

**Universidad de las Ciencias Informáticas
Facultad 6 “Bioinformática”**



**“Evaluación del Proceso de Gestión de
Configuración de los Proyectos Productivos
de la Facultad 6”**

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

Autores: Leodany Salgado Céspedes

Yoe Laurencio Rodríguez

Tutores: Msc. Maypher Román Durán

Ing. Delvis Echeverría Pérez

Ing. Roig Calzadilla Díaz

“Ciudad de la Habana. Junio, 2008”

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La respuesta es ésta, simplemente: porque aún no hemos aprendido a usarla con tino.”

Albert Einstein

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Leodanys Salgado Céspedes

Firma del autor

Yoe Laurencio Rodríguez

Firma del autor

Ing. Delvis Echeverría Pérez

Firma del tutor

Ing. Roig Calzadilla Díaz

Firma del tutor

Msc. Maypher Román Durán

Firma del Autor

DATOS DE CONTACTO

Tutores:

Msc. Maypher Román Durán

Área: Departamento de Ingeniería y Gestión de Software

Cargo: Asesor Técnico Docente

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

maypher@uci.cu

Ing. Delvis Echeverría Pérez

Área: Facultad 6

Cargo: Instructor Recién Graduado

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

dhecheverria@uci.cu

Ing. Roig Calzadilla Díaz

Área: Dirección de Calidad de Software. Infraestructura Productiva (IP)

Cargo: Instructor Recién Graduado

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

rcalzadilla@uci.cu

AGRADECIMIENTOS

Deseo al concluir este Trabajo de Diploma agradecer al Comandante Fidel Castro Ruz, gracias a su altruismo y liderazgo pude estudiar en la universidad del futuro y materializar mi sueño.

A nuestros tutores Maypher Román, Roig Calzadilla y Delvis Hecheverría, que sin su constante y excepcional asesoría, no hubiese sido posible llegar a este peddño.

Agradezco de manera muy especial a mi Mamá, a mis abuelos, hermanos, tíos y tías, a mis primos y primas, a todos ustedes: mis más queridos familiares, por su preocupación, confianza y apoyo incondicional en este importante paso de mi vida, gracias por formar parte de este maravilloso sueño que ya es historia.

A todos mis amigos que los quiero mucho por siempre.

A los que se me olvidan.

Muchas gracias.

LEODANYS SALGADO CÉSPEDES

Llega uno de los momentos más difíciles de mi carrera, y es el de intentar agradecer a las personas que de una forma u otra me apoyaron e influyeron en mi formación profesional. Si existe alguien que no note su presencia en mis agradecimientos, que no se sienta ofendido porque estará siempre presente en mi memoria. Quiero agradecer a mis padres, ositos, como cariñosamente les digo, por estar siempre en cada paso que he dado de mi carrera, por ser mi ejemplo a imitar y ser mi principal fuente de inspiración para seguir adelante y contribuir con su amor a formar el profesional en

*el que me he convertido. Tengo la virtud de tener una familia excelente, la cual
quiero y respeto mucho.*

*A mis hermanos Yanet Laurencio Rodríguez y Jorge Mario Laurencio Díaz, por
darme su apoyo durante el transcurso de mis estudios universitarios.*

*A mis abuelas Herminia Ferrer López o como le llamo cariñosamente, Vieja Momia
endémica y Pobre y a mi viejita Dulce María Rodríguez Simiat mi Cuchita.*

*Mis agradecimientos a Leodany Salgado, alias El Leo, porque a pesar de no
conocernos al inicio de la carrera, hizo siempre el esfuerzo de cumplir con todo
nuestro trabajo y las relaciones de amistad que hoy tenemos, me apoyó cuando más
lo necesité, gracias dios de tesis, eres un genio.*

*A mi novia Ivette Rosa Teodosio Acosta, por estar siempre a mi lado a largo del
presente curso, dándome todo el amor que siente en su corazón, Mi Chiquitica te
quiero mucho, mucho, mucho.*

*A toda mi familia por siempre confiar en mi capacidad de poder escalar hasta esta
elevación.*

A mi suegra y mi suegro, por siempre tenerme como un hijo.

*A mis tutores Maypher Román, Delvis Echevarria y Roig Calzadilla, que sin la
ayuda de ellos no hubiese sido posible lograr llegar a finalizar el presente Trabajo de
Diploma.*

*Para todos aquellos que por sobre todas las pruebas de la vida, estuvieron presentes,
y me brindaron su apoyo, gracias a Keiler Arnaldo, José Rafael alias El Tommy
Viera, Arturito, René Fuentes, Rolando Fernández quien fue más que mi profe guía
en el movimiento de Alumnos Ayudantes, Yoendris Lacoste alias El Dj, ellos
quienes nunca se incomodaron con mis preguntas constantes ni con mis problemas; y
a todos los de mi apartamento: Cantillo, Yeniel, Percy, Pons, Dayron alias El Loco,
gracias a todos, son maravillosos.*

A mi profesora y amiga Dora Emma Níco. Quien me ayudó en mi primer año en Matemáticas. En fin, a todas las personas que se preocuparon por mi desarrollo profesional desde primero hasta quinto año.

Muchas gracias.

YOÉ LAURENCIO RODRÍGUEZ

DEDICATORIA

Dedico este Trabajo de Diploma a mi hermosa madre Reina Amelia Céspedes González (Mima), por ser mi razón de ser; para ella no me alcanzan las palabras, simplemente lo es TODO.

A mi abuelo (Papi) por quererme tanto y educarme, por ser mi vida, por se mi padre.

A mis hermanos Ari y Letser, porque sin su amor y su fe en mí, nunca hubiese realizado este gran sueño.

A mi grandiosa tía Irma, a Mari, Beto y a Juan, los quiero con el alma.

A Frank, Yumí, Elisa, Migue, Frank Alejandro, Osmany y el (Mono), mis primos del corazón, y mucho más.

A mis amigos, los verdaderos, gracias por ser parte de esta formidable familia.

Y no es que se me olvide nadie, sencillamente dejo para el final a lo más grande y puro de mis seres queridos: a mi abuela (Mami), que aunque hoy no está presente entre nosotros, es para mí la más colosal inspiración y a quien ni dedicándole mi vida recompenso jamás. Para ti abuela, este trabajo y el beso que no tuve tiempo para darte.

Leodanyys Salgado Céspedes

Dedico este trabajo a mi hermana Janet Laurencio, a mis padres Jorge Luis Laurencio Rodríguez y Elizabeth Rodríguez Ferrer por haberme apoyado en cada paso importante de mi vida y que hoy este cúmulo de esfuerzos es el resultado del sueño de cualquier padre: ver a su hijo graduado con un título universitario.

Quiero también dedicar este trabajo al Comandante por haber sido el creador de la primera Universidad creada al calor de la Batalla de Ideas Universidad de las Ciencias Informáticas.

Yoe Laurencio Rodríguez

RESUMEN

Se tiene en cuenta la importancia y necesidad que representa la disciplina de Gestión de Configuración del Software en el desarrollo y obtención de un producto (software) con calidad, y los problemas existentes en los Proyectos Productivos de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI) para definir un procedimiento que hace posible la evaluación del proceso de Gestión de Configuración de dichos proyectos.

Para ello se hizo necesario definir todo un flujo de trabajo junto a las actividades, tareas, artefactos y roles correspondientes. El proceso que se propone toma como referencia el flujo de trabajo brindado por RUP y CMMI para la Gestión de Configuración. Además se realizó una descripción detallada de una serie de plantillas mediante las cuales se llevará a cabo el proceso de evaluación que se propone.

La validación de la propuesta se realizó mediante el método de expertos, o sea, varios expertos en el tema de Gestión de Configuración del Software dan su opinión acerca del procedimiento que se propone y mediante esa opinión, se obtienen gráficos, comentarios y conclusiones.

Con la propuesta realizada se cumple con los requisitos establecidos y metas trazadas, además constituye un procedimiento eficaz y una guía por la cual el grupo de calidad de la Facultad 6 puede de forma sistemática llevar a cabo una auditoría de los Proyectos Productivos.

Palabras Claves

Gestión de Configuración del Software, producto (software), procedimiento, proceso de evaluación.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
INTRODUCCIÓN.....	5
1.1 PROCESO DE DESARROLLO DE SOFTWARE.....	5
1.1.1 <i>Proceso Unificado de Desarrollo de Software</i>	6
1.2 CALIDAD DEL SOFTWARE.....	6
1.2.1 <i>Gestión de la Calidad de Software</i>	7
1.3 GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE.....	7
1.3.1 <i>Principales actividades que conforman la GCS</i>	8
1.3.2 <i>Herramientas que soportan la GCS (Subversion, Rational ClearCase LT, etcétera.)</i>	10
1.4 DEFINICIONES Y CRITERIOS DE NORMAS, MODELOS Y METODOLOGÍAS DE CALIDAD DE SOFTWARE	13
1.5 EVALUACIÓN DE LOS PROCESOS DE SW (A NIVEL TEÓRICO, CONCEPTUAL)	20
1.6 MODELOS PARA EVALUACIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE.....	20
1.7 VALIDACIÓN DE LA SITUACIÓN PROBLÉMICA O ACTUAL	23
CONCLUSIONES.....	30
CAPÍTULO 2 DESCRIPCIÓN DE LA EVALUACIÓN DEL PROCESO DE GESTIÓN DE CONFIGURACIÓN DE LOS PROYECTOS PRODUCTIVOS DE LA FACULTAD 6.....	31
INTRODUCCIÓN.....	31
2.1 PROCESO DE EVALUACIÓN	31
2.1.1 <i>Flujo de trabajo, actividades, tareas y artefactos a generar</i>	32
2.1.2 <i>Entradas, Salidas y procedimientos (o técnicas)</i>	42
2.2 PERSONAL INVOLUCRADO EN EL PROCESO DE EVALUACIÓN	44
2.2.1 <i>Roles y responsabilidades</i>	44
2.3 PLANTILLAS.....	45
2.4 DESCRIPCIÓN DE LAS PLANTILLAS	46
2.4.1 <i>Plantilla Informe de Administración de Configuración</i>	46
2.4.2 <i>Plantilla Informe de Medidas del Proyecto</i>	49
2.4.3 <i>Plantilla Informe de Requisitos de Cambio</i>	51
2.4.4 <i>Plantilla Reporte de Elementos de Configuración</i>	54

2.5 PLANTILLAS A MODO DE ENCUESTAS.....	61
2.5.1 Encuesta Informe de Administración de Configuración.....	62
2.5.2 Encuesta Informe de Medidas del Proyecto	67
2.5.3 Encuesta Informe de Requisitos de Cambio.....	69
2.5.4 Encuesta Reporte de Elementos de Configuración	73
CONCLUSIONES.....	78
CAPÍTULO 3 VALIDACIÓN DEL PROCESO DE EVALUACIÓN.....	79
INTRODUCCIÓN.....	79
3.1 VALIDACIÓN DEL PROCEDIMIENTO PARA LA EVALUACIÓN DE LA GCS	79
3.2 CONCLUSIONES DE LA VALIDACIÓN.....	85
CONCLUSIONES.....	86
CONCLUSIONES GENERALES.....	87

ÍNDICE DE FIGURAS

Figura 1 Proceso de desarrollo de software	5
Figura 2 Gráfica de RUP en dos dimensiones	18
Figura 3 Gráfica recursos en función del tiempo del Proceso Unificado	19
Figura 4 Encuesta a Proyectos Productivos de la Facultad 6	26
Figura 5 Resultados cuantitativos de encuestas a proyectos	26
Figura 6 Gráfica del parámetro 1	27
Figura 7 Gráfica del parámetro 2	27
Figura 8 Gráfica del parámetro 3	28
Figura 9 Gráfica del parámetro 4	28
Figura 10 Gráfica del parámetro 5	29
Figura 11 Gráfica del parámetro 7	29
Figura 12 Flujo de Trabajo	33
Figura 13 Roles y tareas del flujo de trabajo propuesto	34
Figura 14 Artefactos generados por los diferentes roles del flujo de trabajo	35
Figura 15 Actividad Plan de Configuración del Proyecto y Control de Cambio	37
Figura 16 Actividad Monitoreo y Reporte de Estados de Configuración	38
Figura 17 Actividad Crear Auditoría y Entrega de Elementos de Configuración	40
Figura 18 Actividad Administrar Requisitos de Cambio	41
Figura 19 Tareas, artefactos de Entrada y de Salida	42
Figura 20 Tareas y procedimientos o técnicas	43
Figura 21 Roles y responsabilidades	44
Figura 22 Plantilla Informe de Administración de Configuración	49
Figura 23 Plantilla Informe de Medidas del Proyecto	51
Figura 24 Plantilla Informe de Requisitos de Cambio	53
Figura 25 Plantilla Reporte de Elementos de Configuración	59
Figura 26 Tabla Leyenda de Puntuación Cuantitativa	60
Figura 27 Tabla Leyenda de Puntuación Promedio	61
Figura 28 Modelo de Encuesta Informe de Administración de Configuración	66
Figura 29 Modelo de Encuesta Informe de Medidas del Proyecto	69
Figura 30 Modelo de Encuesta Informe de Requisitos de Cambio	72
Figura 31 Modelo de Encuesta Reporte de Elementos de Configuración	78

Figura 32 Tabla Resumen del Porcentaje de Aceptación.....	82
Figura 33 Gráfica porcentaje de aceptación de la validación del parámetro 1.....	82
Figura 34 Gráfica porcentaje de aceptación de la validación del parámetro 2.....	83
Figura 35 Gráfica porcentaje de aceptación de la validación del parámetro 3.....	83
Figura 36 Gráfica porcentaje de aceptación de la validación del parámetro 4.....	84
Figura 37 Gráfica porcentaje de aceptación de la validación del parámetro 5.....	84
Figura 38 Gráfica porcentaje de aceptación de la validación del parámetro 6.....	85

INTRODUCCIÓN

La industria del software ha tenido un desarrollo y un crecimiento vertiginoso a nivel global. A su vez, el aseguramiento de la calidad del software se enfoca en identificar y evaluar los defectos que puedan afectar a dicho software, solo pretende dar confianza en que el producto tiene calidad. En la actualidad el tema de Calidad es totalmente un asunto de competitividad, porque baja los costos, retiene a los clientes y aumenta las utilidades. Se evalúa a través de actividades del Control de la Calidad y utiliza modelos de calidad que permiten hacer comparaciones de resultados y estandarizar el proceso de evaluación de la calidad.

Un software con calidad es actualmente una necesidad, se vive en un mundo cada día más global y telecomunicado, donde un producto sin calidad no tiene relevancia alguna. La calidad del software no es más que el grado de cumplimiento de un producto (software) con los requisitos especificados por el cliente. Es por ello que la Gestión de Configuración se hace imprescindible para controlar el ciclo de vida de los activos de desarrollo de un producto (software) y así poder garantizar la calidad del mismo. La Gestión de Configuración del Software es, como se ha citado anteriormente, la actividad que indudablemente protege al software, identifica, organiza y controla todas las transiciones y modificaciones que sufre el mismo. Su aparición está dada por la necesidad de solucionar problemas de calidad presentados en productos en etapas de pleno desarrollo o ya desarrollados.

El objetivo de sus actividades es establecer y mantener la integridad y coherencia del producto software a fin de facilitar el seguimiento de los cambios que sobre él se implementan, de esta forma se asegura la posibilidad de realizar auditorías de control sobre la evolución de las diferentes configuraciones [1]. Su actividad más importante es llevar la precisa organización del Control de Cambios de un software, es decir, tener un control cabal y detallado de las diferentes versiones que pueda presentar cada elemento de dicho software.

En Cuba en los últimos años ha crecido el interés de usar el capital intelectual como principal fuente económica...pero el desarrollo de software es aun incipiente. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar la Industria del Software. Lo que ha traído consigo la creación de variadas estrategias con el fin de elevar la producción y calidad del software cubano, pero el enfoque teórico y científico no ha sido capaz de llegar a la producción, lo que dificulta mucho el desarrollo pues se está a expensas de una competencia cada vez más exigente en tiempo, costo y

calidad [2]. No obstante, poco a poco se ha adquirido la experiencia necesaria para poder llevar la industria cubana a un peldaño más alto en la gran escalinata del desarrollo del software.

Uno de los centros que se encarga de promover y conservar el avance de Cuba en la producción de programas informáticos es la Universidad de las Ciencias Informáticas (UCI); la cual emerge con el surgimiento de la Batalla de Ideas, como un proyecto futuro, como un proyecto revolucionario que rompe con todos los esquemas de universidad, es un tipo de compenetración Universidad-Empresa, una enseñanza superior productiva. La UCI es un ejemplo del avance que se ha tenido en la industria del software cubano. Está concebida para la formación de profesionales altamente calificados. La informatización del país y la integración del mismo al mercado mundial dependen del buen uso y aplicación de metodologías y modelos de calidad que hacen referencia a la disciplina de Gestión de Configuración del Software, la cual constituye una parte esencial de una buena gestión del proyecto y una práctica formal de la ingeniería del software.

Aun después de 5 años de haber comenzado a funcionar la primera universidad de la Batalla de Ideas, existen problemas reales que frenan el buen desarrollo del software; pero para controlar y eliminar cualquier problema existente, es necesario comenzar desde la raíz del mismo, esto conduce a las diferentes facultades por la que está compuesta el centro.

Actualmente la UCI cuenta con 10 facultades, las cuales se especializan en diferentes perfiles, la Facultad 6 cuyo perfil es la Bioinformática, comprende varios Proyectos Productivos vinculados a importantes polos científicos, entre los cuales están: Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Inmunología Molecular (CIM), entre otros; sin embargo, sus Proyectos Productivos median con varias cuestiones que deben ser resueltas ya que entorpecen el ritmo de confección del producto (software).

Si se tiene en cuenta lo que se plantea anteriormente, a continuación pueden identificarse algunas situaciones que se dan en los diferentes Proyectos Productivos de esa facultad:

- ❖ No todos los proyectos hacen Control de Versiones, lo cual impide llevar un monitoreo eficiente de los cambios que se realizan en el software, de esta forma se limita la gestión de control de versiones de elementos del producto (software).

- ❖ No todos los proyectos llevan a cabo las actividades que se deben desarrollar durante la Gestión de Configuración de Software, lo cual no es más que la omisión de actividades reconocidas internacionalmente en metodologías como buenas prácticas de Ingeniería de Software.
- ❖ Algunos proyectos no determinan sus líneas bases de forma periódica, lo que les impide contar con una metodología probada y funcionalmente correcta.
- ❖ No siempre se efectúa de forma periódica el control de los cambios que se realizan en el software, lo que trae como consecuencia la falta de un registro de las personas que inciden en un determinado cambio en el proyecto o que no se evalúe con la debida prioridad el impacto que podría tener dicho cambio en el proyecto.
- ❖ No existe una guía por la cual el grupo de calidad de la Facultad 6 pueda de forma sistemática llevar a cabo una auditoría de todos los Proyectos Productivos, por lo que no se puede tener una idea clara de la situación general de los proyectos.

A partir de la descripción de la **situación actual**, el **problema científico** queda planteado de la siguiente manera:

- ❖ ¿Cómo evaluar el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6?

Como **objeto de estudio** para el mismo se define: La disciplina de Gestión de Configuración del Software, y así queda el **campo de acción** enmarcado en la evaluación del proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6 de la UCI. Si bien se atiende a lo que anteriormente se expone se esboza el **objetivo general**: Definir un procedimiento para evaluar el Proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6.

Los **objetivos específicos** que se trazaron son:

- ❖ Definir un flujo de trabajo, actividades, tareas y artefactos a emplear durante el proceso de evaluación.
- ❖ Definir los roles involucrados y sus responsabilidades en el proceso de evaluación.
- ❖ Definir las técnicas o procedimientos a emplear durante el proceso de evaluación.

Las **tareas de la investigación** surgen a consecuencia del análisis profundo realizado al objetivo general, dichas tareas son:

- ❖ Investigación de otros modelos de referencias: Six-Sigma, ISO 9000, Modelo de Evaluación SCAMPI, etcétera.
- ❖ Investigación del Estado del Arte de Gestión de Configuración del Software.
- ❖ Análisis del Área de Procesos de Rational Unified Process (RUP) y analizar que actividades proponen ellos para evaluar este proceso.
- ❖ Estudio y selección de las técnicas o procedimientos actuales empleados para llevar a cabo la Gestión de Configuración del Software.
- ❖ Realización de un pesquizaje sobre la situación actual de los Proyectos Productivos de la Facultad 6 con respecto al proceso de Gestión de Configuración del Software.
- ❖ Validación de todo el procedimiento que se propone para evaluar el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6.

Estructura de la tesis:

El siguiente trabajo está compuesto por tres capítulos que comprenden fundamentación teórica, descripción y validación de la propuesta.

Capítulo 1: Se realiza la fundamentación teórica, la cuál encierra todos aquellos conceptos de la Gestión de Configuración del Software (GCS) y que se relacionan estrechamente con la Calidad del Software. Además se tratan definiciones y criterios de las normas, modelos y metodologías de calidad más usadas en el mundo actual, así como ejemplos de modelos para la evaluación de procesos de desarrollo de software.

Capítulo 2: Durante el desarrollo de este capítulo se realiza la descripción de la propuesta. Se define todo el proceso de evaluación y un conjunto de modelos de plantillas para llevar a cabo la evaluación el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6.

Capítulo 3: Se realiza la validación de la propuesta mediante el método de expertos para así poder evaluar el proceso de Gestión de Configuración del Software (GCS).

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

El presente capítulo encierra todos aquellos conceptos de la Gestión de Configuración del Software (GCS) y que se relacionan estrechamente con la Calidad del Software. Además se tratan definiciones y criterios de las normas, modelos y metodologías de calidad más usadas en el mundo actual, así como ejemplos de modelos para la evaluación de procesos de desarrollo de software.

Introducción

En este capítulo se realiza un estudio de la Gestión de Configuración del Software, así como normas, modelos, estándares y metodologías que hacen referencia a esta disciplina de la Ingeniería de Software. También se hace referencia al Proceso de desarrollo y Calidad del Software a nivel global, pero siempre se converge en nuestro país y más específicamente en la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI).

1.1 Proceso de Desarrollo de Software

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales y relativos esencialmente a la naturaleza del producto obtenido.

La definición de proceso de desarrollo brindada por RUP expresa lo siguiente:

“...es un conjunto de pasos ordenados, parcialmente dirigidos para alcanzar un objetivo. En Ingeniería de Software un objetivo es construir un producto de software o mejorar uno existente...” [3].



Figura 1 Proceso de desarrollo de software

En fin, cuando todo el proceso de desarrollo se realiza de forma correcta, se puede lograr en un mínimo de tiempo un producto (software) con toda la calidad requerida, debido a la buena dominación de los riesgos de procesos. Todo esto influye positivamente en la satisfacción y confianza del cliente.

1.1.1 Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. El Proceso Unificado es un proceso de desarrollo de software, es un marco de trabajo genérico que puede especializarse para una gran cantidad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos [4].

