Universidad de las Ciencias Informáticas

Facultad 3

Título: Propuesta de una estrategia de Aseguramiento de la Calidad para el proyecto Convenio Cuba-Venezuela para su segunda fase.

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

Autores: Raúl Velázquez Alvarez
Olga Lidia Rodríguez Martínez

Tutor(es): Ing. Dayma Direntau Batista

Co-Tutor: Lic. Rolan Rober Bullain Diéguez

Junio 2008

"La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible."

Ernesto Che Guevara

DECLARACIÓN DE AUTORÍA

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

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

Autor: Olga Lidia Rodríguez Martínez

_____

Autor: Raúl Velázquez Alvarez

_____

Tutor(a): Dayma Direntau Batista

_____

Co – tutor: Rolan Rober Bullain Diéguez

_____

DATOS DE CONTACTO

Lic. Ciencias de la Computación

Centro de trabajo: Universidad de Ciencias Informáticas

Categoría docente: Instructor

Asignatura que imparte: Programación I, II, IV.

Aspirante al grado científico de doctor en ciencias.

Línea de investigación: Enseñanza de la Programación

Participación en eventos: -

AGRADECIMIENTOS

A todas las personas que nos han apoyado en la realización de este trabajo de diploma, y en especial a:

La Revolución Cubana por hacer posible este sueño hecho realidad que es la Universidad de las Ciencias Informáticas, escuela de vida y de futuro.

Nuestro Querido Fidel, por concebir esta idea maravillosa.

Nuestros profesores, que nos abrieron el camino del conocimiento.

El profesor Lic. Rolan Rober Bullain Diéguez, por apoyarnos y ser nuestro Co-Tutor.

La Dra. Olga Lidia Martínez Leyet, por su apoyo en la revisión de la concepción metodológica de la tesis.

A Yeilín por proporcionarnos todo, todo su apoyo.

A todos los especialistas que donaron parte de su tiempo para validar la propuesta hecha.

DEDICATORIA

En este día tan especial para mí, como es la culminación de mis estudios en la Universidad de las Ciencias Informáticas, dedico este trabajo a todos los que de una forma u otra estuvieron siempre conmigo:

A mi familia que durante estos 5 años ha estado más cerca que ninguna para lo que me ha hecho falta.

A mi papá por su espíritu tan especial y por contagiar a todo el que lo rodea. Porque estoy orgullosa del él, por haberse convertido en uno de los mejores camarógrafos de la UCI y aprender junto conmigo.

A mi hermana que aún le queda un largo camino por recorrer y tengo fe en que al igual que yo lo va a lograr.

A mi madre, por darme su apoyo, ser mi amiga, mi tutora y mi todo.

A mi novio Manuel, que se ha convertido en lo más especial que me ha sucedido en la vida.

A todos mis compañeros de estudio, en especial a Mariam, LLily, Any y Giny que siempre han estado junto a mí.

A Raúl (Mackey) que más que un amigo ha sido otro hermano para mí, y que con este trabajo se terminan nuestras agonías.

Olga Lidia Rodríguez Martínez

DEDICATORIA

Bueno ya es bastante difícil superar la dedicatoria anterior pero por lo menos intentaré estar al mismo nivel jejejeje.

Dedico este trabajo (que valga la redundancia gran trabajo que me dio hacerlo) a todas las personas que han formado parte de mi formación y que han contribuido de una forma u otra a que este día pueda decir "Ya soy Ingeniero".

Primero que todo a mis padres y de manera especial a mi queridísima madre, que desde que entré al mundo de la enseñanza no se ha apartado de mí y siempre ha estado pendiente de todo, que debajo de aguaceros y fangueros ha estado a mi lado como una estudiante más, con un maletín al hombro o con una java de comida en la mano.

A mi hermana y a mis abuelos queridos.

A mis amigos especiales del pre que aunque no todos están aquí, todavía los recuerdo y les agradezco todo lo que hicieron por mí: a Giselle, Karen, Adis, Zucel, Yoan, Dasiel, Yasiel, Liudma, Yaíma, Malule y Marilyn.

A mis amigos de la uni por soportarme y por tomarse muy apecho la palabra "amigo" y darme trabajo de vez en cuando, pero por sobre todas las cosas a acompañarme en los momentos buenos y malos: a Mariam, Llilian, Anisbert, Giny, Yamira, Ana (Momby), Oreste y Mailen.

