencias encontradas.....	3
7.3	Recomendaciones del Comité de Control de Cambios.....	3

---



---

 Informe de Estado de la Configuración
 

---

**1 Introducción.***[Incluye un resumen del informe]***1.1 Propósito***[Define el propósito del informe]***1.2 Registros referenciados.***[Lista de registros que se usan para sacar las estadísticas y resúmenes del informe]***1.3 Tipo de informe.***[Planificado o bajo demanda]***1.4 Período que se analiza.***[Fecha inicio - fecha fin]***2 Inventario de bibliotecas, líneas bases y releases.****2. Bibliotecas creadas.**

<i>Código</i>	<i>Nombre</i>	<i>Resumen del contenido de la biblioteca</i>

**2.2 Elementos de configuración.***[Cantidad de elementos de configuración]***2.3 Líneas bases.***[Cantidad de líneas base]***2.4 Releases.***[Cantidad de releases hechas]***3 Informe de estado de los cambios.****3.1 Solicitudes de cambios realizadas.***[Cantidad de solicitudes de cambio]***3.2 Solicitudes de cambios aplazadas.***[Cantidad de solicitudes de cambio aplazadas]***3.3 Solicitudes de cambios denegadas.***[Cantidad de solicitudes de cambio denegadas]***3.4 Coste total cambios realizados.***[Coste en pesos o en horas-hombre]*



---

 Informe de Estado de la Configuración
 

---

3.5 Tiempo total empleado en completar los cambios realizados.

*[Tiempo total]*

3.6 Líneas bases afectadas.

*[Cantidad de líneas bases afectadas]*

3.7 Elementos de configuración afectados

*[Cantidad de elementos de configuración afectados]*

3.8 Versiones afectadas.

*[Cantidad de versiones afectadas]*

3.9 Módulos afectados

*[Cantidad de módulos afectados]*

3.10 Cambios realizados sobre el código

*[Cantidad de cambios realizados sobre el código]*

#### **4 Informe de Incidencias.**

4.1 Incidencias detectadas.

*[Cantidad de incidencias detectadas]*

4.2 Incidencias a las que se les ha dado solución.

*[Cantidad de incidencias a las que se les ha dado solución]*

4.3 Solicitudes de cambio a las que han dado lugar las incidencias.

*[Cantidad de solicitudes de cambio a las que han dado lugar las incidencias]*

4.4 Resumen sobre las afectaciones al producto originadas por las incidencias.

#### **5 Informe de Modificaciones.**

5.1 Modificaciones realizadas.

*[Cantidad de modificaciones realizadas]*

5.2 Lista de elementos de configuración sobre los que se han realizado modificaciones. |

Código	Nombre

5.3 Resumen sobre las afectaciones de las modificaciones.

## Informe de Estado de la Configuración

## 6.1 Versiones realizadas.

<i>Código</i>	<i>Nombre</i>	<i>Cantidad de Versiones</i>
<i>Código del elemento de configuración</i>	<i>Nombre del elemento de configuración.</i>	<i>Cantidad de versiones realizadas.</i>

## 6.2 Resumen del estado de las versiones

**7 Informe de diferencias o discrepancias.**

## 7.1 Elementos de configuración con deficiencias.

*[Cantidad de elementos de configuración con deficiencias]*

## 7.2 Resumen de las deficiencias encontradas.

## 7.3 Recomendaciones del Comité de Control de Cambios.

### **Glosario de Términos.**

#### SEI:

Fundación federal norteamericana para la investigación y desarrollo, cofinanciada por el Departamento de Defensa de los Estados Unidos y dependiente de la Universidad Carnegie Mellon.

#### Release:

Instancia del sistema hecha para ser distribuida a los clientes.

#### Artefactos:

Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

#### DCS:

Dirección de Calidad de Software. Organismo encargado en la UCI de dirigir, organizar, controlar y establecer los procesos de calidad de software para los proyectos productivos de la misma.

#### Proyecto:

Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

#### Plantillas:

Una plantilla es una guía que permite construir un diseño o un esquema predefinido para que quien la utilice se asesore de ella y guarde cierta información que le resulta necesaria documentar.

#### Software:

Son programas de ordenador, procedimientos, y opcionalmente la documentación y los datos asociados que forman parte de un sistema.

#### SCRUM:

El término SCRUM tiene su origen en el ámbito del rugby, se trata de una posición entrelazada en círculo que toman los integrantes de ambos equipos. El propósito del scrum es el de reiniciar el juego, rápida, segura e imparcialmente, después de una infracción menor o de una detención.

SCRUM es una metodología que nace ajena al desarrollo del software, de hecho sus principios fundamentales fueron desarrollados en procesos de reingeniería por Goldratt, Takeuchi y Nonaka en la década de 1980 y no fueron aplicados al proceso de desarrollo de software hasta 1993 por Jeff Sutherland, siendo formalizada con la colaboración de Ken Schwaber en una presentación en OOSPLA 96.