El Proceso Unificado del Software define actividades y artefactos como dos de sus cuatro principales elementos, de esta forma es capaz de proporcionar la vía para *utilizar de forma efectiva procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de un proyecto [5].*

1.2 Calidad del Software

En el mundo actual uno de los problemas que se enfrentan en el campo de la Informática es la calidad del software, que no es más que la *concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente [6].*

La calidad de un producto (software) depende completamente de la eficiencia con que se haya desarrollado cada fase del proceso de desarrollo de dicho software, esto *implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software [7].*

Para poder controlar la calidad de un software, primeramente es necesario haber definido cuales serán los indicadores de medición, es decir, se debe haber llegado a una concordancia con los requerimientos del software, ya que los índices medibles que posee el mismo constituyen el cimiento para el control y perfeccionamiento de la calidad. Si no se controla y mide todo el proceso de

Ingeniería de Software aplicado para la confección de un producto (software), entonces no se tiene una forma real para medir si existen mejoras.

1.2.1 Gestión de la Calidad de Software

La Gestión o Administración de la Calidad de Software facilita medir y controlar las mejoras de un proceso y por tanto permite tener una visión general de todo el proceso de fabricación del software. Lo que se define como “conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto software final satisfaga los requisitos dados de calidad y exigidos por el cliente” [8] no es más que la Gestión de la Calidad de Software.

Todo esto tiende a converger en la disciplina de la Ingeniería de Software encargada de controlar y monitorear la evolución de un producto (software) durante cada etapa de desarrollo del mismo, es decir, durante el ciclo de vida del software.

1.3 Gestión de Configuración del Software

La Gestión de la Configuración del software (GCS) es un conjunto de actividades diseñadas para controlar el cambio y de esta forma identificar los productos del trabajo que probablemente cambien, establecer relaciones entre ellos. Definir mecanismos para gestionar distintas versiones de estos productos. Controlar los cambios realizados, y auditar e informar los cambios realizados [9].

El diccionario de términos IEEE en (PIATINI 1996) define Gestión de la Configuración del software (GCS) como el proceso de identificar y definir los elementos de configuración en un sistema, así como controlar la entrega y el cambio de estos elementos a través del ciclo de vida del sistema, almacenar el estado de los elementos de configuración y de las solicitudes de cambio, y verificar la completitud con respecto a los requerimientos especificados [10].

En resumen se puede decir que la Gestión de la Configuración del software (GCS) es una disciplina de la Ingeniería de Software que se encarga de monitorear y controlar el desarrollo del producto (software) desde su etapa inicial hasta que el mismo es entregado al cliente para su futura explotación. Se puede llegar a la conclusión que su principal responsabilidad es el control de cambios.

Es de suma importancia especificar cuales son las diferencias que existen entre el mantenimiento del software y la Gestión de Configuración del Software. *El mantenimiento es un conjunto de actividades*

de ingeniería del software que se producen después de que el software se haya entregado al cliente y esté en funcionamiento. La gestión de configuración del software es un conjunto de actividades de seguimiento y control que comienzan cuando se inicia el proyecto de ingeniería del software y termina solo cuando el software queda fuera de la circulación [9].

De aquí se puede deducir que la Gestión de Configuración del Software no es el conocido mantenimiento del software; se puede decir que la Gestión de Configuración del Software (GCS) son todas aquellas actividades de control y monitoreo que están presentes durante todo el ciclo de vida de un producto (software), sin embargo el mantenimiento solo es la ingeniería final, solo aparece en la etapa final de desarrollo del producto.

La Gestión de Configuración del Software (GCS) se encuentra fuertemente relacionada con muchas otras disciplinas de la Ingeniería de Software, el punto clave que une a todas estas disciplinas son los Elementos de Configuración de Software (ECS) que conforman el producto, ya que son entrada y salida de muchas actividades en el proceso de desarrollo del software.

Conceptos Importantes de Gestión de Configuración del Software

- ❖ Líneas Bases: Se puede decir que es un producto previamente especificado y formalmente revisado, solo mediante procedimientos formales de control de cambios el mismo puede ser modificado, constituye la base para desarrollos posteriores.
- ❖ Elemento de Configuración de Software (ECS): Es un documento o información que surge como consecuencia de todo el complejo proceso que engloba la ingeniería. Un ECS puede ser definido como un conjunto detallado de casos de prueba o como un elemento o componente de un producto (software) desarrollado.

1.3.1 Principales actividades que conforman la GCS

Como actividades de Gestión de Configuración del Software (GCS) se tienen las siguientes:

- ❖ Identificación de elementos u objetos.
- ❖ Control de versiones.
- ❖ Control de cambios.
- ❖ Auditoría de la configuración.
- ❖ Generación de informes de estado de la configuración.

La identificación de objetos en la configuración de software se realiza de forma única y siempre hay que tener en cuenta las relaciones existentes entre ellos, esto constituye la antesala para organizar los elementos mediante un enfoque orientado a objetos y, posteriormente poder controlar y gestionar dichos elementos de configuración.

Se pueden identificar dos tipos de objetos: objetos básicos y objetos compuestos. Un objeto básico es una <<unidad de texto>> creado por un ingeniero de software durante el análisis, diseño, codificación o pruebas. Un objeto compuesto es una colección de objetos básicos y de otros objetos compuestos [9].

La actividad que durante el proceso de desarrollo del software permite gestionar las versiones de los elementos de configuración es el control de versiones, mediante procedimientos y herramientas previamente definidas y acopladas. Al realizar cambios reveladores en uno o varios objetos queda definida una nueva versión del producto (software).

El control de cambios no es más que la combinación de procedimientos y herramientas con el objetivo de proporcionar mecanismos que faciliten el control de cambio. Vale destacar que un proyecto de Ingeniería de Software puede fracasar rápidamente por cambios incontrolados, pero también puede enfrentarse a diversos obstáculos por un excesivo control de cambios. Esta actividad de la Gestión de Configuración del Software (GCS) es imprescindible para el buen desarrollo de un producto (software) y a la vez incómoda.

Los procesos de <<alta>> y <<baja>> implementan dos elementos importantes del control de cambios –control de acceso y control de sincronización–. El control de acceso gobierna los derechos de los ingenieros de software a acceder y modificar objetos de configuración concretos. El control de sincronización asegura que los cambios en paralelo, realizados por personas diferentes, no se sobrescriben mutuamente [9].

Una auditoría de configuración del software complementa la revisión técnica formal porque comprueba características que generalmente no tiene en cuenta la revisión. La auditoría de Gestión de Configuración del Software (GCS) se lleva a cabo independientemente por el grupo de garantía de calidad [9]. Mediante esta actividad de aseguramiento de la calidad del software (SQA) siglas en inglés, se conoce si un cambio influye positiva o negativamente en la calidad del producto (software).

La generación de informes de estado de la configuración es una actividad de vital importancia para el buen desarrollo de un proyecto. La contabilidad de estado, nombre por el cual también se conoce esta tarea de Gestión de Configuración del Software (GCS), se ocupa de dar respuestas a las siguientes interrogantes: *¿Qué pasó? ¿Quién lo hizo? ¿Cuándo pasó? ¿Qué más se vio afectado?* [9].

Se puede llegar a la conclusión que la Gestión de Configuración del Software (GCS) es una actividad que se dedica al control y monitoreo detallado del proceso de desarrollo de software. La misma a su vez, está compuesta por una serie de actividades que son muy importantes para que un producto (software) cuente con la calidad requerida por el usuario. La identificación de elementos es la base para poder gestionar las versiones de los objetos de configuración a través del control de versiones. La fusión de los procedimientos y herramientas se origina con el control de cambios, con el objetivo de garantizar un producto final sólido y con calidad, la auditoría de configuración es la actividad que se encarga de monitorear y asegurar que la realización de cambios se implemente con calidad, y por último los informes de estado, cuya tarea principal es informar sobre cada cambio a personas específicas.

1.3.2 Herramientas que soportan la GCS (Subversion, Rational ClearCase LT, etcétera.)

En la sección se abordan algunas de las herramientas que se emplean en el proceso de Gestión de Configuración del Software (GCS) para el control de versiones de proyectos de software, así como características, ventajas y desventajas de las mismas.

Subversion es un sistema centralizado de control de versiones (VCS, por sus siglas en inglés) que permite realizar el seguimiento de los cambios en archivos empleados en proyectos de software. Basado en el trabajo inicial de CVS (Concurrent Versions System), constituye una implementación más eficiente de este último y ha conseguido desplazarlo en proyectos de software libre y abierto, y en organizaciones empresariales. Al ser un sistema de control de versiones que tomó sus ideas a partir de una implementación previa (CVS), las ventajas son en su mayoría mejoras a éste último:

- ❖ *El seguimiento de la historia de cambios sufridos por los archivos involucra copias y cambios de nombre.*
- ❖ *Las modificaciones en varios archivos y carpetas son atómicas, es decir, si por algún motivo uno de los archivos que son parte de la actualización no puede ser modificado entonces toda la operación es cancelada. Esto permite tener siempre datos consistentes.*

- ❖ *Al realizar operaciones de sincronización entre los datos de los programas clientes y el servidor SVN sólo se transmiten aquellos archivos que han sufrido cambios, no todos los archivos. Esto redundante en un ahorro del uso del tráfico de red.*
- ❖ *La creación de ramas y etiquetas es una operación más eficiente, ya que por cada rama no crea una nueva copia en el depósito de archivos, sino que utiliza un árbol diferencial de cambios con el cual conoce qué archivos pertenecen a una rama. Permite bloquear archivos o carpetas individualmente, para evitar que sean editados por más de un usuario. Esto es usado generalmente en la edición de archivos binarios.*
- ❖ *Seguir con los archivos binarios, es capaz de mantener el control de las diferencias entre archivos binarios, con lo cual consigue ahorrar espacio en el repositorio de archivos.*
- ❖ *Se integra fácilmente a un servidor web Apache, y por tanto, puede utilizar sus opciones para la definición de controles avanzados y de navegación del depósito de archivos vía web.*

Como todo sistema presenta deficiencias que en su mayoría suelen ser detalles técnicos:

- ❖ *El cambio en el nombre de un archivo es interpretado como dos operaciones: copia y borrado de la primera copia. Esto ocasiona que los cambios en el código sean perdidos de manera inadvertida tras múltiples operaciones de combinación de copias.*
- ❖ *Subversion no implementa algunas operaciones administrativas, como la eliminación de registros de algunos datos almacenados en el repositorio, la cual puede hacerse de forma manual y engorrosa.*
- ❖ *Subversion requiere que cada carpeta en el lado del cliente incluya una carpeta oculta ".svn". Estas carpetas impiden la generación de copias distribuibles de una versión individual (se requiere realizar una operación adicional de exportación) y son las causantes de la mayoría de problemas que enfrentan los usuarios de esta herramienta.*
- ❖ *No resuelve el problema de aplicar repetidamente parches entre ramas, por lo cual no facilita el llevar la cuenta de qué cambios se han trasladado. En la versión actual se resuelve a través de mensajes adecuados al momento de registrar los cambios, pero se espera una solución óptima para la versión 1.5 [9].*

Este eficiente controlador de versiones, a pesar de haber surgido por deficiencias de CVS, es poco probable que lo reemplace...

Rational ClearCase es una herramienta de gran fiabilidad para el control de versiones. Resulta ser un sistema ideal para proyectos medianos y complejos. Esta herramienta de Gestión de Configuración del Software posee una serie de particularidades que la hacen ser un software robusto y confiable. Por ejemplo:

- ❖ Cuenta con un versionado de ficheros y directorios.
- ❖ Posee múltiples repositorios independientes.
- ❖ Completa Interfaz en Línea de Comandos.
- ❖ Posee una herramienta gráfica de diferencias y combinaciones para ficheros de código fuente y directorios.
- ❖ Contiene árbol de versiones.
- ❖ Cuenta con un asistente gráfico para resolución de combinaciones no automáticas.
- ❖ Los espacios de trabajo configurables y múltiples en cada máquina.
- ❖ Presenta autenticación de usuarios por Directorio Activo y LDAP.
- ❖ Soporta varias plataformas como: Windows 2000 / XP / 2003 / Vista, Linux y Solaris 9 y superior (SPARC / x86).

Este sistema también contiene carencias, pero estas no influyen en la eficiencia del control de versiones de la Gestión de Configuración del Software (GCS).

- ❖ La ausencia de conjuntos de cambios y entregas atómicos.
- ❖ La falta de una herramienta gráfica de diferencias para imágenes.
- ❖ La no existencia de un árbol de versiones en 3D.
- ❖ La no presencia de un explorador de ramas con gráfico de herencia de ramas.
- ❖ Y en lo que respecta a autenticación y seguridad la ausencia de herencia de permisos.

Representan algunas deficiencias de la herramienta Rational ClearCase, que sin duda continúa como una de las de más prestigio dentro de toda la gama de herramientas de Ingeniería de Software Asistida por Computación (CASE).

Concurrent Version System (CVS) es un sistema de control de versiones muy importante y en el mundo del software libre es la herramienta por excelencia. Se encarga del monitorear y registrar todos los cambios que se realizan en la implementación de un proyecto durante el ciclo de desarrollo de un producto (software). Cuenta con varias ventajas que lo hacen popular en el campo de la Gestión de Configuración del Software (GCS). A continuación se hace referencia a algunas de ellas:

- ❖ *Es OpenSource: disponibilidad del código fuente y posibilidad de colaborar en el desarrollo del producto.*
- ❖ *Desarrollado en Java. Disponibles bajo cualquier plataforma tanto clientes como servidores.*
- ❖ *Es un producto maduro, testado a lo largo de muchos años por múltiples grupos de desarrolladores.*
- ❖ *Robusto y potente. Sus creadores dicen que es “the state of the art” de los sistemas de control de versiones [11].*

Pero como mismo ofrece ventajas que lo hacen un sistema popular para el control de versiones, también presenta deficiencias que imponen un límite al trabajo con esta herramienta. Entre ellas se pueden mencionar:

- ❖ La operación commit en el repositorio no es atómica.
- ❖ Si se renombra o se copia a otro destino un fichero o directorio, el sistema no puede mantener el historial del elemento.
- ❖ El sistema no puede realizar la propagación de cambios entre repositorios.
- ❖ El sistema no puede asignar mensajes de registro por fichero.

Se puede decir que este sistema de control de versiones que fue creado a partir del llamado RCS, es una herramienta sólida que no solo gestiona ficheros sino proyectos de software en toda su complejidad.

1.4 Definiciones y criterios de normas, modelos y metodologías de calidad de software

En la última década se ha desarrollado gran cantidad de métodos orientados hacia la calidad del software. El principal problema al que se enfrentan las empresas productoras consiste en escoger uno dentro del universo de modelos de calidad, de procesos y de técnicas de trabajo. Sin embargo, el éxito en la selección dependerá de que se tengan en cuenta las características de los proyectos, la visión y la cultura de la organización. La diversidad de métodos obedece a la diversidad de organizaciones y tipos de proyectos posibles [3].

Antes de comenzar a adentrarnos en el complejo mundo de la Ingeniería de Software que comprende el empleo de normas, modelos y metodologías que se utilizan para garantizar la calidad del software, se deben puntualizar conceptos y definiciones necesarias para comprender mejor el manejo de buenas prácticas durante el ciclo de vida en general de un producto (software).

Modelo de calidad de software.

Es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos [12].

Estándar de calidad de software.

Requisitos obligatorios empleados y que se deben cumplir para prescribir un acercamiento uniforme disciplinado al desarrollo del software [13].

Normas de calidad de software.

Constituyen un modelo para el aseguramiento de la calidad del software a lo largo de todo el ciclo de vida del mismo. Permiten la puesta en práctica de sistemas de gestión de la calidad en un proyecto.

Metodologías de calidad de software.

Es el conjunto de procesos y métodos que se articulan de manera coherente para realizar un proyecto particular de acuerdo al tipo de problema que se pretende solucionar [13].

ISO 9000

La serie ISO 9000 es un conjunto de cinco normas relacionadas entre sí, son normas genéricas, no específicas que permiten ser usadas en cualquier actividad ya sea industrial o de servicios [14].

Beneficios internos:

- ❖ *Aumento en la productividad.*
- ❖ *Disminución del desperdicio y los defectos de fabricación.*
- ❖ *Mayor satisfacción del personal.*
- ❖ *Mejoramiento continuo.*
- ❖ *Mejora en la rentabilidad del negocio.*

Beneficios externos:

- ❖ *Poseer un Sistema de Calidad reconocido internacionalmente.*
- ❖ *Mayores oportunidades de ingresar a ciertos mercados.*
- ❖ *Mayor satisfacción de los clientes.*

¿Qué otras ventajas trae ISO 9000?

- ❖ *Asegura que los trabajadores tienen el entrenamiento y la información necesaria para realizar su labor correctamente.*
- ❖ *Establece un sistema permanente que:*
 - - *detecta e identifica problemas.*
 - - *busca y elimina sus causas.*
 - - *asegura que no vuelvan a ocurrir [15].*

Al no ser normas académicas son de mucha importancia para el desarrollo e implementación de sistemas que garanticen la calidad del software. Esta norma práctica ofrece múltiples beneficios ya que constituye la plataforma para poder crear un Sistema de Calidad seguro.

CMMI

CMMI (Capability Maturity Model Integration) es un conjunto de modelos elaborados por el SEI que permiten obtener un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización, y describen las tareas que se tienen que llevar a cabo para mejorar esos procesos [16].

CMMI define seis niveles de capacidad para medir los procesos:

0.- Incompleto.

No se realiza el proceso o los objetivos del mismo no son alcanzados.

1.- Ejecutado.

Se ejecuta el proceso y además se logran los objetivos trazados.

2.- Gestionado.

El proceso se ejecuta, pero también se planifica, se revisa y se evalúa para poder comprobar el cumplimiento de los requisitos.

3.- Definido.

El proceso gestionado se ajusta y alinea a la política de procesos y directivas que existen en la organización y en la empresa.

4.- Cuantitativamente gestionado.

Se emplean técnicas cuantitativas para controlar el proceso definido.

5.- Optimizado.

Se revisa y modifica de forma sistemática el proceso cuantitativamente gestionado, para así poder adaptarlo a las metas del negocio.

Este modelo de calidad de software es muy utilizado cuando se quiere obtener de forma rápida y con costos ínfimos, productos complejos que requieren de un desarrollo perfecto. A nivel mundial posee un alto reconocimiento y constituye la vía de solución para muchas y diferentes organizaciones y proyectos que desean hacer uso de las buenas prácticas para el desarrollo del software. A continuación se mencionan algunos beneficios que brinda CMMI que facilitan el desarrollo de productos con excelente calidad:

- ❖ *Se explica de mejor manera el vínculo entre la gestión y sus actividades de ingeniería para sus objetivos de negocio.*
- ❖ *Se amplía el alcance y la visibilidad dentro del ciclo de vida de productos y actividades de ingeniería para permitir que el producto o el servicio responda a las expectativas del cliente.*
- ❖ *Se incorpora el aprendizaje de mejores prácticas a otras áreas como manejo del riesgo.*
- ❖ *Se implementan prácticas más robustas y maduras en la organización.*
- ❖ *Se obtiene un mejor complemento con estándares ISO [17].*

La Ingeniería de Software, Ingeniería de sistemas y el Desarrollo integrado de producto y proceso (IPPD) son disciplinas que comprende CMMI, de ahí si creación y refinamiento, que junto a los beneficios que brinda, hacen de este modelo u marco de trabajo ideal para producir productos (software) con calidad de forma sólida y predecible.

RUP

Actualmente los clientes necesitan software más complejos, esto por el creciente aumento de los computadores más potentes.

Por lo tanto los desarrolladores necesitan un proceso que:

- ❖ *Proporcione una guía para ordenar las actividades de un equipo.*
- ❖ *Dirija las tareas de cada desarrollador por separado y del equipo como un todo.*
- ❖ *Especifique los artefactos que deben desarrollarse.*
- ❖ *Ofrezca criterios para el control y la medición de productos y actividades del proyecto.*

El Proceso Unificado... (RUP), es un proceso de desarrollo de software. Entendiendo por proceso de desarrollo de software:

“El conjunto de actividades necesarias para transformar los requisitos de un usuario en un software” [18].

Rational Unified Process (RUP) es la metodología más usada para el análisis e implementación de sistemas junto al Lenguaje Unificado de Modelado UML. En la Figura 2 se grafican los flujos de trabajo (Modelación de Negocio, Administración de Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Administración de Configuración y Cambio, Administración de Proyecto y Ambiente) y las fases (Concepción, Elaboración, Construcción y Transición) en que se divide todo el proceso de RUP, además de el número posible de iteraciones de acuerdo a la envergadura del proyecto.

Esta metodología orientada a objetos (OO) es única por estar dirigida por casos de uso, herramienta que sirve para poder especificar los requisitos del sistema; centrada en la arquitectura, lo cual permite tener una visión y una perspectiva de todo el sistema; y además es iterativa e incremental, así se puede lograr un balance entre los casos de uso y la arquitectura; estas tres características constituyen los rasgos esenciales de RUP.

Esta metodología se puede emplear en un gran número y diversidad de sistemas de software, así como para las heterogéneas áreas de aplicación en diferentes tipos de organizaciones y dimensiones de Proyectos Productivos, y diferentes niveles de aptitud, todo esto se debe a que es un proceso de software totalmente genérico y eficaz.

Con la Administración de Configuración y Cambio que propone el Proceso Unificado se establecen guías eficientes para desarrollos en paralelo, automatizar la construcción y administrar defectos que puedan surgir durante todo el ciclo de vida de un producto (software), estas guías constituyen una solución para las actualizaciones simultáneas, notificaciones limitadas y generación de múltiples versiones, problemas habituales en Proyectos Productivos.

Muchos son los objetivos de RUP, pero la principal meta que persigue este proceso de Ingeniería de Software es asegurar ante todo el desarrollo y obtención de un producto (software) con extrema calidad dentro del tiempo establecido, con el mínimo de esfuerzo y con un presupuesto razonable y predecible.

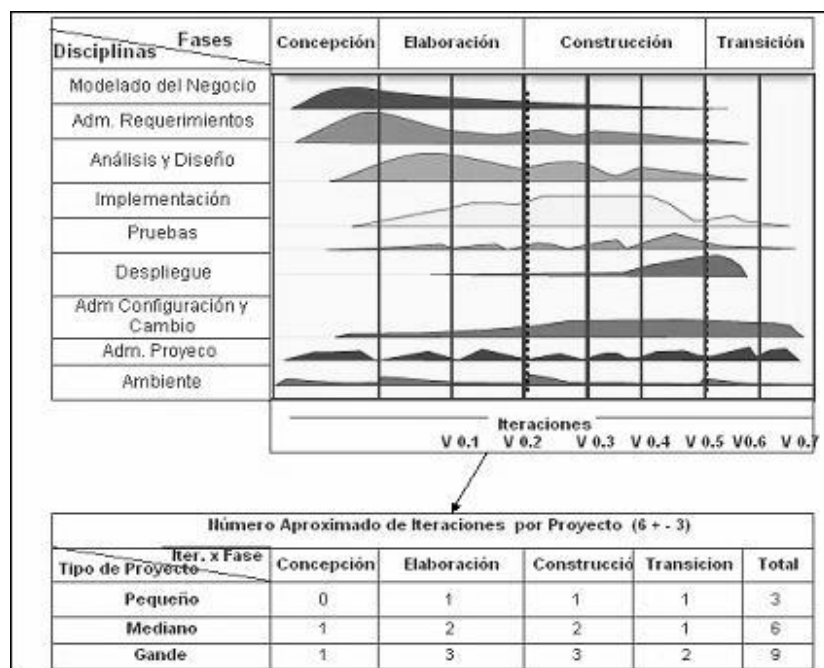


Figura 2 Gráfica de RUP en dos dimensiones

RUP en dos dimensiones presenta:

- ❖ Un eje horizontal que representa las fases y muestra los diferentes flujos de trabajo del ciclo de vida del proceso a lo largo de su desarrollo: (Aspecto Dinámico del Proceso).