A mi querida profesora, tutora y jefa Dariela, que siempre me ha apoyado, defendido, enseñado y ayudado en todo lo que necesité en estos años de estudio.

Y de manera muy especial también:

A mi segunda familia, los padres de Olguita, que en estos años me han acogido en su seno familiar.

A mi flaquyyyyyyy (Olguita), que ha sido mi hermana, mi apoyo, mi proveedora de necesidades materiales jejejeje y sobre todo mi más grande amiga.

A una persona que no puedo decir su nombre.

A mí mismo, porque si no hubiera sido por mí gran parte del trabajo no se hubiera hecho jejejeje.

Raúl Velázquez Álvarez

RESUMEN

En el presente trabajo se hace la proposición de una estrategia de aseguramiento de la calidad para el proyecto Convenio Cuba-Venezuela, para ello fue necesario realizar un estudio sobre las deficiencias y dificultades existentes actualmente en los proyectos de desarrollo de software. Esto permitió detectar los principales puntos débiles que atentan contra la calidad del producto final a los cuales la estrategia estuvo encaminada a resolver.

El análisis del concepto de calidad de software, de revisiones técnicas formales, de modelos y estándares de calidad y de métricas, entre otros aspectos; posibilitó en gran medida obtener el conocimiento necesario para poder desarrollar la propuesta. La misma contribuirá a que el proceso de desarrollo de software esté orientado a obtener un producto con adecuados niveles de calidad.

Esta estrategia está conformada por tres aspectos fundamentales, el primero de ellos está dirigido al chequeo y revisión de los diferentes planes que se generan a lo largo del ciclo de vida del software en sus cuatro fases de desarrollo. Se plantea para ellos una serie de revisiones que debe realizar el equipo de calidad en dichos documentos. Luego se propone el Plan de Aseguramiento de la Calidad y se concluye con un grupo de actividades para hacer cumplir de manera satisfactoria cada aspecto del mismo. Esta estrategia se ha validado positivamente con especialistas en el tema que han dado un criterio acertado sobre la misma, lo que da la posibilidad de que pueda ser aplicable.

PALABRAS CLAVE

Aseguramiento, Calidad, Estrategia, Gestión, Plan

ABSTRACT

The objective of this work was to elaborate a Software Quality Assurance Strategy for the project Agreement Cuba-Venezuela. For that, it was necessary to carry out a study about the deficiencies and existent difficulties in the projects of software development. This allowed to detect the main weak points that attempt against the quality of the final product, problems to be solved with the proposed strategy.

The analysis of the concept of software quality, of formal technical revisions, of models and standards of quality and of metrics, among other aspects facilitated in great measure the assimilation of the necessary knowledge to be able to develop the proposal. The strategy will contribute to the process of software development, guiding the whole process to obtain a product with appropriate levels of quality.

This strategy is conformed by three fundamental aspects. The first is directed to the checkup and revision of the different plans that are generated along the life cycle of the software in its four development phases. A series of revisions of these documents should be carry out by the quality team. Then a Plan of Quality Assurance is proposed and as conclusion a group of activities is developed to complete in a satisfactory way each aspect of it. This strategy has been validated positively by specialists in software quality, who have expressed positive criteria about the feasibility of the application of the proposal.

KEY WORDS

Assurance, Quality, Strategy, Management, Plan

TABLA DE CONTENIDOS

INTRODUCCIÓN

Actualmente los retos que afrontan las empresas las obligan a ser más competitivas y a enfrentar los nuevos mercados que supone la globalización económica. Los consumidores exigen cada vez más productos mejor elaborados y que satisfagan sus necesidades reales. Esto ha incidido en que el concepto de calidad se haya convertido en elemento clave en todos los procesos productivos en las sociedades contemporáneas.

La industria del software no está ajena a esta demanda. La calidad es uno de los problemas que se afronta actualmente en la producción de sistemas informáticos. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, los cuales han realizado numerosas investigaciones al respecto según (Fernández Carrasco, y otros, 1995).