- ❖ Un eje vertical que representa los flujos de trabajo y muestra el nivel de aplicación de las disciplinas en cada una de las diferentes fases e iteraciones que componen el proceso: (Aspecto Estático del Proceso).

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas claves antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración.

Los hitos para cada una de las fases son:

- ❖ Inicio – Visión (Objetivos)
- ❖ Elaboración – Arquitectura
- ❖ Construcción – Capacidad operacional Inicial.
- ❖ Transición – Lanzamiento o entrega del producto [18].

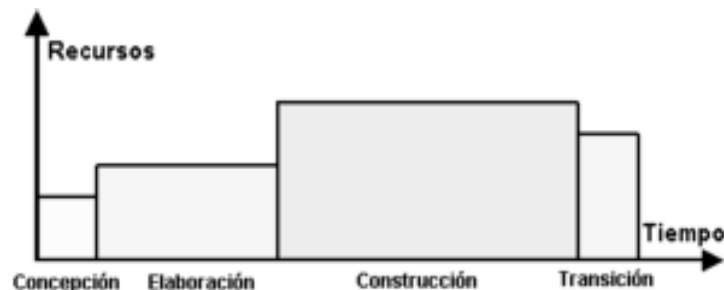


Figura 3 Gráfica recursos en función del tiempo del Proceso Unificado

Beneficios que ofrece esta metodología:

- ❖ Posee mejores prácticas muy bien probadas en la industria del software.
- ❖ Cuenta con un proceso hecho práctico.
- ❖ Se adapta fácilmente a las necesidades de los proyectos.
- ❖ Es una plataforma flexible de procesos de desarrollo de software.
- ❖ Mediante bases de conocimiento, plantillas y herramientas aumenta considerablemente la productividad de los desarrolladores.

MOPROSOFT- Mexico

Modelo de procesos para la industria de Software de México que fomenta la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software [10].

A continuación se mencionan algunas ventajas:

- ❖ Es un modelo fácil de entender y aplicar.
- ❖ Su adopción no resulta costosa.
- ❖ Constituye la base para poder lograr con otros modelos o normas evaluaciones exitosas.

Como se puede apreciar, la adopción de este modelo brinda beneficios que permiten a cualquier organización elevar la capacidad y facilidad para desarrollar productos con calidad reconocida y de esta forma alcanzar niveles internacionales de competitividad en el complejo mundo del software.

1.5 Evaluación de los procesos de SW (a nivel teórico, conceptual)

Cuando se mejora la calidad de los procesos se propicia un aumento de la calidad del producto (software), y a la vez se aumenta la eficiencia de costes y tiempo, la probabilidad de desarrollar con éxito un proyecto, el control de riesgos, y la confianza y satisfacción por parte del cliente.

Cuando se habla de utensilios definidos para estimar y determinar la capacidad de los procesos de una institución, se hace alusión a las evaluaciones de los procesos de software. A través de entrevistas, talleres, estudios o investigaciones que se realizan durante una evaluación, se puede determinar con exactitud, el estado de una organización.

Una evaluación comprende diversos fines, pero se puede decir que el hecho de comprobar la manera en que los disímiles elementos se estimen convenientes a la cadena de producción y verificar si la completa cadena de proceso tiene la capacidad de entregar el resultado esperado con la calidad exigida que requiere todo producto que se desarrolle profesionalmente sin que esta exceda el tiempo y los costes definidos, constituye la meta más importante.

1.6 Modelos para evaluación de procesos de desarrollo de software

Los modelos de evaluación y mejora de procesos, desempeñan un concluyente papel al reconocer, integrar, medir y optimizar el uso de las buenas prácticas existentes para desarrollar un producto (software).

Muchas son las necesidades por las cuales se utilizan los diferentes modelos de evaluación de procesos de desarrollo de software existentes, por ejemplo: minimizar la complejidad y relatividad para lograr la calidad de un software, tratar diversas perspectivas de modelado, gestionar la evolución de los cambios y asegurar su consistencia constituyen algunas de estas necesidades.

Un modelo para evaluación de procesos de desarrollo de software puede definirse como una cuidadosa investigación de los procesos que se usan en una organización para así *determinar la capacidad de esos procesos para ser realizados dentro de los objetivos de calidad, coste y planificación* [19].

SCAMPI

En los últimos años el modelo CMMI (Capability Maturity Model Integration) se ha convertido en el modelo de referencia de procesos como un estándar internacional para la Industria de la Tecnología de Información, es actualmente unos de los modelos con mayor reconocimiento y aceptación a nivel mundial.

SCAMPI (Standard CMMI Appraisal Method for Process Improvement) es el método de evaluación oficial para CMMI desarrollado por el Instituto de Ingeniería de Software de Carnegie Mellon University (SEI) que permite determinar en su ámbito más completo (SCAMPI clase A) el nivel de capacidad y/o madurez de una organización y/o área interna de desarrollo de software [20].

Este método de evaluación ofrece numerosas ventajas:

- ❖ Es capaz de hacer más fácil y entendible el proceso.
- ❖ Puede proveer guías adicionales.
- ❖ Consigue fortalecer la planificación y conducción de la evaluación del proceso.
- ❖ Logra reforzar el reporte de la evaluación del proceso de software.
- ❖ Alcanza definir la durante el periodo de evaluación, la validez del mismo.
- ❖ Logra reforzar los requisitos.

Todos estos beneficios que brinda SCAMPI, lo convierten en un método preciso, eficiente y muy fiable para la evaluación de procesos de desarrollo de software.

Six-Sigma

Metodología rigurosa, integradora y altamente efectiva para generar ahorro, maximizar la creación de valor a los clientes y la rentabilidad a los accionistas.

El nombre Six-Sigma proviene de la medida de variabilidad estadística, conocida como el "desvío estándar", que es la base del sistema [21].

Este modelo consta de cinco fases: Definición, Medición, Análisis, Mejoría y Control, diseñadas para la mejora constante de procesos y proyectos durante todo el ciclo de vida de desarrollo de un producto (software). Además, tiene como objetivo principal proporcionar la mejor información posible para garantizar la calidad del producto (software) durante la implementación del mismo, y a la vez crear un ambiente confortable de comunicación y confianza entre todo el personal involucrado en el desarrollo del producto desde la fase de inicio hasta que el software es entregado al cliente para su futura explotación.

El proceso de la mejora del programa Six-Sigma, se elabora en base a una serie de pasos que se muestran a continuación:

- ❖ *Definir el producto y servicio.*
- ❖ *Identificar los requisitos de los clientes.*
- ❖ *Comparar los requisitos con los productos.*
- ❖ *Describir el proceso.*
- ❖ *Implementar el proceso.*
- ❖ *Medir la calidad y producto [22].*

Actualmente muchas compañías de software en el mundo aplican la metodología Six-Sigma, ya que su empleo ha generado notables avances en lo que refiere a calidad de software.

SPICE

Software Process Improvement and Capability dEtermination (SPICE) siglas en inglés, constituye una norma que trata los procesos de ingeniería y gestión de software, similar a CMM; es un marco de referencia para la evaluación de procesos de software, además se utiliza para mejorar procesos y determinar capacidades.

SPICE provee:

- ❖ *Un marco de referencia para determinar las fortalezas y debilidades de los procesos.*
- ❖ *Un marco de referencia para mejorar los procesos de software, y medir sus mejoras.*
- ❖ *Un marco de referencia para los que adquieren un sistema para evaluar la capacidad de los proveedores de sistemas.*

- ❖ *Un marco de referencia para determinar los riesgos de negocio para una empresa que considera desarrollar un nuevo producto de software o servicio.*
- ❖ *Una ruta para la armonización o migración de los modelos de evaluación de procesos con referencia al modelo de procesos y capacidad [23].*

Cinco son las categorías en que el modelo agrupa a los procesos:

- ❖ Procesos Cliente – Proveedor.
- ❖ Procesos de Ingeniería.
- ❖ Procesos de Proyecto.
- ❖ Procesos de Soporte.
- ❖ Procesos de la Organización.

Como este estándar tiene como propósito mejoras continuas, así como evaluar la capacidad y servir de plataforma para el comercio internacional de software, es acogido por muchas instituciones que están inmersas en el mundo de la competencia por el desarrollo de software con calidad.

1.7 Validación de la Situación Problémica o Actual

Para validar la situación problémica o actual de los Proyectos Productivos de la Facultad 6, se realizó una encuesta a seis proyectos de esa facultad:

- ❖ SAMAD Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias.
- ❖ LIMS Control de la Calidad.
- ❖ SIMDEC Sistema de Manejo de Datos de Ensayos Clínicos.
- ❖ Mapeo Cerebral Humano Cubano.
- ❖ GRAPh-TOol.
- ❖ BioSyS.

ENCUESTA A PROYECTOS PRODUCTIVOS DE LA FACULTAD 6

Nombre: _____

Fecha: _____

() Líder de Proyecto

() Otro: _____

PREGUNTA 1

¿Existe algún especialista con entrenamiento o experiencia formal de (GCS) capacitado para poner en práctica esta disciplina de la Ingeniería de Software?

Si ____

No ____

Parcialmente ____

PREGUNTA 2

¿Conoce y aplica usted las actividades que se deben desarrollar durante la (GCS)? (Ej. Identificación de la configuración, Control de cambios en la configuración, Generación de Informes de Estado, Auditoría de Configuración).

Si ____

No ____

Parcialmente ____

PREGUNTA 3

¿Aplica usted en su proyecto un detallado control de versión del código? (Ej. Establecimiento de una línea base del actual código, Caracterización de cada ítem de configuración, Documentación de cada versión).

Si ____

No ____

Parcialmente ____

PREGUNTA 4

¿Existe en su proyecto un Comité de Gestión de Configuración? (Encargado de establecer líneas bases de software, Revisar y autorizar cambios en dichas líneas, Autorizar la creación de productos a partir de ellas).

Si ____

No ____

Parcialmente ____

PREGUNTA 5

¿Se lleva a cabo en su proyecto un detallado control de los cambios que se realizan en el software?

Si ____

No ____

Parcialmente ____

PREGUNTA 6

¿Qué herramienta utilizan en el proyecto para desarrollar las actividades de (GCS) de manera eficaz?

_____ Ninguna

PREGUNTA 7

¿Existe personal entrenado para ejecutar actividades de (GCS) con las herramientas existentes?

Si ____

No ____

Parcialmente ____

Parámetros	Si	No	Parcialmente
1-Existencia de especialista con experiencia formal de (GCS)			
2-Conocimiento y aplicación de las actividades de (GCS)			
3-Control de Versiones			
4-Existencia de un Comité de Gestión de Configuración			
5-Control de Cambios			
7-Existencia de personal capacitado para el trabajo con herramientas de (GCS)			

Figura 4 Encuesta a Proyectos Productivos de la Facultad 6

Los datos brindados por cada uno de los líderes de esos proyectos revelan los siguientes resultados:

Parámetros	% de aprobación	% de insuficiencia	% de existencia o cumplimiento parcial
1-Existencia de especialista con experiencia formal de (GCS)	0	83.33	16.67
2-Conocimiento y aplicación de las actividades de (GCS)	0	0	100
3-Control de Versiones	0	50	50
4-Existencia de un Comité de Gestión de Configuración	0	100	0
5-Control de Cambios	0	66.67	33.33
7-Existencia de personal capacitado para el trabajo con herramientas de (GCS)	16.67	50	33.33

Figura 5 Resultados cuantitativos de encuestas a proyectos

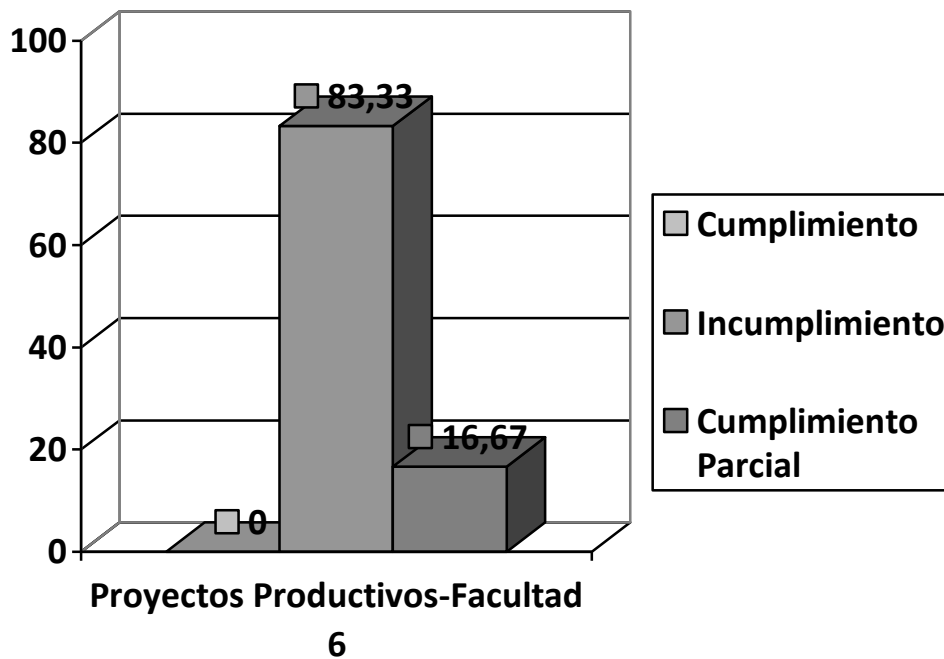


Figura 6 Gráfica del parámetro 1

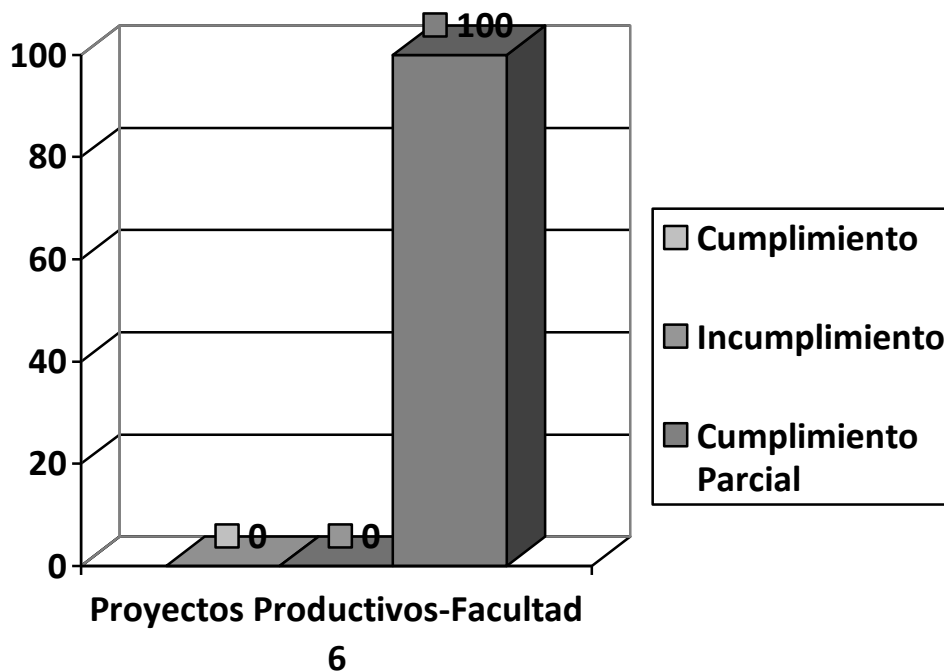


Figura 7 Gráfica del parámetro 2

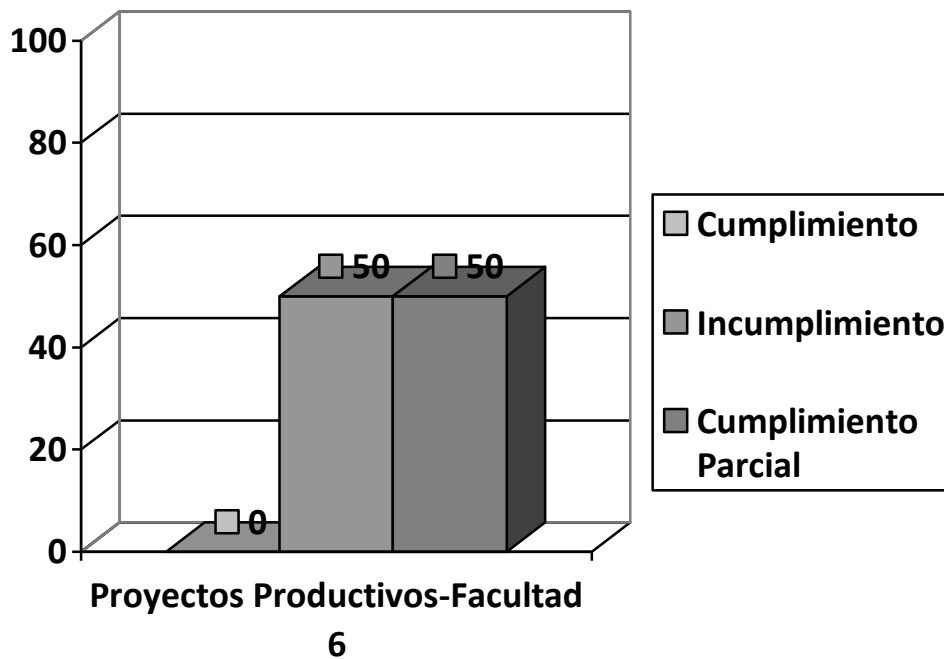


Figura 8 Gráfica del parámetro 3

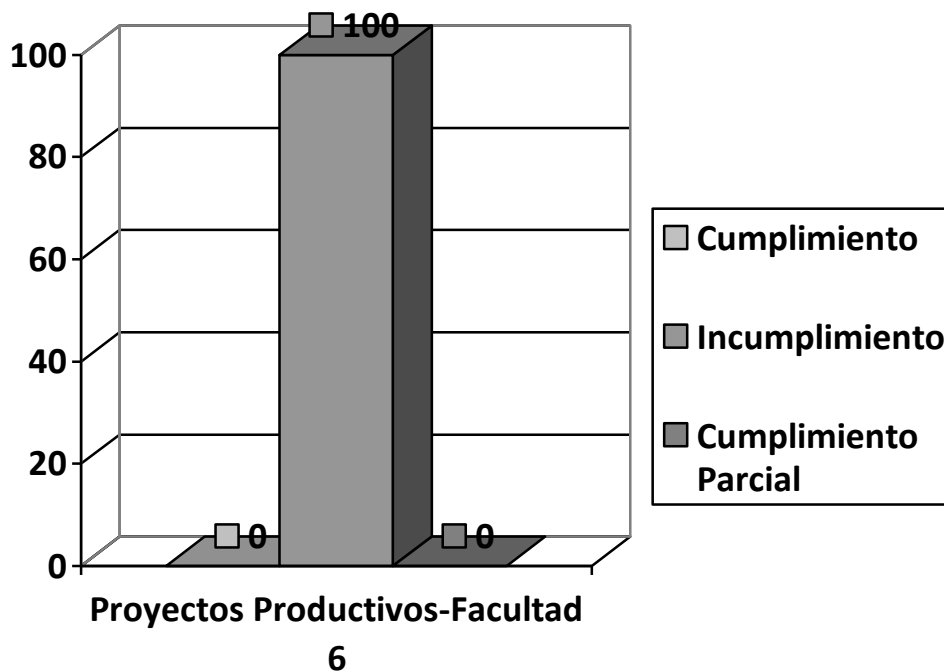


Figura 9 Gráfica del parámetro 4

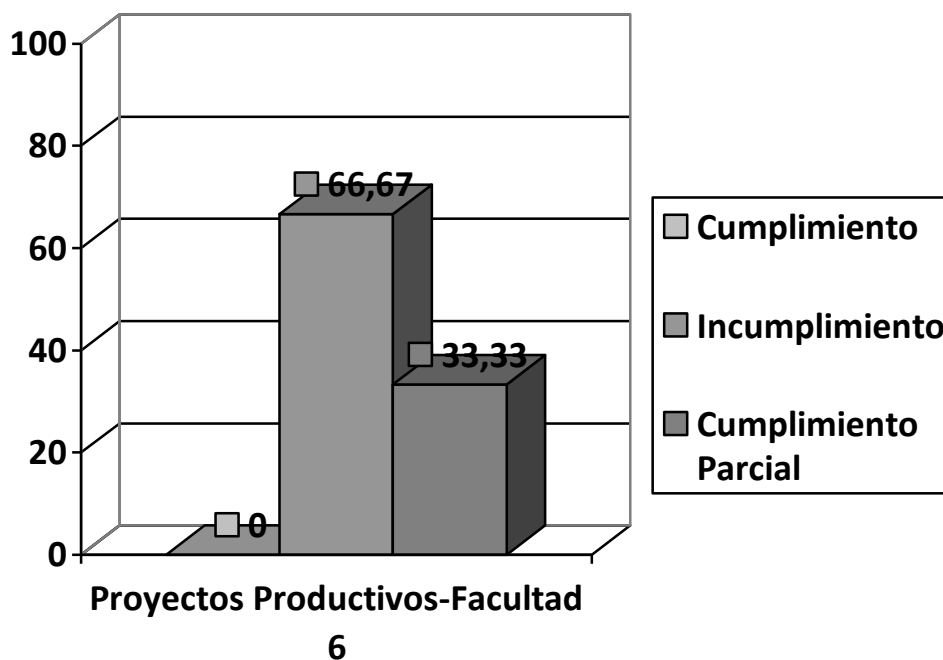


Figura 10 Gráfica del parámetro 5

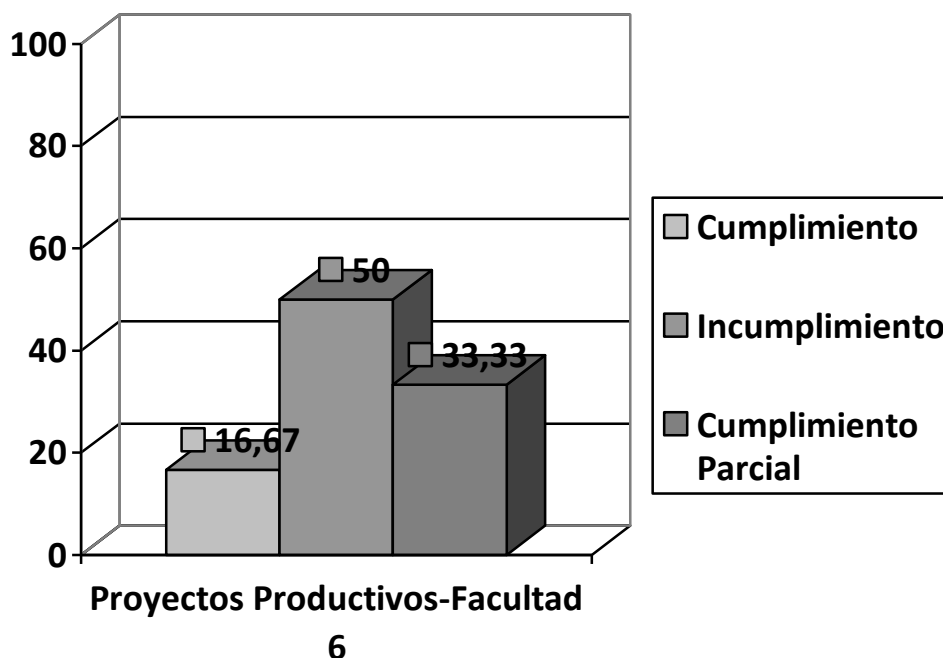


Figura 11 Gráfica del parámetro 7

Conclusiones

Durante el desarrollo de este capítulo se han abordado y descrito temas que hacen referencia a la Gestión de Configuración de Software como disciplina imprescindible para garantizar la calidad de un producto (software), así como normas y metodologías de calidad que se utilizan para llevar a cabo el proceso de desarrollo del mismo.

CAPÍTULO 2

DESCRIPCIÓN DE LA EVALUACIÓN DEL PROCESO DE GESTIÓN DE CONFIGURACIÓN DE LOS PROYECTOS PRODUCTIVOS DE LA FACULTAD 6

Durante el desarrollo de este capítulo se realiza la descripción de la propuesta. Se define todo el proceso de evaluación y un conjunto de modelos de plantillas para llevar a cabo la evaluación el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6.

Introducción

En el capítulo se realiza una descripción gráfica y textual del proceso de evaluación que se propone, además se realiza la descripción del personal involucrado y técnicas que se utilizan durante el desarrollo de dicho proceso, así como la importancia y necesidad del mismo para Proyectos Productivos. Además se brinda una descripción detallada de todos aquellos parámetros que componen las diferentes plantillas que se proponen para la evaluación del proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6. También se describe el sistema de cálculo para la evaluación que se utiliza, así como la medición cuantitativa y cualitativa de cada parámetro.

2.1 Proceso de Evaluación

Un proceso de evaluación constituye una oportunidad invaluable para tener el control y garantizar el buen desarrollo de cualquier proyecto, así como tener una idea clara y precisa del nivel de madurez con que se realizan las diferentes actividades que se llevan a cabo durante el mismo. A través de este proceso se pretende conocer qué tanto un proyecto ha logrado consumir sus objetivos o bien qué tanta capacidad tendría para cumplirlos. Además permite la rectificación de errores y la planificación correcta de estrategias para el perfeccionamiento, por lo que puede ser considerada una actividad encauzada a mejorar la eficacia de los Proyectos Productivos en relación con sus metas. Finalmente se puede precisar y afirmar que la evaluación no es un fin en sí misma, sino un medio para optimar la Gestión de Configuración de Software de los Proyectos Productivos.

La evaluación se realiza con el objetivo de conocer el nivel de cumplimiento de todas aquellas metas que el proyecto establece previamente. La evaluación puede ser clasificada de tres formas diferentes si se tiene en cuenta el momento en que se efectúa la misma:

Evaluación previa: se realiza para determinar la situación inicial de un producto (software) o proyecto. La envergadura del problema se conoce a través de sistemas de evaluación cuantitativos y cualitativos, lo que hace posible seleccionar todos los mecanismos y recursos necesarios e indispensables para satisfacer las necesidades.

Evaluación media: se realiza para determinar si el producto (software) o proyecto se rige por las buenas prácticas de Ingeniería de Software y si los recursos que se ponen a disposición del equipo de trabajo son utilizados según lo programado. El principal objetivo de esta evaluación es detectar todos aquellos problemas que influyen directamente en el ritmo de confección del producto (software). Se evalúa mediante procedimientos que permitan una evaluación cuantitativa y cualitativa.

Evaluación final: se realiza al finalizar el producto (software) o proyecto, su objetivo principal es determinar el nivel de cumplimiento de todas las metas trazadas.

2.1.1 Flujo de trabajo, actividades, tareas y artefactos a generar

Flujo de trabajo (workflow en inglés), no es más que el estudio de todos aquellos aspectos operacionales que comprende una actividad de trabajo, se puede hacer mención de los mismos a través de las siguientes incógnitas:

- ❖ ¿Cómo se estructuran y ejecutan las tareas?
- ❖ ¿Cuál es su orden correlativo?
- ❖ ¿Cómo se sincronizan y fluye toda la información que soportan las mismas?
- ❖ ¿Cómo se realiza el monitoreo o seguimiento al cumplimiento de estas?

Se toma como soporte la ayuda de RUP y a continuación se describe el flujo de trabajo que se propone para el proceso de evaluación:

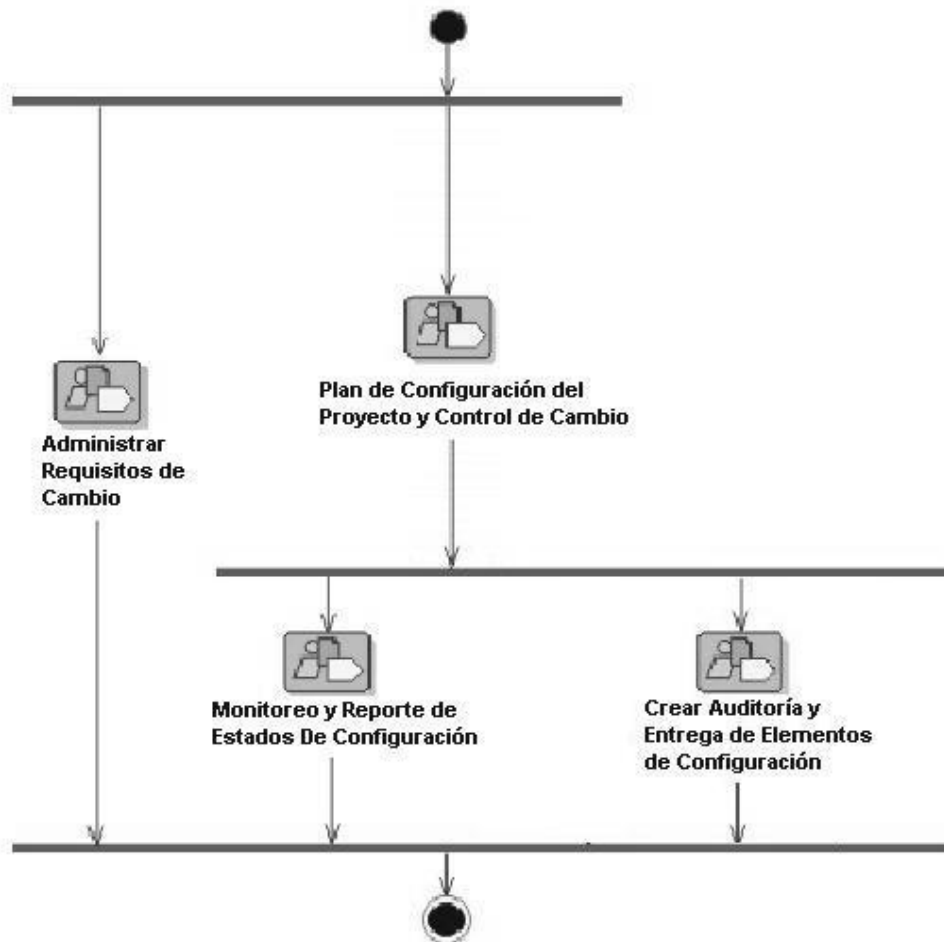


Figura 12 Flujo de Trabajo

Para poder realizar una evaluación satisfactoria de la Gestión de Configuración de los Proyectos Productivos de la Facultad 6, primeramente es necesario comprobar que dichos proyectos cuenten con un plan o procedimiento para la configuración del proyecto en general y el control de los cambios que se realizan en el mismo desde la etapa de inicio hasta que el producto es entregado al cliente. Establecer un entorno amigable para la administración de configuración del proyecto es el siguiente escalón que se debe verificar para el buen funcionamiento del proceso de evaluación que se propone, así como conocer el monitoreo los reportes que se generan del estado de la configuración, además de generar documentos de configuración de entrega y cambio. Comprobar que se administran las líneas bases y las liberaciones es un paso muy importante que se ha de tener en cuenta, no así restándole

importancia a la confirmación de la objetividad de administrar requisitos de cambio, ya que se debe realizar durante todo el progreso del proyecto. Con la graficación del anterior flujo de trabajo se pretende proporcionar la idea de cómo deben estar organizadas y enfocadas las diferentes actividades de evaluación del proceso de Gestión de Configuración del Software (GCS) para Proyectos Productivos.

Las tareas de un flujo de trabajo, no son más que la unidad primaria de modelación. Las mismas pueden ser cometidas por el sistema o por un usuario. Las tareas pueden figurar estructuras lógicas de control que precisen el ámbito y administren el flujo de la ejecución del flujo de trabajo, así como tener propiedades, métodos y eventos.

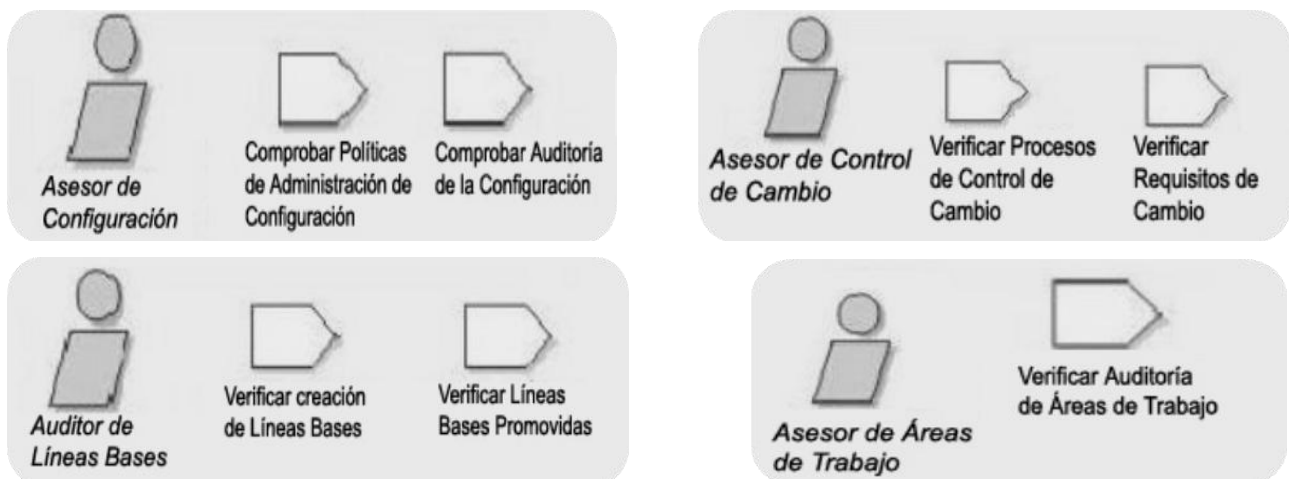


Figura 13 Roles y tareas del flujo de trabajo propuesto

Un artefacto designa cualquier tipo de información generada, cambiada o usada por los trabajadores del equipo de trabajo durante el desarrollo del sistema o producto (software). Los artefactos pueden estar presentes tanto en la Ingeniería como en la Gestión de Configuración de Software. Se puede decir formalmente que un artefacto es una pieza de información notoria, la cual es creada, modificada y utilizada por los trabajadores del proyecto al realizar las diferentes actividades dentro del mismo. Un artefacto puede ser un modelo, un elemento de un modelo o inclusive un documento.



Figura 14 Artefactos generados por los diferentes roles del flujo de trabajo

Las Figuras 5 y 6 muestran todas las tareas que comprende el flujo de trabajo presentado y los artefactos generados por los diferentes roles. De esta forma se plantea que para elaborar un buen **Plan de Configuración del Proyecto y Control de Cambio** con el que se pretende establecer un plan apropiado para tratar y controlar los cambios de los artefactos que se obtienen durante todo el proceso de desarrollo de software, es necesario evaluar la preexistencia de las siguientes tareas:

Comprobar Políticas de Administración de Configuración

No es más que comprobar que se lleva un control y almacenamiento de todos aquellos recursos que el proyecto pone a disposición de los trabajadores, es decir, se pretende evaluar todas aquellas prácticas de desarrollo del software. Los objetivos fundamentales de las políticas establecidas en un determinado proyecto productivo deben ser: lograr una buena comunicación entre los integrantes del

equipo de trabajo y conseguir que la fusión de sus trabajos minimicen los problemas que puedan surgir a lo largo del ciclo de vida del producto (software).

Verificar Procesos de Control de Cambio

Como su nombre lo indica, es confirmar el buen desarrollo de los procesos de control de cambio totalmente documentados que permiten asegurar que los cambios efectuados dentro de un proyecto productivo son realizados de manera consistente y al conocimiento de los stakeholders apropiados, es decir, que estos estén informados del costo e impacto que originan los cambios, así como del estado actual del producto.

Durante estas dos tareas a las que se hace referencia se genera un artefacto muy importante e indispensable para todo proyecto:

Informe de Administración de Configuración

El mismo permite conocer si existe un control y planificación reales del horario en que se realiza y debe ser llevada a cabo cada actividad, herramientas necesarias para la obtención del producto final y los imprescindibles recursos computacionales, incluyendo además el personal de trabajo, las responsabilidades asignadas a cada miembro del equipo de trabajo y recursos que se requieren para el buen funcionamiento del mismo, tiene como propósito definir si se cumple o no, con el control de los cambios durante el desarrollo de cualquier producto informático en un proyecto.

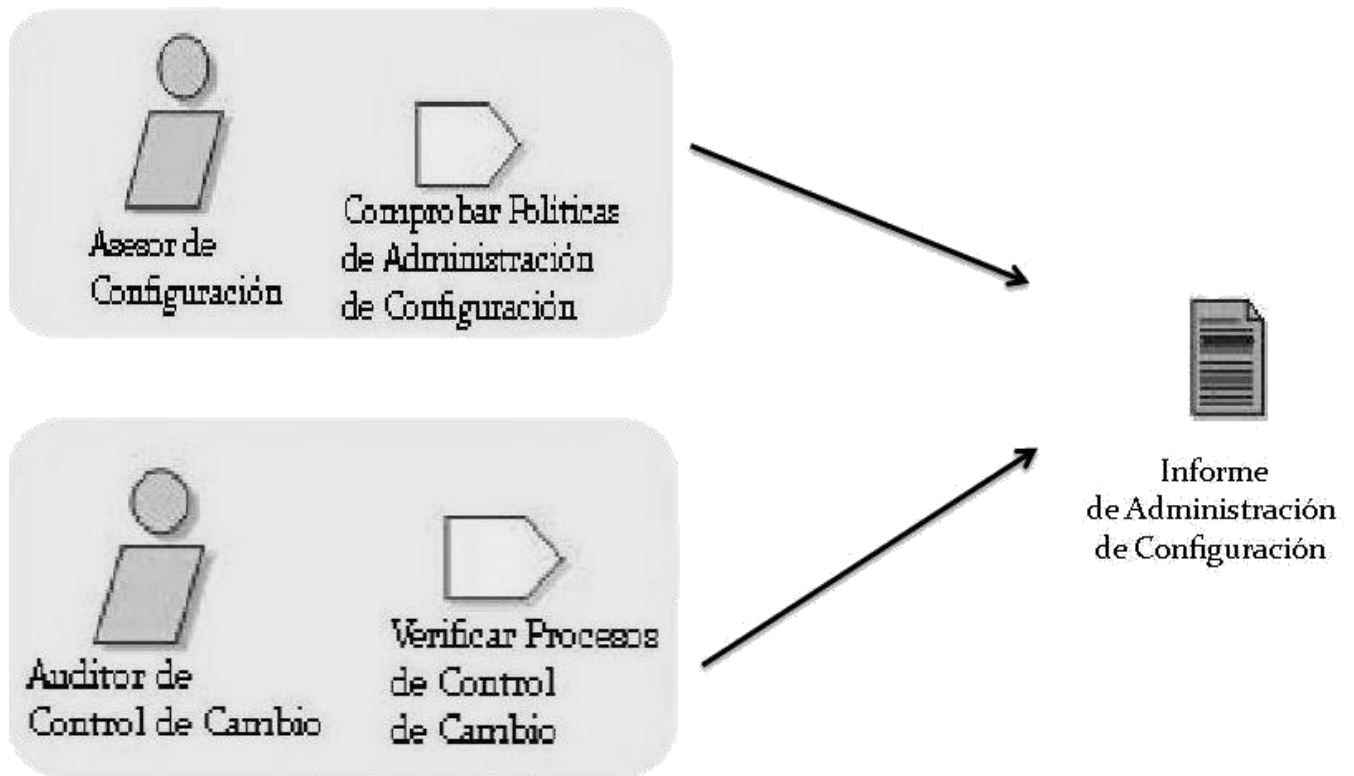


Figura 15 Actividad Plan de Configuración del Proyecto y Control de Cambio

Proporcionar visión sobre el cumplimiento del estado de la configuración de cambio: propósito de la actividad **Monitoreo y Reporte de Estados de Configuración**, es un trabajo substancial que se ha de evaluar en cualquier proceso de evaluación de la Gestión de Configuración de Software (GCS) para Proyectos Productivos. La misma incluye la tarea:

Comprobar Auditoría de la Configuración

La cual permite determinar si realmente una línea base definida en el proyecto contiene todos los artefactos requeridos que se generan durante el avance del mismo, y reúne los requisitos necesarios que se establecen durante su desarrollo.

Existen dos artefactos muy importantes que se generan a partir del cumplimiento de la tarea mencionada anteriormente, los mismos se abordan a continuación:

Informe de Auditoría de Configuración

Proporciona la intensidad con la que el proyecto ejecuta tareas como almacenar datos métricos, actualización del repositorio con la versión del proyecto más actual, medidas de datos originarios del producto durante todo el ciclo de vida del proyecto, identificar una línea base previamente definida así

como cualquier artefacto requerido perdido y probado parcialmente o carente de requisitos que lo sustenten, además reportar si la actuación del software desarrollado por el equipo de trabajo del proyecto cumple con los requisitos convenidos con el cliente, y si los artefactos requeridos se encuentran presentes físicamente.

Informe de Medidas del Proyecto tiene como propósito evaluar si se almacenan los datos métricos activos del proyecto, así como los recursos, procesos medidas del producto en su etapa inicial y niveles derivados.

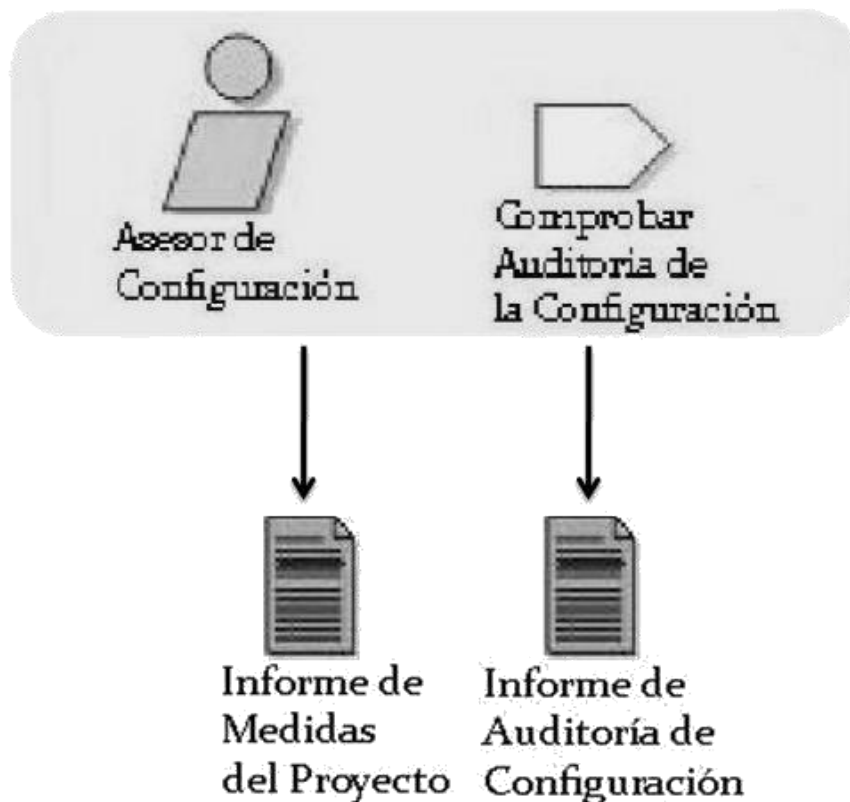


Figura 16 Actividad Monitoreo y Reporte de Estados de Configuración

Se hace necesaria la evaluación de la actividad **Crear Auditoría y Entrega de Elementos de Configuración**, porque así se garantiza conocer si sus tareas se realizan de manera íntegra y si los artefactos generados durante estas cumplen con las expectativas del proyecto.

Verificar Auditoría de Áreas de Trabajo

La misma tiene como meta primeramente, identificar si existe un área de trabajo de desarrollo, es decir, un área de desarrollo totalmente privada, si la misma es capaz de proporcionar un ambiente en el cual un miembro determinado del equipo de trabajo pueda hacer cambios a los artefactos sin que estos se tornen inmediatamente visibles a otros miembros del equipo, si esta área de trabajo es capaz de actualizarse para asegurar a los miembros del equipo que se trabaja en las más recientes versiones de los archivos y datos del proyecto; la idea es conocer si se puede tener completamente actualizados todos aquellos archivos desplegados en la vista de desarrollo (el área de trabajo) y línea base.

Verificar Creación de Líneas Bases

Esta es una tarea que posee el objetivo de comprobar que todos aquellos artefactos que se desarrollan durante el ciclo de vida de un proyecto se capturen y archiven, los cuales se utilizarán como puntos de control para el desarrollo del producto.

Verificar Líneas Bases Promovidas

Tarea que tiene como objetivo principal comprobar que las líneas bases estén etiquetadas para reflejar el nivel de madurez o fase, estabilidad y calidad del producto (software) lograda por el equipo de trabajo.

Varios son los artefactos que se generan a partir de esta actividad, primeramente se hace referencia a:

Notificación sobre Áreas de Trabajo

Medio que permite conocer si se ha cumplido o no con la comunicación previa sobre lo que será hecho, y cuando, al personal responsable del equipo de trabajo; permite saber si la planificación se convierte en la acción.

Notificación sobre Líneas Bases

Informa si se han creado satisfactoria y formalmente las líneas bases, si las mismas sirven como base para un desarrollo posterior del proyecto y si pueden o no cambiarse solamente a través de procedimientos formales de control de cambios.