A pesar de la experiencia adquirida a través de los años y de haberse producido software para diferentes ramas de la economía y la sociedad, se puede decir que éste se encuentra en una crisis de la cual no ha podido salir, pues todavía existen las mismas fallas que existían antes de la creación de métodos científicos y rigurosos como la Ingeniería de Software, creada con el objetivo de llevar a cabo todo un proceso de desarrollo que permitiera la producción eficaz de sistemas informáticos. A medida que éstos se han venido desarrollando, los especialistas se han dado cuenta de que todavía hay algo que les falta, y han asumido la tarea de investigar cuáles son los motivos que influyen de manera negativa en la calidad del producto final.

Hoy día, es una prioridad del estado cubano lograr que la informática se convierta en una de las ramas más productoras de la sociedad, pues a lo largo de estos años se han venido creando nuevas empresas productoras de software dedicadas principalmente a informatizar varios sectores de la sociedad cubana de manera acelerada. Los objetivos de estas empresas son principalmente comercializar e implementar software y servicios que les permitan a las entidades organizar y gestionar adecuadamente sus recursos. Entre estas instituciones se encuentran: Desoft, Softel, el Instituto Central de Investigaciones Digitales (ICID) y además la reciente creación de la Universidad de las Ciencias Informáticas (UCI). Esto es un ejemplo de lo que se está haciendo para lograr dicho objetivo.

La UCI es ya una institución que produce software tanto para la exportación como para la informatización de toda la sociedad cubana y desde sus inicios surge la preocupación de crear software teniendo en cuenta los estándares internacionales de calidad. Se trata de que los

proyectos, a partir de que comiencen a desarrollarse, cuenten con un sistema de gestión de la calidad eficiente, que reduzca cualquiera de los problemas identificados los proyectos.

Una línea estratégica de la dirección del país es que esta universidad se convierta en una institución de referencia en el mercado internacional, por lo que su mayor interés es desarrollar sistemas informáticos con niveles adecuados de calidad, pero ¿Cómo saber cuándo un producto tiene niveles adecuados de calidad? Para el entendimiento de esta propuesta es necesario tratar de manera puntual este polémico aspecto que se resume de la manera siguiente:

Un producto software contará con un grupo de características operativas, será capaz de soportar los cambios y de adaptarse a nuevos entornos; por tanto, aspectos como la corrección, la fiabilidad, la eficiencia, la seguridad, la facilidad de uso y de mantenimiento, la reusabilidad y la portabilidad, entre otros, forman parte de lo que se considera como adecuado en cuestiones de calidad, así como el uso correcto de métricas y estándares de calidad.

Como caso de estudio se tomaron los resultados obtenidos de una auditoría *(Anexo 1)* aplicada por el Grupo de Calidad de la Facultad 3 a diferentes proyectos de la misma, tales como ONE (Oficina Nacional de Estadística), RN (Registros y Notarías), Plataforma de Comercio Electrónico, ERP y el proyecto Convenio Cuba – Venezuela, detectando una serie de dificultades:

- ✓ La ausencia del plan de aseguramiento de la calidad, que crea problemas con la planificación y hace que el software no sea entregado en el tiempo acordado y que su coste de su producción sea mayor a lo presupuestado. Además, sin este plan no se definen los procedimientos y reglas fundamentales necesarias en la creación de un producto con adecuados niveles de calidad producido por el proyecto.
- ✓ La inexistencia de métricas para la medición de los procesos que se realizan en las diferentes fases del desarrollo del software, lo que incide de manera negativa en la calidad del producto final, ya que dichos procesos no pueden ser mejorados.
- ✓ La mala captura de los requisitos que han realizado los proyectos, lo que provoca el no cumplimiento de las especificaciones que se necesitan desarrollar.
- ✓ La dificultad de encontrar todos los errores antes de entregar el software al cliente, que facilita el surgimiento de inconformidades en los acuerdos tomados por ambas partes, implicando que la universidad pueda perder crédito ante sus clientes.

- ✓ Los proyectos le dedican muy poco tiempo a las pruebas incidiendo negativamente en el buen desarrollo del producto y atentando contra la calidad del mismo.

Todas estas dificultades muestran de manera objetiva los problemas que enfrentan los proyectos productivos en la facultad 3 y que inciden negativamente en la calidad de los productos que se desarrollan en los mismos.

Específicamente en el proyecto Convenio Cuba – Venezuela, los problemas se centraron en tres aspectos fundamentales:

- ✓ La mala captura de requisitos, ya que el proceso que se realizó para la misma dificultó el trabajo, pues no se realizaban entrevistas con el cliente sino a través de un intermediario que en algunas ocasiones emitía juicios no acertados y la concepción del negocio era diferente.