Reporte de Elementos de Configuración

Es el artefacto que permite conocer si en un determinado proyecto productivo existe o no un plan de proyecto, especificación de requisitos, manual de usuario preliminar y final, descripciones del diseño de los módulos, descripciones del diseño de interfaces, descripciones de los objetos (si se utilizan

técnicas de P.O.O.), descripción de la base de datos y estándares y procedimientos de ingeniería del software.

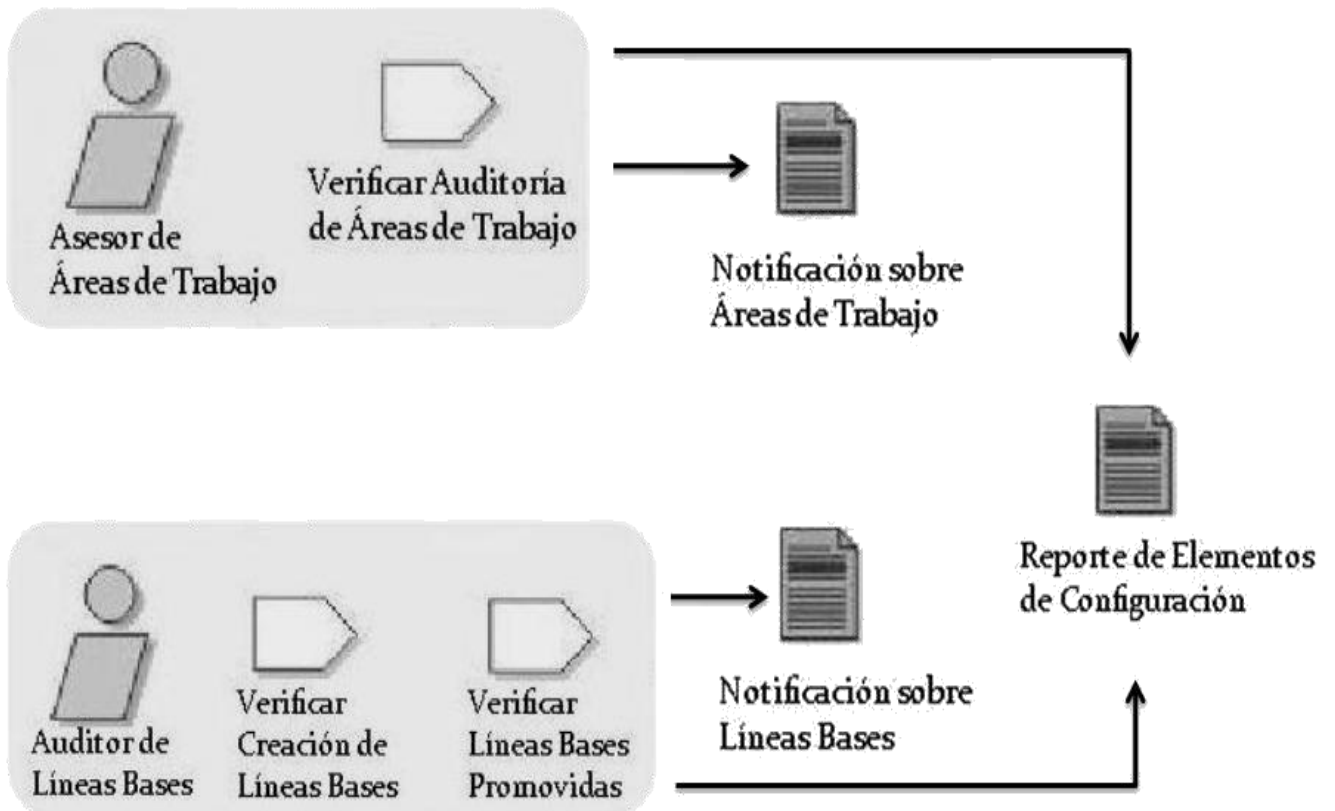


Figura 17 Actividad Crear Auditoría y Entrega de Elementos de Configuración

Cuando se evalúa la actividad **Administrar Requisitos de Cambio**, cuyo propósito es registrar si los cambios aprobados dentro de un proyecto se realizan de forma estable, en primer lugar se evalúa la tarea:

Verificar Requisitos de Cambio

La cual comprueba si se anota un cambio solicitado, si se incluyen solicitudes para nuevas características, mejoras, correcciones, requisitos cambiados.

Durante esta actividad se genera un solo artefacto:

Informe de Requisitos de Cambio

El cual indica si en el proyecto existe o no una pieza de información notoria que pueda usarse para documentar y monitorear todos los tipos de demandas para cambios al sistema, incluso demandas de mejora y defectos.



Figura 18 Actividad Administrar Requisitos de Cambio

2.1.2 Entradas, Salidas y procedimientos (o técnicas)

Tareas	Artefactos de Entrada	Artefactos de Salida
Verificar Políticas de Administración de Configuración	Informe de Administración de Configuración	Informe de Administración de Configuración
Comprobar Auditoría de la Configuración	Informe de Auditoría de Configuración	Informe de Medidas del Proyecto
Verificar Procesos de Control de Cambio	Informe de Administración de Configuración	Informe de Administración de Configuración
Verificar Requisitos de Cambio	Informe de Requisitos de Cambio	Informe de Requisitos de Cambio
Verificar Creación de Líneas Bases	Notificación sobre Líneas Bases	Reporte de Elementos de Configuración
Verificar Líneas Bases Promovidas	Notificación sobre Líneas Bases	Reporte de Elementos de Configuración
Verificar Auditoría de Áreas de Trabajo	Notificación sobre Áreas de Trabajo	Reporte de Elementos de Configuración

Figura 19 Tareas, artefactos de Entrada y de Salida

Varios son los procedimientos que se emplean para desarrollar cada una de las tareas del flujo de trabajo que se propone, a continuación se hace referencia a los mismos:

Tareas	Procedimientos
Comprobar Políticas de Administración de Configuración	Evaluar control de recursos del proyecto y Gestión de Configuración
	Evaluar almacenamiento de recursos del proyecto
	Evaluar prácticas de desarrollo del software
Verificar Procesos de Control de Cambio	Evaluar desarrollo de procesos de control de cambio documentados
Comprobar Auditoría de la Configuración	Evaluar consistencia de líneas bases definidas en el proyecto
Verificar Auditoría de Áreas de Trabajo	Evaluar existencia y funcionalidad del área de trabajo de desarrollo
	Verificar actualización del área de trabajo de desarrollo
	Verificar actualización de líneas bases
Verificar Creación de Líneas Bases	Evaluar captura y almacenamiento de artefactos
Verificar Líneas Bases Promovidas	Comprobar líneas bases etiquetadas
Verificar Requisitos de Cambio	Evaluar anotaciones de cambio
	Evaluar solicitudes de características
	Evaluar solicitudes de requisitos cambiados

Figura 20 Tareas y procedimientos o técnicas

2.2 Personal involucrado en el proceso de evaluación

Se propone un sistema de evaluación que permite evaluar el proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6, se utiliza para ello una serie de roles, responsabilidades, actividades, tareas y artefactos para la selección de indicadores y estándares a utilizar en la evaluación. El flujo de trabajo propuesto, de manera general puede ser considerado una guía para la evaluación que se llevará a cabo durante el proceso, refiriéndose principalmente a los resultados de todo el proceso de evaluación de la GCS de los Proyectos Productivos, además, comprende un conjunto importante de personas que se encarga de llevar a cabo y ejecutar cada actividad y tarea para garantizar un buen proceso de evaluación, en fin, el personal involucrado en el mismo constituye la unidad básica para efectuar una evaluación satisfactoria de esta disciplina de la Ingeniería de Software.

2.2.1 Roles y responsabilidades

Roles	Responsabilidades
Asesor de Configuración	Informe de Administración de Configuración
	Informe de Medidas del Proyecto
	Informe de Auditoría de Configuración
Auditor de Control de Cambio	Informe de Requisitos de Cambio
Auditor de Líneas Bases	Reporte de Elementos de Configuración Notificación sobre Líneas Bases
Asesor de Áreas de Trabajo	Reporte de Elementos de Configuración Notificación sobre Áreas de Trabajo

Figura 21 Roles y responsabilidades

Rol Asesor de Configuración

Es la persona encargada de evaluar todas aquellas prácticas de desarrollo del software y líneas bases definidas en el proyecto. Posee tres responsabilidades importantes durante las actividades Plan de Configuración y Control de Cambio y Monitoreo y Reporte de Estados de Configuración, las mismas son: Informe de Administración de Configuración, Informe de Auditoría de Configuración e Informe de Medidas del Proyecto.

Rol Auditor de Control de Cambio

Es la persona destinada a confirmar el buen desarrollo de los procesos de control de cambio. Durante el desarrollo de las actividades Plan de Configuración y Control de Cambio y Administrar Requisitos de Cambio es responsable del Informe de Requisitos de Cambio, y durante la actividad Administrar Requisitos de Control de Cambio es responsable del artefacto Informe de Requisitos de Cambio.

Rol Auditor de Líneas Bases

Es la persona que comprueba la captura y almacenamiento de todos los artefactos que se desarrollan durante el ciclo de vida de un proyecto y que las líneas bases del mismo reflejen el nivel de madurez del software. Reporte de Elementos de Configuración y Notificación sobre Líneas Bases son las responsabilidades de este individuo durante la actividad Crear Auditoría y Entrega de Elementos de Configuración.

Rol Asesor de Áreas de Trabajo

Es la persona que identifica si existe un área de trabajo de desarrollo. Posee dos responsabilidades, Reporte de Elementos de Configuración y Notificación sobre Áreas de Trabajo durante la actividad, ambas durante la actividad Crear Auditoría y Entrega de Elementos de Configuración.

2.3 Plantillas

Plantilla es un instrumento que sirve de guía, soporte o un medio que permite crear un diseño o esquema predeterminado; proporciona una estructura y contenido totalmente separados. Varias son las plantillas que se utilizan para llevar a cabo todo el proceso de evaluación que se propone, las mismas surgen como consecuencia de las diferentes técnicas o procedimientos que se emplean durante el desarrollo de las disímiles tareas que comprenden el flujo de trabajo propuesto, las mismas quedan de esta forma reflejadas en cuatro de las responsabilidades desempeñadas por los distintos

roles involucrados en la evaluación que se propone para el proceso de Gestión de Configuración para Proyectos Productivos de la facultad.

2.4 Descripción de las plantillas

A continuación se hará una descripción detallada de cada uno de los parámetros que comprenden las cuatro plantillas que se utilizarán como apoyo para llevar a cabo toda la evaluación del proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6. Los parámetros serán medibles y se evaluarán por la persona requerida (evaluador) en un rango de cero a cinco puntos, en dependencia del nivel de cumplimiento o con que se realicen cada una de las actividades a evaluar en el proyecto.

2.4.1 Plantilla Informe de Administración de Configuración

Es la encargada de evaluar el control de recursos del proyecto y Gestión de Configuración, es decir, evaluar el proceso de vigilar aquellas actividades que aseguren un cumplimiento del registro de recursos necesarios para cada una de las tareas o acciones por realizar. Existen tres tipos de recursos que se requieren para la realización de un proyecto en la facultad: tiempo, personas y equipos. Para realizar una evaluación satisfactoria de los recursos que un proyecto productivo pone a disposición de los trabajadores, es necesario determinar cuales necesitas para llevar a cabo cada acción, es decir, tiempo del que dispones y cuántas personas, instrumentos requieres. En primer lugar se debe evaluar si el tiempo con el que se cuenta para desarrollar el producto (software) es suficiente para la confección del mismo, para esto se debe tener en cuenta el ritmo de avance en la elaboración del producto, el número de personas que componen el equipo de trabajo y la experiencia y capacidad de cada una para realizar las tareas asignadas, además de las instalaciones como laboratorios, computadoras, tecnologías y sistemas de seguridad.

También se registran todas las revisiones o aprobaciones de los diferentes estados por los que pasa el producto (software) durante todo el ciclo de desarrollo del mismo. *Se define el entorno tecnológico de soporte a la gestión de la configuración del sistema de información y se determinan los componentes hardware y software que van a permitir la mecanización de los procesos y controles que establece el plan. Es importante mencionar que el entorno tecnológico en el que se apoyará el sistema de gestión de la configuración puede ser diferente de aquel en el que se desarrollará el sistema de información [24].* Al finalizar cada proceso se debe identificar y registrar el producto correspondiente.

La realización de un cambio, bien por mantenimiento correctivo o evolutivo, provoca la aparición de una nueva versión de cada sistema de información afectado por el cambio, por lo que debe registrarse en el sistema de gestión de la configuración con la versión y estado correspondiente según establezca el plan de gestión de la configuración [24] del proyecto.

Con esta plantilla además, se evalúa el almacenamiento de recursos del proyecto al evaluar o comprobar la existencia de un cronograma o registro del tiempo que se emplea en cada tarea realizada en el proyecto, ya que de esta forma se tiene una idea clara sobre si se puede cumplir o no la meta trazada, y se facilita una planificación más precisa del tiempo que se utilizará por parte de los miembros del equipo de trabajo del proyecto productivo en la realización de las diferentes actividades. También se comprueba la existencia de un plan para la superación de cada integrante del proyecto, es decir, métodos como: propuestas de cursos optativos relacionados con el lenguaje de programación y con el trabajo y manejo de las herramientas que se utilizan durante el desarrollo y avance del proyecto, estas son algunas de las técnicas de las que se debe valer el jefe de proyecto para promover y elevar el intelecto y experiencia en cada uno de los miembros del proyecto. Y por último se hace necesario comprobar si existe o no un inventario o compilación de datos que permitan conocer con exactitud la cantidad física de cada instrumento que el proyecto pone a disposición de los integrantes del proyecto y el estado actual de cada equipo, así como un registro que sea capaz de brindar toda la información necesaria sobre la tecnología software que se explota en dicho proyecto productivo.

Evaluar prácticas de desarrollo del software es la última técnica que se evalúa en la plantilla Informe de Administración de Configuración, con la misma se pretende comprobar si en el proyecto existe una gestión de trabajo, o sea, una lista con todas las funcionalidades de la aplicación o producto (software) que se desarrollará y un cronograma que muestre el nivel de prioridad que presenta cada tarea a ejecutar durante el ciclo de vida del proyecto, además se evalúa la periodicidad con que se realizan las reuniones con los miembros del equipo de trabajo del proyecto para analizar aquellas tareas ya cumplidas y las que aun están pendientes, así como los posibles impedimentos que puedan surgir para la realización de las mismas, se comprueba la existencia de un programa de seguimiento y control de errores, de esta forma se evita la ausencia de un registro de errores, una mala descripción, gestión, asignación y priorización de los mismos, una deficiente comunicación por parte de los miembros del proyecto se garantiza la objetividad de un listado de errores con sus respectivos estados de resolución.

Plantilla Informe de Administración de Configuración	
Parámetros	Puntuación
1- Se determina el tiempo disponible para realizar cada tarea específica	0-5
2- Se determina número de personas necesarias para realizar cada tarea específica	0-5
3- Se determina instrumentos necesarios para realizar cada tarea específica	0-5
4- Se registran todas las revisiones o aprobaciones de los diferentes estados por los que pasa el producto (software)	0-5
5- Se define el entorno tecnológico de soporte a la gestión de la configuración del sistema de información	0-5
6- Se determinan los componentes hardware y software que van a permitir la mecanización de los procesos y controles que establece el plan	0-5
7- Al finalizar cada proceso se identifica y registra el producto correspondiente	0-5
8- Cuando ocurre un cambio el mismo se registra en el sistema de Gestión de la Configuración con la versión nueva	0-5
9- Existe registro del tiempo que se emplea en cada tarea realizada	0-5
10- Existen propuestas de cursos optativos relacionados con el lenguaje de programación	0-5
11- Existen propuestas de cursos optativos relacionados con el trabajo y manejo de las herramientas que se utilizan durante el desarrollo y avance del proyecto	0-5
12- Existe inventario de los instrumentos que el proyecto pone a disposición de los integrantes	0-5
13- Existe lista de todas las funcionalidades de la aplicación o producto (software)	0-5
14- Existe cronograma que muestra el nivel de prioridad que presenta cada tarea a ejecutar durante el ciclo de vida del proyecto	0-5
15- Se realizan con periodicidad reuniones con los miembros del equipo de trabajo del proyecto	0-5
16- Existe un programa de seguimiento y control de errores	0-5

17- Criterio del evaluador	0-5
Promedio de cumplimiento de Administración de Configuración	0.00-5.00

Figura 22 Plantilla Informe de Administración de Configuración

Con el parámetro número cinco de la Plantilla Informe de Administración de Configuración mostrada anteriormente en la Figura 9 se quiere hacer referencia a la implantación de un entorno de alta competitividad y cambio, que permita a la gestión de la configuración del sistema de información realizar de forma flexible adaptaciones a nuevas exigencias en el mundo del software.

El parámetro seis aborda lo referente a componentes hardware y software que van a permitir llevar a cabo el desarrollo del producto (software), el hardware no es más que la parte física de la tecnología de una computadora, es decir, periféricos como: disco duro, CD-ROOM, disquetera (floppy), cables, etcétera; y el software se refiere a la parte lógica o soporte lógico de un ordenador, por ejemplo: aplicaciones informáticas y sistemas operativos.

El parámetro siete hace alusión a los productos que se identifican y registran al finalizar cada proceso. Como procesos se pueden mencionar los siguientes: *Análisis del Sistema de Información (ASI)*, *Diseño del Sistema de Información (DSI)*, *Construcción del Sistema de Información (CSI)* e *Implantación y Aceptación del Sistema (IAS)* [24], como productos: *Especificación de Requisitos Software*, *Diseño Detallado del Sistema*, *Sistema de Información* y *Sistema de Información Implantado* [24].

Y por último, el parámetro ocho se refiere al control de versiones, es decir, al registro de versiones nuevas por parte del proyecto cada vez que se efectúa un cambio, tenga o no este un impacto grande en el mismo.

2.4.2 Plantilla Informe de Medidas del Proyecto

Es otra de las plantillas que se utilizan para el proceso de evaluación que se propone, mediante la misma se evalúa la consistencia de las líneas bases definidas en el proyecto, se comprueba que la revisión y aprobación de las mismas se hayan realizado formalmente, que sirvan de soporte para un posterior desarrollo, y que solo pueden ser modificadas a través de procedimientos formales de control de cambios. A continuación se explican los tipos de líneas bases más comunes en ciclos de vida de Proyectos Productivos y que serán evaluadas durante el proceso de evaluación que se propone:

- ❖ *Línea Base de Sistema: Se establece al finalizar la fase de Especificación de Requisitos del sistema global, y comprende todos aquellos documentos en los que se define el problema a resolver a nivel sistema, el plan de tiempos del proyecto y estimaciones, los modelos de la situación y del dominio, el estudio de viabilidad y las especificaciones del sistema.*
- ❖ *Línea Base Funcional: Se fija al finalizar la fase de Análisis de Requisitos y contendrá los documentos donde se ha establecido la Especificación formal de Requisitos de Software (Descripción de datos de entrada y salida, funciones que realizará el sistema, rendimiento requerido del sistema, interfaces, restricciones generales y flujo de información), el Plan de Pruebas y la conformidad del cliente sobre la especificación formal de requisitos.*
- ❖ *Línea Base de Diseño: Se establece al finalizar la fase de Diseño Detallado y comprende la documentación relacionada con las descripciones de diseño del software, de la arquitectura, de los flujos de información, de las bases de datos (sí es de aplicación), de las interfaces, de los algoritmos, las referencias cruzadas con los requisitos, las previsiones para las pruebas y el manual del usuario preliminar.*
- ❖ *Línea Base de Producto: Se establece al finalizar las fases de Codificación y Pruebas y contiene no sólo la documentación relacionada con estas fases (listados de los fuentes, librerías y bases de datos, documentación del sistema, datos para casos de prueba, documentación para el usuario, documentación relacionada con las herramientas de hardware y software (características de la máquinas empleadas para el desarrollo, compiladores, depuradores) utilizados para generar el producto, plan de pruebas, especificación de las pruebas, informe resumen de pruebas y plan de integración) sino también archivos sobre soporte magnético (fuentes, objetos, ejecutables, librerías, bases de datos asociadas).*
- ❖ *Línea Base Operativa: Se establece al finalizar la fase de Instalación o Implantación, comprende la documentación relacionada con las tareas de operación y mantenimiento, plan de formación, recomendaciones de mantenimiento y Plan de Retiro [1].*

Existen tres tipos de ciclos de vida: Ciclo de Vida Tradicional, Ciclo de Vida Prototipado Evolutivo y Ciclo de Vida en Espiral, pero la evaluación del proceso de Gestión de configuración de los Proyectos Productivos de la Facultad 6 que se propone se realizara rigiéndose por el primero mencionado y explicado anteriormente.

Plantilla Informe de Medidas del Proyecto	
Parámetros	Puntuación
1- Al finalizar la fase de Especificación de Requisitos del sistema global se establece y se revisa formalmente una Línea Base de Sistema	0-5
2- Al finalizar la fase de Especificación de Requisitos se establece y se revisa formalmente una Línea Base Funcional	0-5
3- Al finalizar la fase de Diseño Detallado se establece y se revisa formalmente una Línea Base de Diseño	0-5
4- Al finalizar las fases de Codificación y Pruebas se establece y se revisa formalmente una Línea Base de Producto	0-5
5- Al finalizar la fase de Instalación o Implantación se establece y se revisa formalmente una Línea Base Operativa	0-5
6- Las Líneas Bases definidas en el proyecto sirven de soporte para un posterior desarrollo	0-5
7- Las Líneas Bases definidas en el proyecto solo pueden ser modificadas a través de procedimientos formales de control de cambios	0-5
8- Criterio del evaluador	0-5
Promedio de cumplimiento de Medidas del Proyecto	0.00-5.00

Figura 23 Plantilla Informe de Medidas del Proyecto

2.4.3 Plantilla Informe de Requisitos de Cambio

A través de la plantilla se evalúa el desarrollo de los procesos de control de cambio documentados, es aquí donde se comprueba si el proyecto cuenta o no con una guía que le permita elaborar informes precisos sobre la evaluación del rendimiento de dichos procesos, para ello es imprescindible cubrir aspectos como:

- ❖ Número de cambios realizados en el proyecto, los mismos deben estar clasificados de acuerdo al impacto y prioridad que pueden originar.
- ❖ Porcentaje de cambios exitosos enumerados por instancias.
- ❖ Revisiones realizadas después de la implementación.

- ❖ Cantidad de incidencias relacionadas con los cambios realizados.
- ❖ Número de reuniones efectuadas en el proyecto donde se refleje toda la información asociada al número de cambios aprobados por reunión, asistencia, duración, etcétera.

Además, también se evalúa las anotaciones de cambio, que no es más que identificar y anotar aquellos elementos del proyecto que deben estar bajo configuración, anotar restricciones de cambios a dichos elementos, así como el rastreo de los cambios que se necesiten realizar a los mismos. Muchos son los beneficios que se obtienen de la realización de un buen control y anotación de cambio, a continuación se mencionan algunos de ellos:

- ❖ Se mantiene la consistencia de los productos que se desarrollan.
- ❖ Se proporciona un entorno totalmente controlado para trabajo.
- ❖ Se restringen y controlan los cambios que realizados a archivos y datos del proyecto.
- ❖ Se proporciona mecanismos que permiten conocer el motivo, la hora y la persona que realizó un cambio.