- ✓ Insuficiencia a la hora de realizar las prueba y el poco tiempo que se le dedica a las mismas. Desde sus inicios se comenzó a realizar un prototipo funcional para que el cliente viera cómo quedaría la aplicación y desde la primera fase las pruebas que se comenzaron a realizar eran insuficientes y el tiempo dedicado a esto era mínimo.

- ✓ La ausencia de aplicación de métricas no permitía mejorar el proceso de desarrollo ni se podía evaluar con resultados concretos.

Al analizarse la anterior **situación problemática** se determinó que el **problema** que se enfrenta en esta investigación es: ¿Cómo contribuir al mejoramiento del proceso de Gestión de la Calidad en el proyecto Convenio Cuba - Venezuela para que el producto final sea entregado con niveles de calidad adecuados?

El proceso de Gestión de la Calidad en los proyectos de desarrollo de software constituye el **objeto de estudio** debido a que es la ciencia que estudia el fenómeno y se ha delimitado como **campo de acción** el aseguramiento de la calidad para el proyecto Convenio Cuba-Venezuela de la Facultad 3.

Este trabajo tiene como **objetivo general** para dar respuesta al problema planteado elaborar una estrategia de aseguramiento de la calidad en la segunda fase del proyecto Convenio Cuba – Venezuela de la Facultad 3.

La **hipótesis** de la investigación es que si se propone una buena estrategia de aseguramiento de la calidad para la segunda fase del proyecto Convenio Cuba-Venezuela, entonces se contribuiría a mejorar la gestión eficiente de los procesos de desarrollo de software en el proyecto, para que el producto final sea entregado con adecuados niveles de calidad.

Los **objetivos específicos:**

1. Seleccionar a partir del estudio de los procesos involucrados en el Aseguramiento de la Calidad del Software, cuáles serán los más apropiados para desarrollar una estrategia que permita al proyecto Convenio-Cuba Venezuela obtener un producto final con adecuados niveles de calidad.
2. Describir las características de la solución que se propone basada en la revisión de los planes del proyecto tales como Plan de Desarrollo de Software, Plan de Gestión de la Configuración de Software y Plan de Verificación y Validación de Software, en la creación del Plan de Aseguramiento de la Calidad y en las actividades para hacer cumplir de manera efectiva dicho plan.
3. Validar la estrategia propuesta mediante la evaluación de especialistas.

Las **tareas de la investigación**:

- ✓ Realización de un estudio del estado del arte de la temática en Cuba y el mundo.
- ✓ Identificación de las debilidades y fortalezas que posee el proceso de aseguramiento de la calidad en los proyectos productivos de la Facultad 3.
- ✓ Elaboración y proposición de un mecanismo de aseguramiento de la calidad para el proyecto Convenio Cuba-Venezuela de la Facultad 3.
- ✓ Definición de las actividades de aseguramiento de la calidad partiendo de la solución que se propone.
- ✓ Evaluación de la estrategia propuesta por especialistas.

En el Capítulo 1 se realiza un análisis del concepto de calidad de manera general y luego se analiza enfocado en la industria de software partiendo de sus antecedentes, o sea, de cómo fue surgiendo a medida que se fue desarrollando dicha industria. Se trata la calidad a nivel de la organización y de proyectos de software. Se sintetizan algunos modelos y estándares de calidad considerados como los más importantes aplicados al desarrollo de software, además de

que se realiza una descripción sobre las Revisiones Técnicas Formales y se concluye con un análisis de la gestión de calidad a diferentes niveles, todo esto para permitir desarrollar la estrategia propuesta para el proyecto Convenio Cuba-Venezuela.

En el Capítulo 2 se hace una descripción de las características de la solución propuesta dividida en tres partes fundamentales. Inicialmente se propone la realización de un proceso de revisión en el que debe estar presente el personal de Aseguramiento de la Calidad (SQA) en cada fase del proceso de desarrollo de software, centrado en los planes del proyecto. Por cada fase ya sea Inicio, Elaboración, Construcción y Transición se describen todas las revisiones que SQA debe realizar en los planes: Plan de Desarrollo de Software, Plan de Gestión de Configuración de Software y Plan de Validación y Verificación de Software. En la segunda parte del capítulo, se propone un Plan de Aseguramiento de la Calidad para el proyecto Convenio Cuba-Venezuela y finalmente se concluye con la propuesta de una serie de actividades encaminadas a cumplir satisfactoriamente dicho plan.