Evaluar solicitudes de características y de requisitos cambiados, aspectos que tratan las solicitudes para nuevas particularidades referentes al diseño y funcionalidad del producto (software), así como las solicitudes de aquellos requerimientos que en algún momento durante el ciclo de vida del proyecto fueron modificados por petición de los clientes o por conveniencia de los miembros del equipo de trabajo. Ambos parámetros están enfocados a un objetivo común: solicitud, que no es más que el documento en el que se solicita formalmente algo.

Una vez que se compruebe la existencia de la guía, así como el cumplimiento de las anotaciones y solicitudes pertinentes, se puede llevar a cabo la evaluación para determinar si el desarrollo de los procesos y resultados han sido satisfactorios.

Plantilla Informe de Requisitos de Cambio	
Parámetros	Puntuación
1- Los cambios realizados en el proyecto se clasifican de acuerdo al impacto y prioridad que pueden originar	0-5
2- El porcentaje de cambios exitosos se enumeran por instancias	0-5
3- Se realizan revisiones después de la implementación	0-5
4- Se controla la cantidad de incidencias relacionadas con los cambios realizados	0-5
5- Se controla el número de reuniones efectuadas en el proyecto donde se refleje toda la información asociada al número de cambios aprobados por reunión, asistencia, duración, etcétera	0-5
6- Se identifican y anotan los elementos del proyecto que deben estar bajo configuración	0-5
7- Se anotan las restricciones de cambios a dichos elementos	0-5
8- Se rastrean los cambios que se necesitan realizar a dichos elementos	0-5
9- Se registran las solicitudes para nuevas particularidades referentes al diseño y funcionalidad del producto (software)	0-5
10- Se registran las solicitudes de los requerimientos que en algún momento fueron modificados por petición de los clientes o por conveniencia de los miembros del equipo de trabajo	0-5
11- Criterio del evaluador	0-5
Promedio de cumplimiento de Requisitos de Cambio	0.00-5.00

Figura 24 Plantilla Informe de Requisitos de Cambio

El parámetro número seis de la Plantilla Informe de Requisitos de Cambio se refiere a aquellos elementos que deben estar bajo configuración, a continuación se explica qué son los llamados elementos de configuración y se muestra un listado con algunos ejemplos.

Elementos de configuración de software

Se conoce como elemento de configuración de software a la unidad física y/o lógica que lo mismo pudo haber sido desarrollada o adquirida, su administración se hace totalmente necesaria e importante.

Elementos de Configuración de Software en un proyecto productivo:

- ❖ *El plan de proyecto*
- ❖ *El plan de Gestión de Configuración*
- ❖ *El documento de definición de requerimientos*
- ❖ *Estándares de análisis, diseño, codificación, pruebas, y auditoria*
- ❖ *Documentos de análisis del sistema*
- ❖ *Documentos de diseño del sistema*
- ❖ *Prototipos*
- ❖ *Documentos de diseño de alto nivel*
- ❖ *Documentos de diseño de bajo nivel*
- ❖ *Especificaciones de prueba del sistema*
- ❖ *El plan de pruebas del sistema*
- ❖ *El Código fuente del programa*
- ❖ *Código objeto y ejecutable*
- ❖ *Especificaciones de pruebas de unidad*
- ❖ *Planes de pruebas de unidad*
- ❖ *Documentos de diseño de base de datos*
- ❖ *Datos de prueba*
- ❖ *Datos del proyecto*
- ❖ *Manuales de usuario [25]*

2.4.4 Plantilla Reporte de Elementos de Configuración

Mediante esta plantilla se evalúa la eficiencia de la captura y almacenamiento de artefactos y registros del proyecto; el sistema de administración de configuración es el encargado de realizar este tipo de tarea, ya que es capaz de llevar una correcta gestión, distribución, almacenamiento y archivado de la documentación y evitar a la vez registros duplicados, confusión en versiones de ficheros, pérdida de información y datos cruciales del proyecto. Durante la evaluación se comprueba si los artefactos y

registros como el Plan de Proyecto, Contratos, Registros Históricos, etcétera son capturados cuando se generan y almacenados en el repositorio del proyecto para ser utilizados posteriormente como puntos de control para el desarrollo del producto. Mediante esta plantilla también se comprueba que las líneas bases estén etiquetadas para luego evaluar en que fase se encuentra el producto (software); las fases son las siguientes:

❖ Versión Inicial

Las características del producto (software) están incompletas. Aun no se tienen definidas todas las funcionalidades del producto.

❖ Versión Alfa

Es la primera versión del producto que reúne todas las funcionalidades.

❖ Versión Beta

Es la primera versión que reúne todas las características del producto, a partir de entonces solo se trabajará en mejoras y correcciones del mismo.

❖ Versión Final

Es la versión del producto final, solamente posee correcciones de último momento. Esta es una versión muy estable y limitada de errores, ostenta de una calidad adecuada para una futura distribución y explotación por parte de los usuarios.

La calidad del producto (software) es otro de los parámetros que se tiene en cuenta durante el proceso de evaluación que se lleva a cabo mediante esta plantilla, comprueba que el proyecto realiza pruebas de calidad internas y externas al producto. Dichas pruebas comprenden los siguientes aspectos:

❖ Funcionalidad

Capacidad del producto (software) para proporcionar funciones, resultados correctos y fieles, para interactuar con uno o más sistemas y para proteger información y datos.

❖ Fiabilidad

Capacidad del producto (software) para evitar fallos, para mantener un nivel especificado de prestaciones y salvar datos en caso de fallos.

❖ Usabilidad

Capacidad del producto (software) que permite al usuario entender cómo puede ser usado, operado y controlado el mismo, además se determina la capacidad del producto para ser atractivo al usuario.

❖ Eficiencia

Capacidad del producto (software) para procesar datos, así como la velocidad con la que realiza esta función, además se determina si el nivel de recursos que utiliza es alto, medio o bajo.

❖ Mantenibilidad

Capacidad del producto (software) para asimilar cambios en el código fuente y para evitar efectos inesperados debidos a modificaciones del mismo.

❖ Portabilidad

Capacidad del producto (software) para ser adaptado e instalado en diferentes entornos y para compartir recursos con otro software en un entorno común.

Existe otro tipo de prueba de calidad, y es la denominada prueba de calidad en uso, la cual comprende parámetros como:

❖ Efectividad

Capacidad del producto (software) para permitir a los usuarios alcanzar objetivos específicos con precisión.

❖ Productividad

Capacidad del producto (software) para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada.

❖ Seguridad física

Capacidad del producto (software) para alcanzar niveles aceptables del riesgo de hacer daño a clientes, al proyecto, al software o al medio ambiente.

❖ Satisfacción

Capacidad del producto (software) para satisfacer las necesidades de los usuarios.

Evaluar la existencia y funcionalidad del área de trabajo de desarrollo, es otro de los tantos parámetros que recluye esta plantilla, mediante la misma se comprueba si en el proyecto existe un área de desarrollo capaz de llevar a cabo la fabricación de cualquier sistema en función de las exigencias definidas. El área de trabajo a la cual se hace referencia juega un papel muy importante dentro de todo proyecto productivo, ya que es el área capacitada para facilitar el desarrollo de proyectos sencillos y complejos, es aquí donde se utiliza y emplea un número importante de herramientas y tecnologías sobre diversas plataformas para poder crear un entorno privado en el cual un determinado miembro del equipo de trabajo pueda realizar todos aquellos cambios necesarios en los artefactos generados sin

que estos se tornen automáticamente perceptibles a otros miembros del equipo, de esta forma se hace indispensable la actualización del área de trabajo de desarrollo, es decir, la actualización de versiones de herramientas, tecnologías y plataforma que se emplean para desarrollar un producto (software). Trabajar con herramientas de última generación y tecnologías de punta, permiten realizar actualizaciones planificadas y constantes de todos aquellos artefactos y archivos que sufren cambios durante el ciclo de vida de desarrollo de un producto (software), de esta forma se puede asegurar a los miembros del equipo de trabajo que se trabaja con las versiones más recientes de los archivos y datos del proyecto.

Verificar actualización de líneas bases es el último parámetro medible en esta plantilla, el mismo se centra en evaluar todos los puntos de referencia principales dentro de un proyecto productivo, los cuales pueden ser categorizados de la siguiente forma:

- ❖ Fecha de inicio del proyecto

Es la fecha en la que está programado el comienzo de un proyecto productivo. Esta fecha se basa en la planificación previa de todas las posibles tareas a realizar durante el ciclo de desarrollo del proyecto.

- ❖ Fecha de finalización del proyecto

Es la fecha en la que está programado que se culmine el proyecto productivo.

- ❖ Duración de cada actividad del proyecto

Es el período total de tiempo de trabajo activo que se emplea para completar una tarea asignada.

- ❖ Trabajo realizado en el proyecto

Es la mano de obra total necesaria para completar una tarea asignada.

- ❖ Registro de cada fase del proyecto

Es el registro que permite comparar los valores previstos en cada mes con los datos reales alcanzados por el proyecto.

Como la línea base proporciona todos los puntos de referencia con los se compara el avance real del proyecto, constituye un factor de suma importancia dentro del proceso de evaluación que se propone.

Plantilla Reporte de Elementos de Configuración	
Parámetros	Puntuación
1- Se capturan y almacenan con eficiencia el Plan de Proyecto, Contratos, Registros Históricos, etcétera	0-5
2- Las Líneas Bases son etiquetadas para mostrar la fase en que se encuentra el producto (software)	0-5
3- Se realizan pruebas de calidad para determinar la funcionalidad del producto (software)	0-5
4- Se realizan pruebas de calidad para determinar la fiabilidad del producto (software)	0-5
5- Se realizan pruebas de calidad para determinar la usabilidad del producto (software)	0-5
6- Se realizan pruebas de calidad para determinar la eficiencia del producto (software)	0-5
7- Se realizan pruebas de calidad para determinar la mantenibilidad del producto (software)	0-5
8- Se realizan pruebas de calidad para determinar la portabilidad del producto (software)	0-5
9- Se realizan pruebas de calidad para determinar la efectividad del producto (software)	0-5
10- Se realizan pruebas de calidad para determinar la productividad del producto (software)	0-5
11- Se realizan pruebas de calidad para determinar la seguridad física del producto (software)	0-5
12- Se realizan pruebas de calidad para determinar la satisfacción del producto (software)	0-5
13- Existe un área de trabajo de desarrollo	0-5
14- El área de trabajo de desarrollo es capaz de crear un entorno privado	0-5
15- Se actualiza la Línea Base o punto de referencia principal Fecha de inicio del proyecto	0-5
16- Se actualiza la Línea Base o punto de referencia principal Fecha de finalización del proyecto	0-5
17- Se actualiza la Línea Base o punto de referencia principal Duración de cada actividad del proyecto	0-5
18- Se actualiza la Línea Base o punto de referencia principal Trabajo realizado en el proyecto	0-5
19- Se actualiza la Línea Base o punto de referencia principal Registro de cada fase del proyecto	0-5

20- Criterio del evaluador	0-5
Promedio de cumplimiento de Elementos de Configuración	0.00-5.00

Figura 25 Plantilla Reporte de Elementos de Configuración

Dentro de las plantillas mencionadas anteriormente, existe un parámetro muy importante, el mismo es: Criterio del evaluador, que no es más que la opinión cuantitativa del resultado de la evaluación llevada a cabo por la persona destinada. Se puntualiza que la persona que realiza la evaluación debe estar totalmente calificada como experto en Gestión de Configuración de Software y en todas las demás ramas de Ingeniería de Software que hacen referencia a esta disciplina, ya que sus conocimientos y experiencia influyen directa y determinadamente en el resultado de cualquier evaluación realizada en un proyecto productivo.

La Figura 13 representa una leyenda de puntuación cuantitativa que comprende cada una de las observaciones que se debe tener en cuenta a la hora de emitir por parte del Evaluador un criterio cuantitativo referente a cada parámetro comprendido en las plantillas anteriormente descritas.

Tabla Leyenda de Puntuación Cuantitativa	
Puntuación	Comentario
0	No se conoce la actividad de Gestión de Configuración de Software que debe existir o se debe realizar
1	Se conoce la actividad de Gestión de Configuración de Software que debe existir o se debe realizar, pero no existe o no se lleva a cabo en el proyecto. No existe interés por aplicar esta disciplina de Ingeniería de Software
2	Se conoce la actividad de Gestión de Configuración de Software que debe existir o se debe realizar, pero no existe o no se lleva a cabo en el proyecto. Existe gran interés y esfuerzo por aplicar esta disciplina de Ingeniería de Software
3	La actividad de Gestión de Configuración de Software que debe existir o se debe realizar existe o se realiza con muy bajo nivel de madurez que el se requiere. El control se realiza mediante cualquier miembro del proyecto
4	La actividad de Gestión de Configuración de Software que debe existir o se debe realizar existe o se realiza pero sin el nivel de madurez requerido. No se controla mediante expertos pero si mediante personas con alguna experiencia en la materia
5	La actividad de Gestión de Configuración de Software que debe existir o se debe realizar existe o se realiza con el nivel de madurez requerido. Se controla mediante expertos

Figura 26 Tabla Leyenda de Puntuación Cuantitativa

La Tabla Leyenda de Puntuación Promedio representa las conclusiones a las que se llega una vez que el proyecto haya sido evaluado por el Evaluador y se conozca con exactitud la puntuación promedio que el mismo haya alcanzado.

Tabla Leyenda de Puntuación Promedio	
Puntuación	Conclusiones
0.00-1.00	Los integrantes del proyecto no tienen ningún tipo conocimiento de las actividades de Gestión de Configuración de Software que se deben llevar a cabo en todo proyecto productivo
1.01-2.00	Los integrantes del proyecto tienen escasos conocimientos acerca de las actividades de Gestión de Configuración de Software que se deben llevar a cabo en todo proyecto productivo, no realizan ninguna actividad referente a esta disciplina de Ingeniería de Software
2.01-3.00	El proyecto no aplica todas las actividades de Gestión de Configuración de Software que se deben llevar a cabo durante el ciclo de desarrollo de un producto (software), ni las realiza con el nivel de madurez requerido
3.01-4.00	El proyecto aplica todas las actividades de Gestión de Configuración de Software que se deben llevar a cabo durante el ciclo de desarrollo de un producto (software) pero no con el nivel de madurez requerido
4.01-5.00	El proyecto aplica todas las actividades de Gestión de Configuración de Software que se deben llevar a cabo durante el ciclo de desarrollo de un producto (software) con el nivel de madurez requerido

Figura 27 Tabla Leyenda de Puntuación Promedio

2.5 Plantillas a modo de encuestas

A continuación se reflejan a modo de encuestas todos aquellos parámetros que recogen las diferentes plantillas que se proponen para la evaluación del proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6.

2.5.1 Encuesta Informe de Administración de Configuración

PREGUNTA 1

¿Se determina en su proyecto el tiempo disponible para realizar cada tarea específica?

Si ____

No ____

Parcialmente ____

PREGUNTA 2

¿Se determina en su proyecto el número de personas necesarias para realizar cada tarea específica?

Si ____

No ____

Parcialmente ____

PREGUNTA 3

¿Se determina en su proyecto los instrumentos necesarios para realizar cada tarea específica?

Si ____

No ____

Parcialmente ____

PREGUNTA 4

¿En su proyecto se registran todas las revisiones o aprobaciones de los diferentes estados por los que pasa el producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 5

¿En su proyecto se define el entorno tecnológico de soporte a la gestión de la configuración del sistema de información?

Si ____

No ____

Parcialmente ____

PREGUNTA 6

¿En su proyecto se determinan los componentes hardware y software que van a permitir la mecanización de los procesos y controles que establece el plan?

Si ____

No ____

Parcialmente ____

PREGUNTA 7

¿En su proyecto al finalizar cada proceso se identifica y registra el producto correspondiente?

Si ____

No ____

Parcialmente ____

PREGUNTA 8

¿En su proyecto cuando ocurre un cambio se registra en el sistema de Gestión de la Configuración con la versión nueva?

Si ____

No ____

Parcialmente ____

PREGUNTA 9

¿En su proyecto existe registro del tiempo, el cual se emplea en cada tarea realizada?

Si ____

No ____

Parcialmente ____

PREGUNTA 10

¿En su proyecto existen propuestas de cursos optativos relacionados con el lenguaje de programación?

Si ____

No ____

Parcialmente ____

PREGUNTA 11

¿En su proyecto existen propuestas de cursos optativos relacionados con el trabajo y manejo de las herramientas que se utilizan durante el desarrollo y avance del mismo?

Si ____

No ____

Parcialmente ____

PREGUNTA 12

¿En su proyecto existe un inventario de los instrumentos que el mismo pone a disposición de los integrantes?

Si ____

No ____

Parcialmente ____

PREGUNTA 13

¿En su proyecto existe una lista de todas las funcionalidades de la aplicación o producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 14

¿En su proyecto existe cronograma que muestra el nivel de prioridad que presenta cada tarea a ejecutar durante el ciclo de vida del mismo?

Si ____

No ____

Parcialmente ____

PREGUNTA 15

¿Se realizan con periodicidad reuniones en su proyecto con los miembros del equipo de trabajo del mismo?

Si ____

No ____

Parcialmente ____

PREGUNTA 16

¿En su proyecto existe un programa de seguimiento y control de errores?

Si ____

No ____

Parcialmente ____

Parámetros	Si	No	Parcialmente
Se determina el tiempo disponible para realizar cada tarea específica			
Se determina el número de personas necesarias para realizar cada tarea específica			
Se determinan los instrumentos necesarios para realizar cada tarea específica			
Existencia del registro del tiempo que se emplea en cada tarea realizada			
Existencia de propuestas de cursos optativos relacionados con el lenguaje de programación que se emplea en el proyecto			
Existencia de propuestas de cursos optativos relacionados con el trabajo y manejo de las herramientas que se utilizan durante el desarrollo y avance del proyecto			
Existencia de un inventario de los instrumentos que se ponen a disposición de los integrantes del proyecto			
Existencia de una lista con todas las funcionalidades de la aplicación o producto (software)			
Existencia de un cronograma que muestra el nivel de prioridad que presenta cada tarea a ejecutar durante el ciclo de vida del proyecto			
Periodicidad en reuniones con los miembros del equipo de trabajo del proyecto			
Existencia de un programa de seguimiento y control de errores			
Evaluación Final	Mal —	Regular —	Bien —

Figura 28 Modelo de Encuesta Informe de Administración de Configuración

2.5.2 Encuesta Informe de Medidas del Proyecto

PREGUNTA 1

¿Al finalizar la fase de Especificación de Requisitos del sistema global en su proyecto se establece y se revisa formalmente una Línea Base de Sistema?

Si ____

No ____

Parcialmente ____

PREGUNTA 2

¿Al finalizar la fase de Especificación de Requisitos en su proyecto se establece y se revisa formalmente una Línea Base Funcional?

Si ____

No ____

Parcialmente ____

PREGUNTA 3

¿Al finalizar la fase de Diseño Detallado en su proyecto se establece y se revisa formalmente una Línea Base de Diseño?

Si ____

No ____

Parcialmente ____

PREGUNTA 4

¿Al finalizar las fases de Codificación y Pruebas en su proyecto se establece y se revisa formalmente una Línea Base de Producto?

Si ____

No ____

Parcialmente ____

PREGUNTA 5

¿Al finalizar la fase de Instalación o Implantación en su proyecto se establece y se revisa formalmente una Línea Base Operativa?

Si ____

No ____

Parcialmente ____

PREGUNTA 6

¿Las Líneas Bases definidas en su proyecto sirven de soporte para un posterior desarrollo?

Si ____

No ____

Parcialmente ____

PREGUNTA 7

¿Las Líneas Bases definidas en su proyecto solo pueden ser modificadas a través de procedimientos formales de control de cambios?

Si ____

No ____

Parcialmente ____

Parámetros	Si	No	Parcialmente
Al finalizar la fase de Especificación de Requisitos del sistema global se establece y se revisa formalmente una Línea Base de Sistema			
Al finalizar la fase de Especificación de Requisitos se establece y se revisa formalmente una Línea Base Funcional			
Al finalizar la fase de Diseño Detallado se establece y se revisa formalmente una Línea Base de Diseño			
Al finalizar las fases de Codificación y Pruebas se establece y se revisa formalmente una Línea Base de Producto			
Al finalizar la fase de Instalación o Implantación se establece y se revisa formalmente una Línea Base Operativa			
Las Líneas Bases definidas sirven de soporte para un posterior desarrollo			
Las Líneas Bases definidas solo pueden ser modificadas a través de procedimientos formales de control de cambios			
Evaluación Final	Mal ___	Regular ___	Bien ___

Figura 29 Modelo de Encuesta Informe de Medidas del Proyecto

2.5.3 Encuesta Informe de Requisitos de Cambio

PREGUNTA 1

¿Los cambios realizados en su proyecto se clasifican de acuerdo al impacto y prioridad que pueden originar?

Si ___

No ___

Parcialmente ___

PREGUNTA 2

¿En su proyecto el porcentaje de cambios exitosos se enumeran por instancias?

Si ____

No ____

Parcialmente ____

PREGUNTA 3

¿En su proyecto se realizan revisiones después de la implementación?

Si ____

No ____

Parcialmente ____

PREGUNTA 4

¿En su proyecto se controla la cantidad de incidencias relacionadas con los cambios realizados?

Si ____

No ____

Parcialmente ____

PREGUNTA 5

¿Se controla el número de reuniones efectuadas en su proyecto donde se refleje toda la información asociada al número de cambios aprobados por reunión, asistencia, duración, etcétera?

Si ____

No ____

Parcialmente ____

PREGUNTA 6

¿Se identifican y anotan los elementos de su proyecto que deben estar bajo configuración?

Si ____

No ____

Parcialmente ____

PREGUNTA 7

¿En su proyecto se anotan las restricciones de cambios a dichos elementos?

Si ____

No ____

Parcialmente ____

PREGUNTA 8

¿En su proyecto se rastrean los cambios que se necesitan realizar a dichos elementos?

Si ____

No ____

Parcialmente ____

PREGUNTA 9

¿En su proyecto se registran las solicitudes para nuevas particularidades referentes al diseño y funcionalidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 10

¿En su proyecto se registran las solicitudes de los requerimientos que en algún momento fueron modificados por petición de los clientes o por conveniencia de los miembros del equipo de trabajo?

Si ____

No ____

Parcialmente ____

Parámetros	Si	No	Parcialmente
Los cambios realizados se clasifican de acuerdo al impacto y prioridad que pueden originar			
El porcentaje de cambios exitosos se enumeran por instancias			
Se realizan revisiones después de la implementación			
Se controla la cantidad de incidencias relacionadas con los cambios realizados			
Se controla el número de reuniones efectuadas donde se refleje toda la información asociada al número de cambios aprobados por reunión, asistencia, duración, etcétera			
Se identifican y anotan los elementos del proyecto que deben estar bajo configuración			
Se anotan las restricciones de cambios a dichos elementos			
Se rastrean los cambios que se necesitan realizar a dichos elementos			
Se registran las solicitudes para nuevas particularidades referentes al diseño y funcionalidad del producto (software)			
Se registran las solicitudes de los requerimientos que en algún momento fueron modificados por petición de los clientes o por conveniencia de los miembros del equipo de trabajo			
Evaluación Final	Mal —	Regular —	Bien —

Figura 30 Modelo de Encuesta Informe de Requisitos de Cambio

2.5.4 Encuesta Reporte de Elementos de Configuración

PREGUNTA 1

¿En su proyecto se capturan y almacenan con eficiencia el Plan de Proyecto, Contratos, Registros Históricos, etcétera?

Si ____

No ____

Parcialmente ____

PREGUNTA 2

¿En su proyecto las Líneas Bases son etiquetadas para mostrar la fase en que se encuentra el producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 3

¿En su proyecto se realizan pruebas de calidad para determinar la funcionalidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 4

¿En su proyecto se realizan pruebas de calidad para determinar la fiabilidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 5

¿En su proyecto se realizan pruebas de calidad para determinar la usabilidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 6

¿En su proyecto se realizan pruebas de calidad para determinar la eficiencia del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 7

¿En su proyecto se realizan pruebas de calidad para determinar la mantenibilidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 8

¿En su proyecto se realizan pruebas de calidad para determinar la portabilidad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 9

¿En su proyecto se realizan pruebas de calidad para determinar la efectividad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 10

¿En su proyecto se realizan pruebas de calidad para determinar la productividad del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 11

¿En su proyecto se realizan pruebas de calidad para determinar la seguridad física del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 12

¿En su proyecto se realizan pruebas de calidad para determinar la satisfacción del producto (software)?

Si ____

No ____

Parcialmente ____

PREGUNTA 13

¿Existe un área de trabajo de desarrollo en su proyecto?

Si ____

No ____

Parcialmente ____

PREGUNTA 14

¿En su proyecto el área de trabajo de desarrollo es capaz de crear un entorno privado?

Si ____

No ____

Parcialmente ____

PREGUNTA 15

¿En su proyecto se actualiza la Línea Base o punto de referencia principal: Fecha de inicio del proyecto?

Si ____

No ____

Parcialmente ____

PREGUNTA 16

¿En el proyecto se actualiza la Línea Base o punto de referencia principal: Fecha de finalización del proyecto?

Si ____

No ____

Parcialmente ____

PREGUNTA 17

¿En su proyecto se actualiza la Línea Base o punto de referencia principal: Duración de cada actividad del proyecto?

Si ____

No ____

Parcialmente ____

PREGUNTA 18

¿En su proyecto se actualiza la Línea Base o punto de referencia principal: Trabajo realizado en el proyecto?

Si ____

No ____

Parcialmente ____

PREGUNTA 19

¿En su proyecto se actualiza la Línea Base o punto de referencia principal: Registro de cada fase del proyecto?

Si ____

No ____

Parcialmente ____

Parámetros	Si	No	Parcialmente
Se capturan y almacenan con eficiencia el Plan de Proyecto, Contratos, Registros Históricos, etcétera			
Las Líneas Bases son etiquetadas para mostrar la fase en que se encuentra el producto (software)			
Se realizan pruebas de calidad para determinar la funcionalidad del producto (software)			
Se realizan pruebas de calidad para determinar la fiabilidad del producto (software)			
Se realizan pruebas de calidad para determinar la usabilidad del producto (software)			
Se realizan pruebas de calidad para determinar la eficiencia del producto (software)			
Se realizan pruebas de calidad para determinar la mantenibilidad del producto (software)			
Se realizan pruebas de calidad para determinar la portabilidad del producto (software)			
Se realizan pruebas de calidad para determinar la efectividad del producto (software)			
Se realizan pruebas de calidad para determinar la productividad del producto (software)			
Se realizan pruebas de calidad para determinar la seguridad física del producto (software)			
Se realizan pruebas de calidad para determinar la satisfacción del producto (software)			
Existencia de un área de trabajo de desarrollo			
El área de trabajo de desarrollo es capaz de crear un entorno privado			
Se actualiza la Línea Base o punto de referencia principal: Fecha de inicio del proyecto			
Se actualiza la Línea Base o punto de referencia principal: Fecha de finalización del proyecto			

Se actualiza la Línea Base o punto de referencia principal: Duración de cada actividad del proyecto			
Se actualiza la Línea Base o punto de referencia principal: Trabajo realizado en el proyecto			
Se actualiza la Línea Base o punto de referencia principal: Registro de cada fase del proyecto			
Evaluación Final	Mal —	Regular —	Bien —

Figura 31 Modelo de Encuesta Reporte de Elementos de Configuración

Conclusiones

Con la realización de este capítulo se ha logrado describir un procedimiento para la evaluación del proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6. Se definió un flujo de trabajo, junto con una serie de roles, actividades, tareas y artefactos que hicieron posible el diseño de la propuesta, así como el modelo de cuatro plantillas compuestas por una serie de parámetros medibles que hacen posible llevar a cabo la evaluación de todo el proceso de Gestión de Configuración de Software de los Proyectos Productivos de la facultad.

CAPÍTULO 3

VALIDACIÓN DEL PROCESO DE EVALUACIÓN

Se realiza la validación de la propuesta mediante el método de expertos para así poder evaluar el proceso de Gestión de Configuración del Software (GCS).

Introducción

Luego de haber definido todo el procedimiento para la evaluación del proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6, el presente capítulo se traza como objetivo principal validar la propuesta y determinar si la misma se ajusta a las necesidades actuales de la Facultad 6 y si es lo suficientemente correcta como para ser empleada por el grupo de calidad de dicha facultad para realizar evaluaciones sistemáticas a los proyecto productivos.

3.1 Validación del procedimiento para la evaluación de la GCS

La validación de un procedimiento para la evaluación de la Gestión de Configuración de Software de Proyectos Productivos puede realizarse de dos formas diferentes:

- ❖ Validación mediante una aplicación que permita llevar a cabo el proceso de evaluación.
- ❖ Validación mediante la evaluación por expertos del proceso evaluativo que se propone.

Esta última es la que se utiliza para evaluar el proceso de evaluación que se propone y describe durante el capítulo 2.

A continuación se muestra la encuesta realizada a tres personas con suficiente experiencia en la disciplina de Gestión de Configuración del Software como para validar la propuesta realizada:

ENCUESTA DE EVALUACIÓN CUALITATIVA POR EXPERTOS

Nombre: _____

Fecha: _____

PREGUNTA 1

¿Cree usted que la representación y descripción del flujo de trabajo que se propone, junto con las actividades, tareas, roles y artefactos que comprende se da solución al objetivo general del presente trabajo de diploma, que no es más que definir un procedimiento para evaluar el proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

PREGUNTA 2

¿Cree usted que el procedimiento que se propone para evaluar el proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6 encierra todos aquellos parámetros medibles pertenecientes a esta disciplina de Ingeniería de Software?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

PREGUNTA 3

¿Cree usted que la descripción de cada plantilla que se propone para llevar a cabo el proceso de evaluación está lo suficientemente detallada y legible?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

PREGUNTA 4

¿Cree usted que el sistema de puntuación que se emplea para evaluar cada parámetro de las diferentes plantillas es correcto y capaz de reflejar el nivel de madurez con que se realiza cada tarea en el proyecto?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

PREGUNTA 5

¿Cree usted que el sistema de puntuación cuantitativa que se emplea para evaluar cada plantilla se acerca más al verdadero nivel de aplicación y cumplimiento de la GCS en los diferentes proyectos de la facultad que al emplear el sistema de evaluación cualitativa de las mismas, reflejado en las plantillas a modo de encuestas?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

PREGUNTA 6

¿Cree usted que con el presente trabajo de diploma se da respuesta a la siguiente interrogante: cómo evaluar el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6?

Si ____

No ____

Parcialmente ____

a) Argumente si cree que es necesario

Parámetros	% de aceptación
1-El procedimiento propuesto da solución al objetivo general del presente Trabajo de Diploma	75
2-El procedimiento que se propone encierra todos los parámetros medibles de referentes a la disciplina de (GCS)	100
3-Las plantillas están lo suficientemente detalladas	100
4-El sistema de puntuación es correcto y capaz de reflejar el nivel de madurez con que se realiza cada tarea en el proyecto	100
5-El sistema de puntuación cuantitativo que se emplea es más exacto que el sistema de puntuación cualitativo	100
6-La propuesta realizada constituye un procedimiento válido y eficaz para llevar a cabo la evaluación del proceso de Gestión de Configuración del Software	75

Figura 32 Tabla Resumen del Porciento de Aceptación

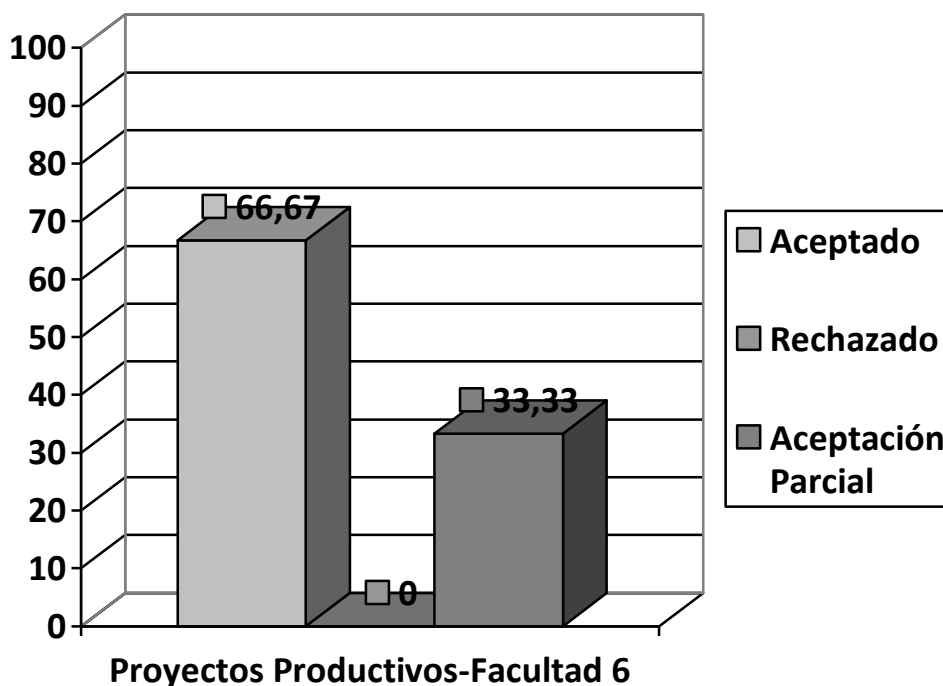


Figura 33 Gráfica porciento de aceptación de la validación del parámetro 1

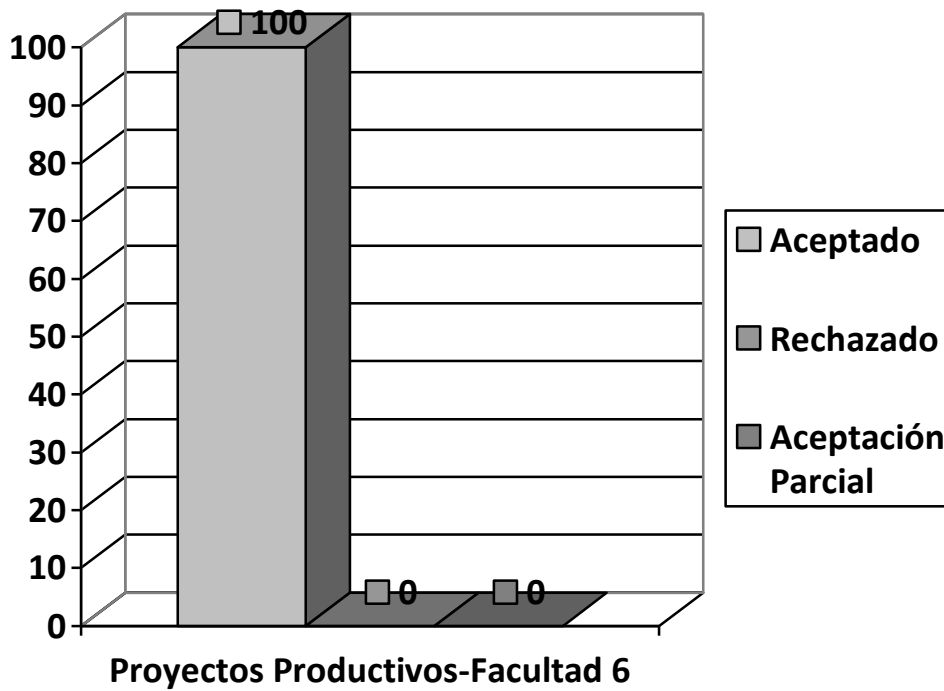


Figura 34 Gráfica por ciento de aceptación de la validación del parámetro 2

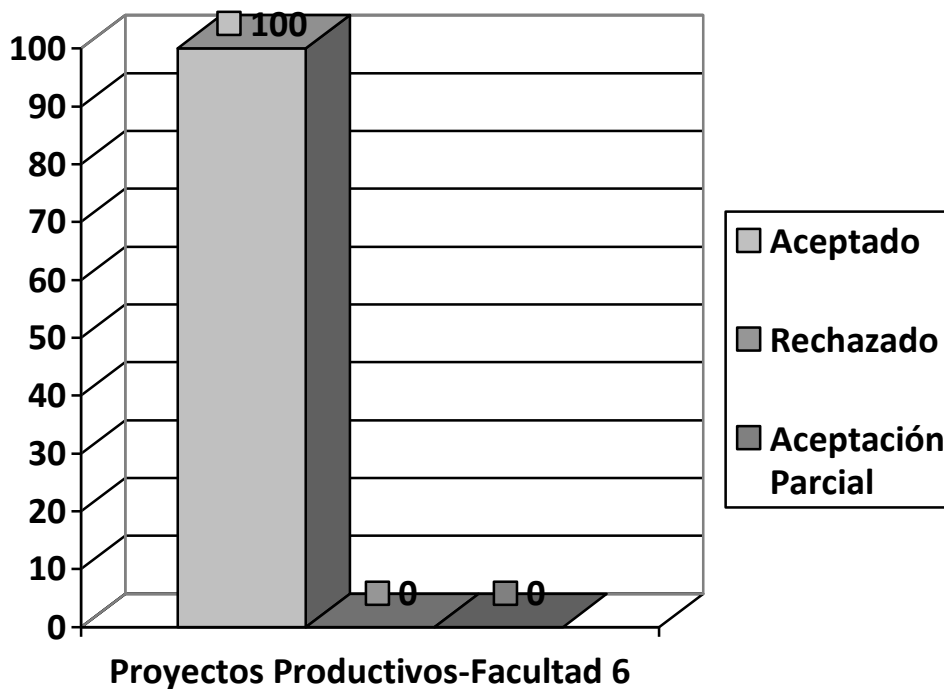


Figura 35 Gráfica por ciento de aceptación de la validación del parámetro 3

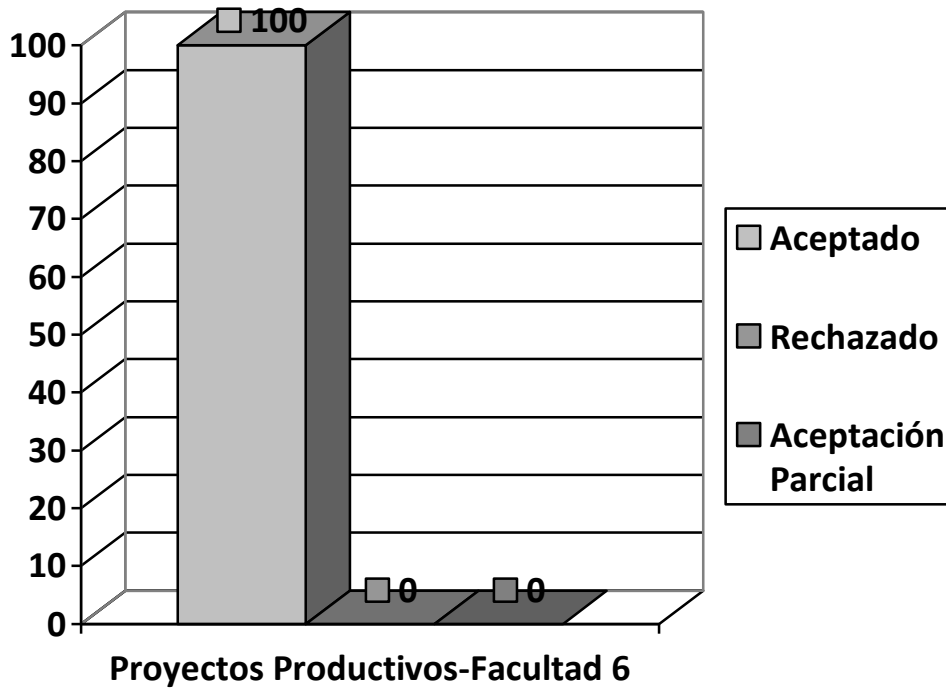


Figura 36 Gráfica por ciento de aceptación de la validación del parámetro 4

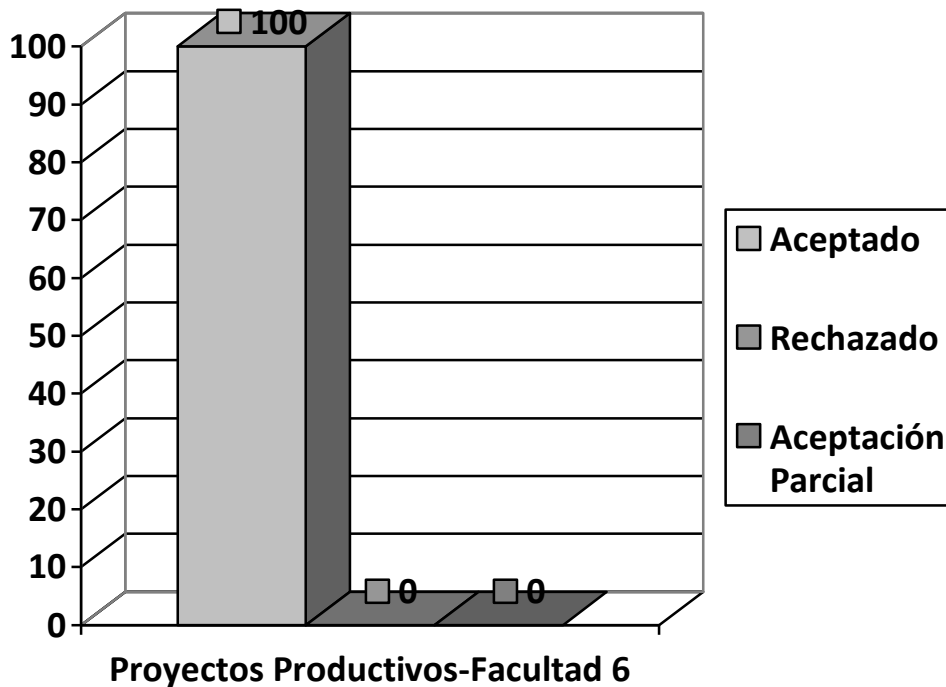


Figura 37 Gráfica por ciento de aceptación de la validación del parámetro 5

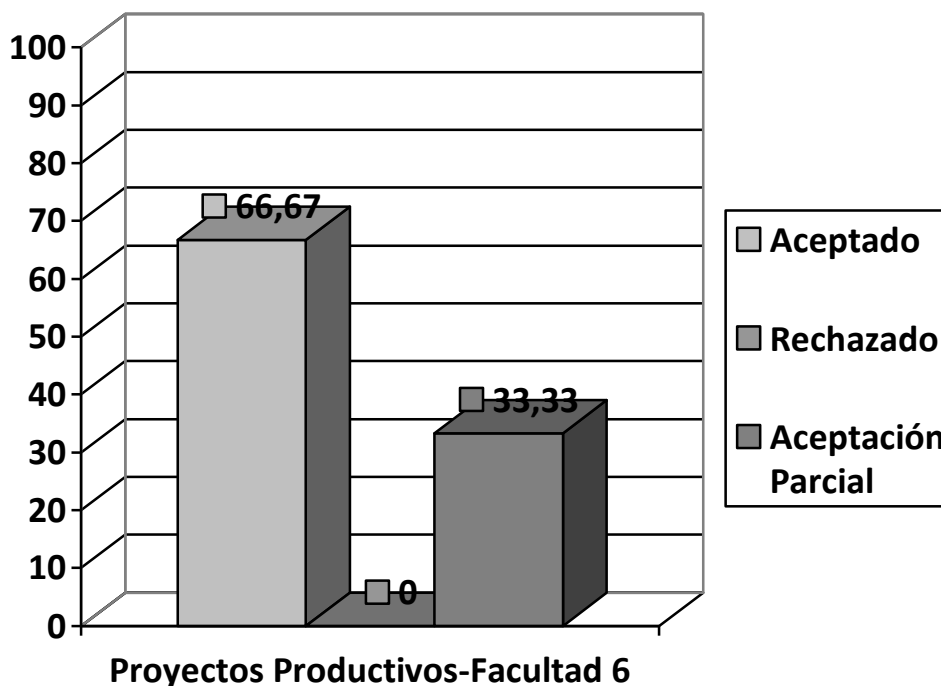


Figura 38 Gráfica por ciento de aceptación de la validación del parámetro 6

3.2 Conclusiones de la validación

Esta encuesta está compuesta por una serie de preguntas que engloban y resumen un objetivo general junto a varios objetivos específicos a cumplir con la definición del procedimiento que se propone para la evaluación de la Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6.

Se tiene en cuenta el criterio y argumentos que brindan los tres expertos a los cuales se les realizó la encuesta para llegar a las siguientes conclusiones:

- ❖ La representación y descripción del flujo de trabajo, junto con las actividades, tareas, roles y artefactos que se proponen y definen en el capítulo 2, proporcionan una solución al objetivo general del presente trabajo de diploma.
- ❖ El procedimiento que se propone para evaluar el proceso de Gestión de Configuración de Software de los Proyectos Productivos de la Facultad 6 encierra todos aquellos parámetros medibles pertenecientes a esta disciplina de Ingeniería de Software.
- ❖ La descripción de cada plantilla que se propone para llevar a cabo el proceso de evaluación está lo suficientemente detallada y legible.

- ❖ El sistema de puntuación que se emplea para evaluar cada parámetro de las diferentes plantillas es correcto y capaz de reflejar el nivel de madurez con que se realiza cada tarea en el proyecto.
- ❖ Con el sistema de puntuación cuantitativo que se emplea para evaluar cada plantilla se tiene una idea más real del nivel de aplicación y cumplimiento de la Gestión de Configuración de Software de los diferentes proyectos de la Facultad 6.
- ❖ Con el presente trabajo de diploma se da respuesta a la siguiente interrogante: ¿cómo evaluar el proceso de Gestión de Configuración de los Proyectos Productivos de la Facultad 6?

Por todo ello, se puede afirmar que mediante el proceso de evaluación propuesto se puede determinar detalladamente problemas de la situación actual de los Proyectos Productivos de la Facultad 6.