En el Capítulo 3 se realiza la validación de la propuesta a partir de la aplicación del criterio de especialistas, o sea, se presentan los resultados de un cuestionario después de haber sido aplicado a personas con experiencia profesional en el tema y que tienen el conocimiento necesario para emitir su criterio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

Este capítulo tiene como objetivo fundamental, tratar diferentes aspectos que sirven de soporte teórico para el desarrollo de la solución que se propone (Estrategia de Aseguramiento de la Calidad). Se hace una reseña sobre el surgimiento del concepto de la Calidad de Software, o sea, cómo fue perfeccionándose hasta llegar a su definición actual. Esto permitirá realizar un análisis de las diferentes definiciones planteadas por varios autores para determinar cuál será el concepto más apropiado y conveniente a utilizar.

Además se hace un estudio sobre la calidad a nivel de organización y de proyecto software, sobre los Modelos y Estándares de Calidad permitiendo hacer un análisis de los mismos y comparando cuál puede ser más viable y aplicable a esta solución. También se tratan las Revisiones Técnicas Formales, así como la Gestión de la Calidad de software.

1.2. Concepto de Calidad.

El término Calidad está definido por la Real Academia Española de la Lengua como: *"el conjunto de cualidades que constituyen la manera de ser de una persona o cosa"* (R.A.E. Diccionario de la lengua española 2002) y se puede encontrar hoy día en multitud de contextos. Por tanto el concepto técnico de calidad representa una forma de hacer las cosas en las que, fundamentalmente, predominan la preocupación por satisfacer al cliente y por mejorar, día a día, procesos y resultados.

El concepto actual de Calidad ha evolucionado hasta convertirse en una forma de gestión que introduce el concepto de mejora continua en cualquier organización y a todos los niveles de la misma, y que afecta a todas las personas y a todos los procesos involucrados en ella. En este trabajo se presentan algunas definiciones de personalidades e instituciones, todos considerados de relevancia en la literatura del tema que se investiga.

La International Standard Organization (ISO por sus siglas en inglés) en su norma 8402, define la calidad como: *"el conjunto de características de una entidad que le confieren la aptitud para satisfacer las necesidades establecidas o implícitas"*. Esta definición, junto con la norma ISO 9000, ha permitido su armonización a escala mundial y ha supuesto el crecimiento del impacto de la calidad en el mercado internacional.

Según Armand V. Feigenbaum (…..) es *"La composición total de las características de los productos y servicios de marketing, ingeniería, fabricación y mantenimiento, a través de los cuales los productos y los servicios cumplirán las expectativas de los clientes".* Como es de suponer, este nuevo término ha revolucionado en todos los sentidos la visión que se tenía de la calidad, ya no es simplemente el hecho de que un producto cumpla con ciertas características, sino que ahora es dirigido a lo que el cliente o consumidor desea, a lo que pueda satisfacer sus necesidades, es por eso que esta definición es mucho más completa y acertada.

Todos estos autores han tenido una influencia directa y notoria en el desarrollo del concepto actual de calidad y en la puesta a punto de estrategias y herramientas para implantarla en las empresas. Lo cierto es que no siempre se ha aplicado de la manera correcta y en muchas organizaciones empresariales este tema no ha sido llevado de la mano como realmente debiera ser, influyendo tan drásticamente, que muchas de ellas han perecido ante tal situación. El tema de la calidad debe ser tomado en serio desde etapas tempranas en la organización, pero lo más importante es que debe mantenerse como máxima prioridad si se quiere lograr una empresa sólida y con reconocimiento en el mercado internacional.

La industria del software no está exenta de estos problemas y a medida que ha venido desarrollándose son cada vez más las exigencias del mercado por un producto de calidad. La experiencia demostraba que se necesitaba definir cuándo un software cumplía requisitos de calidad y fue así como comenzaron a surgir diferentes ideas y enfoques encaminados a resolver dicho problema.