Por ejemplo, mediante el procedimiento que se propone para la evaluación se puede determinar si los proyectos de la facultad:

- ❖ Realizan Control de Versiones.
- ❖ Llevan a cabo las actividades que se deben desarrollar durante la Gestión de Configuración de Software.
- ❖ Determinan de forma periódica sus líneas bases.
- ❖ Efectúan de forma periódica el control de cambios que se realizan en el producto (software).

Por lo tanto, se puede asegurar que el procedimiento al que anteriormente se hace referencia, cumple con los requisitos establecidos y metas trazadas. En fin, el proceso de evaluación propuesto constituye un procedimiento eficaz y con gran relación con los complejos y extensos Proceso de Desarrollo y Calidad del Software, ya que interviene directa y determinantemente en el ritmo de confección y calidad del producto (software), así como una guía por la cual el grupo de calidad de la Facultad 6 puede de forma sistemática llevar a cabo una auditoría de los Proyectos Productivos.

Conclusiones

La satisfactoria validación de la propuesta realizada en este capítulo proporciona un procedimiento efectivo e importante para determinar si el producto (software) o proyecto se rige por las buenas prácticas de Ingeniería de Software; además de detectar todos aquellos problemas que originan impactos negativos durante el desarrollo del ciclo de vida del producto (software).

CONCLUSIONES GENERALES

El desarrollo de productos (software) con calidad, se ha convertido en una necesidad en la actualidad debido al creciente y vertiginoso crecimiento de la Industria del Software a nivel global.

Como resultado de la investigación realizada de todos aquellos conceptos, normas, modelos, metodologías y herramientas que hacen referencia a la Gestión de Configuración del Software (GCS) como disciplina de la Ingeniería de Software conjuntamente con la realización de la propuesta para la evaluación de la Gestión de Configuración de los Proyectos Productivos de la Facultad 6, se llegaron a las siguientes conclusiones:

- ❖ Se definió un flujo de trabajo que permite llevar a cabo el proceso de evaluación que se propone.
- ❖ Se definieron cuatro actividades mediante las cuales se organiza todo el procedimiento para la evaluación propuesta.
- ❖ Se definieron siete tareas que figuran estructuras lógicas de control que precisan el ámbito y administran el flujo de la ejecución del flujo de trabajo propuesto.
- ❖ Se definieron siete artefactos que designan cualquier tipo de información generada por los Proyectos Productivos durante el proceso de evaluación que se propone.
- ❖ Se definieron cuatro roles que son los responsables de evaluar todas aquellas actividades realizadas en los Proyectos Productivos que promueven las buenas prácticas en la Gestión de Configuración del Software.
- ❖ Se definieron trece técnicas o procedimientos que se emplean para desarrollar cada una de las tareas del flujo de trabajo que se propone.
- ❖ Se establecieron cuatro modelos de plantillas que hacen posible de forma detallada realizar la evaluación propuesta en los Proyectos Productivos de la Facultad 6.
- ❖ Se validó el procedimiento propuesto, lo que demuestra que el proceso de evaluación es eficaz y completamente aplicable en los Proyectos Productivos de la Facultad 6.

De esta forma se le dio cumplimiento al objetivo general, así como a los objetivos específicos planteados en el presente Trabajo de Diploma.

RECOMENDACIONES

Para garantizar un correcto desempeño del procedimiento de evaluación que se propone se recomienda:

- ❖ Que todo el personal involucrado y responsable de llevar a cabo el proceso de evaluación propuesto esté totalmente capacitado en las actividades de Gestión de Configuración de Software.
- ❖ Que se desarrolle una aplicación informática que sea capaz de generar las plantillas de evaluación cuantitativa y cualitativa que se definen, de esta forma se agilizará el proceso de evaluación que se propone.
- ❖ Que el procedimiento que se define para llevar a cabo la evaluación de la Gestión de Configuración de los Proyectos Productivos de la Facultad 6 se extienda y aplique a otros proyectos en otras facultades de la Universidad de las Ciencias Informáticas (UCI).

REFERENCIAS

1. **Rancán, Claudio.** Trabajo Final Especialidad en Control y Gestión de Software. Gestión de Configuración de Productos Software e Etapa de Desarrollo. *itba.edu*. [En línea] julio de 2003. <http://www.itba.edu.ar/capis/epg-tesis-y-tf/rancan-trabajofinaldeespecialidad.pdf>.
2. **Casañola, Yaimí Trujillo.** Propuesta de Modelo de Producción de Software para la Universidad de las Ciencias Informáticas. *informaticahabana*. [En línea] 2007. http://www.informaticahabana.com/evento_virtual/?q=node/160&ev=III%20Taller%20Internacional%20de%20Calidad%20en%20las%20TICs.
3. **Grisel, Infante Costa.** *Análisis de la Calidad del Proceso de Desarrollo de Software en el Proyecto Akademos*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. [ed.] Andrés Otero. Madrid : Pearson Educación. S. A., 2000. pág. 464. 84-7829-036-2.
5. **Kindelán Ferrá, Adonis Alexey.** *Proceso de Gestión de la Configuración de Software. Propuesta para su implementación en el proyecto de Atención Primaria de la de la Salud (APS)*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
6. **Presman, Roger S.** *Ingeniería del Software. Un enfoque práctico*.
7. **Fernández Carrasco, Oscar M., García León, Delva y Beltrán Benavides, Alfa.** Un enfoque actual sobre la calidad del software. *bvs.sld*. [En línea] septiembre-diciembre de 1995. http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
8. **Valeris, Ismel Herrera.** *Propuesta de Control para la Gestión de la Calidad del*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
9. **Sánchez Rosas, Juan Eladio.** Principios de Control de Versiones con Subversion. *tuxpuc.pucp.edu*. [En línea] 4 de noviembre de 2007. <http://tuxpuc.pucp.edu.pe/content/view/789/12/>.
10. **Vargas, Odette Gordillo.** *Propuesta para la aplicación de la Gestión de Configuración de Software en el proceso productivo de la UCI*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
11. Soluciones Informáticas. *ixis*. [En línea] http://www.ixis.net/IxisNet/OPEN_SOURCE/CVS.

12. **A., Ernesto Quiñones.** Modelos de Calidad de Software y Software Libre. *eqsoft*. [En línea] 2003-2008. http://www.eqsoft.net/presentas/modelos_de_calidad_y_software_libre.pdf.
13. **Anaya, Raquel, Jaramillo, Carlos Andrés y Fernando Londoño, Luis.** El proceso de desarrollo de software: una visión desde la academia y la industria. *eafit*. [Online] abril 21, 2005. <http://www.eafit.edu.co/NR/rdonlyres/8CA48405-0691-4B9C-A178-6A59754C0BDA/0/Proceso2.pdf>.
14. **Añasco, Juan Carlos.** Preguntas y respuestas acerca de las normas ISO 9000. *pilar*. [En línea] marzo de 2006. <http://www.pilar.com.ar/industrias/temasgenerales/normas.htm#¿Cúal%20es%20el%20origen>.
15. Normas ISO 9000 y Calidad. *normas-iso-9000.blogspot*. [En línea] <http://normas-iso-9000.blogspot.com/2007/11/que-contiene-y-porque-es-importante-el.html>.
16. **López Pérez, Carmelo.** Modelo de Madurez de la Capacidad del Software. *cii-murcia*. [En línea] http://www.cii-murcia.es/informas/ene05/articulos/Modelo_de_Madurez_de_la_Capacidad_del_Software.html.
17. **CIENTEC.** CMMI: Mejorando Procesos en Forma Integrada. *cientec*. [En línea] 13 de diciembre de 2006. <http://www.cientec.com/analisis/cmml.asp>.
18. **Castillo F., Luis A.** Producción de Software. *uvmsf*. [En línea] 2008. <http://www.uvmsf.cl/~lcastillo/media/3-Produc.pdf>.
19. **de la Villa, Manuel, Ruiz, Mercedes y Ramos, Isabel.** Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo. *ftp.informatik.rwth-aachen*. [En línea] noviembre de 2004. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-120/paper4.pdf>.
20. **Perez-Vargas, Mariana y Serrano, Miguel.** Evaluación de procesos de software con SCAMPI. *sg.com*. [En línea] 14 de octubre de 2007. <http://www.sg.com.mx/sg07/media/SG07.LAB10.Evaluacion%20con%20SCAMPI.pdf>.
21. **Lac Prugent, Carlos.** Six Sigma: mejora de procesos. *clarin*. [En línea] 7 de marzo de 2004. <http://www.clarin.com/suplementos/economico/2004/03/07/n-00602.htm>.
22. **López, Gustavo.** Metodología Six-Sigma: Calidad Industrial. *mercadeo*. [En línea] 22 de julio de 2002. <http://www.mercadeo.com/archivos/six-sigma.pdf>.

23. *usm.edu*. [En línea] <http://www.usm.edu.ec/abedini/spice/spicess.htm>.
24. **Ministerio de Administraciones Públicas**. Gestión de la Configuración. *csi.map*. [En línea] <http://www.csi.map.es/csi/metrica3/gescon.pdf>.
25. **Hista Internacional**. Gestión de Configuración del Software. *histaintl*. [En línea] 27 de febrero de 2007. <http://www.histaintl.com/soluciones/configuracion/configuracion.php>.

BIBLIOGRAFÍA

1. **A., Ernesto Quiñones.** Modelos de Calidad de Software y Software Libre. *eqsoft*. [En línea] 2003-2008. http://www.eqsoft.net/presentas/modelos_de_calidad_y_software_libre.pdf.
2. **Almeraz, Elizabeth.** Lecciones aprendidas y estrategias de Implantación de CMMI en México. *software.net*. [En línea] agosto de 2006. <http://www.software.net.mx/NR/rdonlyres/BBE8566F-CA03-45F2-8C2E-985F830FA9E2/1366/ElizabethAlmeraz.pdf>.
3. **Anaya, Raquel, Jaramillo, Carlos Andrés y Fernando Londoño, Luis.** El proceso de desarrollo de software: una visión desde la academia y la industria. *eafit*. [En línea] 21 de abril de 2005. <http://www.eafit.edu.co/NR/rdonlyres/8CA48405-0691-4B9C-A178-A59754C0BDA/0/Proceso2.pdf>.
4. **Añasco, Juan Carlos.** Preguntas y respuestas acerca de las normas ISO 9000. *pilar*. [En línea] marzo de 2006. <http://www.pilar.com.ar/industrias/temasgenerales/normas.htm#¿Cúal%20es%20el%20origen>.
5. **Arenales Gómez, Esther.** La Calidad como Herramienta de Mejora en Organizaciones de Desarrollo de Software. *ingeniusteam*. [En línea] diciembre-enero de 2006-2007. http://www.ingeniusteam.com/index.aspx?pag=texto_noticia.aspx?ref=78.
6. **Buades Rubio, Gabriel.** Ingeniería del Software III. *dmi.uib*. [En línea] febrero de 1999. <http://dmi.uib.es/~bbuades/workflow/sld005.htm>.
7. **Canela Vicente, Juan.** Gestión y Calidad del Software. *kybele.escet.urjc*. [En línea] 2007-2008. http://kybele.escet.urjc.es/docencia/GCSW/2007-2008/Material/%5BGCSW-2006-07%5DTema4_Calidad_de_Producto.pdf.
8. **Carretero López, Rosa María.** Introducción a Rational ClearCase LT. *dsi.uclm*. [En línea] http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo_ClearCase.pdf.
9. **Casañola, Yaimí Trujillo.** Propuesta de Modelo de Producción de Software para la Universidad de las Ciencias Informáticas. *informaticahabana*. [En línea] 2007. http://www.informaticahabana.com/evento_virtual/?q=node/160&ev=III%20Taller%20Internacional%20de%20Calidad%20en%20las%20TICs.
10. **Castillo F., Luis A.** Producción de Software. *uvmsf*. [En línea] 2008. <http://www.uvmsf.cl/~lcastillo/media/3-Produc.pdf>.

11. **CIENTEC.** CMMI: Mejorando Procesos en Forma Integrada. *cientec*. [En línea] 13 de diciembre de 2006. <http://www.cientec.com/analisis/cmml.asp>.
12. **codicesoftware.** Tabla comparativa con otras herramientas SCM. *codicesoftware*. [En línea] 2007. <http://www.codicesoftware.com/xeevalresources2/xecomparisontable.aspx>.
13. **Costa, Grisel Infante.** *Análisis de la Calidad del Proceso de Desarrollo de Software en el Proyecto Akademos*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
14. **Daniele, Marcela, y otros.** Departamento de Computación. *dc.exa.unrc.edu*. [En línea] 2007. <http://dc.exa.unrc.edu.ar/nuevodc/materias/ingenieria/2007>.
15. **de la Villa, Manuel, Ruiz, Mercedes y Ramos, Isabel.** Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo. *ftp.informatik.rwth-aachen*. [En línea] noviembre de 2004. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-120/paper4.pdf>.
16. **Det Norske Veritas.** Sistemas de Gestión. *dnv*. [En línea] <http://www.dnv.es/certificacion/sistemasdegestion/calidad/>.
17. **Fernández Carrasco, Oscar M., García León, Delva y Beltrán Benavides, Alfa.** Un enfoque actual sobre la calidad del software. *bvs.sld*. [En línea] septiembre-diciembre de 1995. http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
18. **Gervás, Pablo.** Estandaresy gestión de configuración. *profesores.sanvalero*. [En línea] octubre de 2003. <http://profesores.sanvalero.net/~w0172/Calidad%20de%20Software/Gestion%20Configuracion/04%2520Estandares%2520y%2520Gestion%2520de%2520configuracion.pdf>.
19. **Granollers i Saltiveri, Toni.** MPlu+a. Una metodología que integra la Ingeniería del Software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares. *griho2.udl*. [En línea] julio de 2004. http://griho2.udl.es/publicacions/2004/Tesis_Toni/Tesis_T_Granollers_caps0-1.pdf.
20. **Gutiérrez Mayoral, Antonio.** Ingeniería Técnica en Informática de Sistemas. *gsync*. [En línea] febrero de 2005. <http://gsync.es/~agutierr/pfc-tecnica-html/node3.html>.
21. **Hista Internacional.** Gestión de Configuración del Software. *histaintl*. [En línea] 27 de febrero de 2007. <http://www.histaintl.com/soluciones/configuracion/configuracion.php>.

22. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* [ed.] Andrés Otero. Madrid : Pearson Educación. S. A., 2000. pág. 464. 84-7829-036-2.
23. **Kindelán Ferrá, Adonis Alexey.** *Proceso de Gestión de la Configuración de Software. Propuesta para su implementación en el proyecto de Atención Primaria de la de la Salud (APS).* Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
24. **la Torre Hernández, Ludisley y Cepero Nuñez, Mariela.** *Propuesta de Métricas para perfeccionar la Gestión de la Calidad en los Procesos de Desarrollo de Software.* Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
25. **Lac Prugent, Carlos.** Six Sigma: mejora de procesos. *clarin.* [En línea] 7 de marzo de 2004. <http://www.clarin.com/suplementos/economico/2004/03/07/n-00602.htm>.
26. **Letelier Torres, Patricio.** Mis Presentaciones y Cursos. *dsic.upv.* [En línea] abril de 2004. <http://www.dsic.upv.es/~letelier/pub/>.
27. **López Pérez, Carmelo.** Modelo de Madurez de la Capacidad del Software. *cii-murcia.* [En línea] http://www.cii-murcia.es/informas/ene05/articulos/Modelo_de_Madurez_de_la_Capacidad_del_Software.html.
28. **López, Gustavo.** Metodología Six-Sigma: Calidad Industrial. *mercadeo.* [En línea] 22 de julio de 2002. <http://www.mercadeo.com/archivos/six-sigma.pdf>.
29. **Matamoros Ramírez, Ofelia.** Normas ISO 9000: su Base. *una.ac.* [En línea] noviembre de 2004. <http://www.una.ac.cr/bibliotecologia/boletinbiblioteca/1999/Normas.doc>.
30. **Méndez, Carlos.** Introducción al Modelo CMMI. *acis.org.* [En línea] http://www.acis.org.co/fileadmin/Conferencias/IntroduccionCMMI_CarlosMendez.pdf.
31. **Ministerio de Administraciones Públicas.** Gestión de la Configuración. *csi.map.* [En línea] <http://www.csi.map.es/csi/metrica3/gescon.pdf>.
32. Modelos de Gestion de la Calidad del Software. *modelosdegestiondelacalidad.blogspot.* [En línea] enero de 2007-2008. <http://modelosdegestiondelacalidad.blogspot.com/>.

33. **Moreno García, María N., García Peñalvo, Francisco J. y Polo Martín, María José.** Medición de la calidad del software en el ámbito de la especificación de requisitos. *sc.ehu*. [En línea] 2000. <http://www.sc.ehu.es/jiwdocoj/remis/docs/mmg-2000.ppt>.
34. Normas ISO 9000 y Calidad. *normas-iso-9000.blogspot*. [En línea] <http://normas-iso-9000.blogspot.com/2007/11/que-contiene-y-porque-es-importante-el.html>.
35. **Palacio Bañeres, Juan.** Visión ejecutiva de procesos y prácticas para desarrollo de software. *navegapolis*. [En línea] diciembre de 2005. http://www.navegapolis.net/files/presentaciones/iempresa_05_12_17.ppt.
36. **Perez-Vargas, Mariana y Serrano, Miguel.** Evaluación de procesos de software con SCAMPI. *sg.com*. [En línea] 14 de octubre de 2007. <http://www.sg.com.mx/sg07/media/SG07.LAB10.Evaluacion%20con%20SCAMPI.pdf>.
37. Prácticas y métodos para mejorar el desarrollo de Proyectos de Software. *ingenierosoftware*. [En línea] 2003-2008. <http://www.ingenierosoftware.com/>.
38. **Presman, Roger S.** *Ingeniería del Software. Un enfoque práctico*.
39. **Rancán, Claudio.** Trabajo Final Especialidad en Control y Gestión de Software. Gestión de Configuración de Productos Software e Etapa de Desarrollo. *itba.edu*. [En línea] julio de 2003. <http://www.itba.edu.ar/capis/epg-tesis-y-tf/rancan-trabajofinaldeespecialidad.pdf>.
40. **Sánchez Rosas, Juan Eladio.** Principios de Control de Versiones con Subversion. *tuxpuc.pucp.edu*. [En línea] 4 de noviembre de 2007. <http://tuxpuc.pucp.edu.pe/content/view/789/12/>.
41. Soluciones Informáticas. *ixis*. [En línea] http://www.ixis.net/lxisNet/OPEN_SOURCE/CVS.
42. *usm.edu*. [En línea] <http://www.usm.edu.ec/abedini/spice/spicess.htm>.
43. **Valencia, María Eugenia.** Gestión de la onfiguración del Software. *eisc.univalle.edu*. [En línea] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/GESTION%20DE%20LA%20CONFIGURACION.pdf.
44. **Valeris, Ismel Herrera.** *Propuesta de Control para la Gestión de la Calidad del*. Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.

45. **Vargas, Odette Gordillo.** *Propuesta para la aplicación de la Gestión de Configuración de Software en el proceso productivo de la UCI.* Ciudad de la Habana : s.n., 2007. Trabajo de Diploma.
46. **Villarroel, Rodolfo y Visconti, Marcello.** Un Modelo de Madurez para el Proceso de Gestión de Configuración de Software. *inf.utfsm.* [En línea] noviembre de 2008. <http://www.inf.utfsm.cl/~visconti/papers/paperscm1999.pdf>.

GLOSARIO DE TÉRMINOS

Artefacto: Cualquier tipo de información generada, cambiada o usada por los trabajadores del proyecto.

ASI: Análisis del Sistema de Información.

Auditoría: Análisis crítico y sistemático que se realiza en un Proyecto Productivo.

BioSyS: Proyecto Productivo de Sistemas Biológicos.

CASE: Ingeniería de Software Asistida por Computación.

CD-ROOM: Soporte digital que se utiliza para almacenar cualquier tipo de datos.

CIGB: Centro de Ingeniería Genética y Biotecnología.

CIM: Centro de Inmunología Molecular.

CMMI: Capability Maturity Model Integration.

CSI: Construcción del Sistema de Información.

CVS: Sistema de Versiones Concurrente.

DSI: Diseño del Sistema de Información.

ECS: Elementos de Configuración de Software.

Evaluación: Proceso que se lleva a cabo para medir el nivel de madurez y cumplimiento de las actividades en un Proyecto Productivo.

Evaluador: Persona que lleva a cabo el proceso de evaluación.

Floppy: Disco flexible o disco de 3 ½.

GCS: Gestión de Configuración del Software.

GRaph-TOol: Proyecto Productivo de trabajo con Herramientas Gráficas.

Herramienta: Producto informático que sirve de soporte para desarrollar nuevos productos (software).

IAS: Implantación y Aceptación del Sistema.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

Industria del Software: Industria que comprende el desarrollo, comercialización y distribución de productos (software).

Informe o reporte: Documento que contiene información importante.

IPPD: Desarrollo integrado de producto y proceso.

ISO 9000: Conjunto de cinco normas genéricas y no específicas relacionadas entre sí.

LDAP: Protocolo Ligero de Acceso a Directorios.

LIMS CC: Proyecto Productivo Control de la Calidad.

Línea Base: Producto o documento que definen los Proyectos Productivos como base para desarrollos posteriores.

Metodología: Conjunto de procesos y métodos.

MOPROSOFT- Mexico: Modelo de procesos para la industria de software de México.

OO o P.O.O: Referente a Programación Orientada a Objeto.

Parámetro: Argumento que se emplea para guiar el proceso de evaluación.

Plantilla: Instrumento que permite guiar, portar o diseñar un esquema predeterminado.

Producto (software): Aplicación informática o producto desarrollado mediante computadores.

Proyectos Productivos: Equipo de trabajo compuesto por personas con el objetivo de desarrollar productos (software).

RCS: Sistema de Control de Revisiones.

Requisito: Característica que debe cumplir todo producto (software).

Rol: Papel que desempeña una determinada persona en un Proyecto Productivo.

RUP: Proceso Unificado de Desarrollo.

SAMAD: Proyecto Productivo Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias.

SCAMPI: Standard CMMI Appraisal Method for Process Improvement.

SEI: Instituto de Ingeniería de Software.

SIMDEC: Proyecto Productivo Sistema de Manejo de Datos de Ensayos Clínicos.

Six-Sigma: Metodología rigurosa, integradora y altamente efectiva.

SPARC: Arquitectura de Procesador Escalable.

SPICE: Software Process Improvement and Capability dEtermination.

SQA: Garantía o Aseguramiento de Calidad del Software.

Stakeholder: Personal involucrado con el Proyecto Productivo, se puede utilizar para referirse a proveedores y clientes.

Tarea: Unidad primaria de modelación.

Técnica o procedimiento: Definición de pasos lógicos para realizar una ocupación o trabajo correctamente.

UCI: Universidad de las Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado.

Validar: Aprobar, dar fuerza o firmeza a un procedimiento propuesto.

VCS: Sistema de Control de Versiones.

Workflow: Flujo de Trabajo u operacionales de una actividad de trabajo.