1.3. Antecedentes de la Calidad de Software. Enfoques. Definición.

Como se ha visto, se ha tratado el concepto de calidad de manera general en el epígrafe anterior. Esta definición ha venido perfeccionándose desde su surgimiento hasta nuestros días, de ahí que sea necesario realizar todo un estudio de este proceso evolutivo que posibilitará conocer cómo se ha desarrollado este concepto en relación con la Calidad de Software.

Este proceso viene enmarcado en varias etapas. La primera (siglo XIX), inicia el desarrollo del Control de la Calidad por Inspección, que se basaba en la detección de los productos que no se ajustaban a los estándares deseados, o sea, falta de uniformidad en los mismos. Durante esta fase, se consideró que la inspección era la única manera de asegurar la calidad. Según un Manual de Gestión de la Calidad Total a la Medida, del Proyecto OEA/GTZ, de 1983, “la ejecución de la práctica se orientó a tareas tales como la selección y clasificación de los productos, el rescate de productos de lotes dañados, reprocesamiento, la ejecución de mezclas

para salvar materias primas con daños leves, la toma de acciones correctivas y la búsqueda de las fuentes de no conformidad" (Proyecto, 1983).

La segunda etapa conocida como el Control Estadístico de la Calidad aparece en la década de los 30´s y "estaba enfocada al control de los procesos y a la aparición de métodos estadísticos para el mismo fin y la reducción de los niveles de inspección" (Proyecto, 1983). En esta fase se establecen procedimientos para la elaboración, el control y la difusión de informes que se basaban en la recolección de información sobre el comportamiento de los procesos, así como la aparición de los primeros manuales de calidad que formarían parte de una planificación básica para controlar la calidad.

Ya en la década de los 50´s, se determina que el control de la calidad no puede basarse solo en la inspección y métodos estadísticos que evalúen el estado del producto una vez terminado, es entonces cuando surge la tercera etapa conocida como Aseguramiento de la Calidad. Se consideraba que el aseguramiento se centraba en el conjunto de procesos, desde que se inicia, durante la producción y al final. Es por esta razón que surge la necesidad de involucrar a todos los departamentos de las organizaciones en la implantación de políticas de calidad, lo que implicaba que se gestionaran y establecieran estándares en cada elemento clave del proceso para asegurar la calidad del producto. Ya se incluyen en esta fase, términos como: costes de calidad, ingeniería de la fiabilidad y cero defectos.

La Calidad Total o Estrategia de la Calidad Total como también se le conoce, es la cuarta y última etapa en la evolución del tema que se está analizando. El objetivo de la misma es mejorar la organización, los productos y la satisfacción del cliente, o sea, el mejoramiento continuo en todas las áreas de la empresa encaminadas a satisfacer a cada consumidor. La diferencia esencial con las etapas anteriores es que no depende de las especificaciones o del uso, sino de la satisfacción del comprador, encontrándose aspectos tan variados como la satisfacción del cliente en la gestión, administración y atención personal, priorizándose así las exigencias del mercado. El papel de la dirección y la implicación de todos los trabajadores se convierten en el factor esencial del cambio.

Estrategia de la Calidad Total es el término más evolucionado dentro de las sucesivas transformaciones que se han venido desarrollando a lo largo del tiempo y que engloba ahora todo un sistema de gestión empresarial íntimamente relacionado con el concepto de Mejora Continua.

Es indudable que esta última etapa ha sido muy importante para la industria del software y cualquier organización en general, de hecho ha concebido un nuevo punto de vista, una

concepción diferente en comparación con las anteriores. Aspectos como mejora continua de la organización, satisfacción al cliente y exigencias del mercado han demostrado que necesitan ser analizados para crear un producto que tenga calidad y ya en estos momentos es imposible para cualquier empresa mantenerse en la competencia si no acata seriamente esta nueva forma de producción.

Los principios de la Calidad Total incluyen: ejemplaridad de la dirección, preocupación por la mejora continua, adhesión de todos los profesionales, cambio en la cultura de la organización para introducir y compartir los valores de la preocupación por la mejora, evaluación y planificación de la calidad, rápida circulación de la información, incorporación del punto de vista del cliente e importancia del cliente interno.

Luego de explicado el proceso de aparición de la Calidad de Software, es necesario en primer lugar conocer que los procesos de desarrollo, artefactos, gestión de proyectos, análisis y diseño, especificación de requerimientos y arquitectura, son solo algunos de los componentes que se aglomeran para conformar la Ingeniería de Software. Ahora bien, dentro de ésta, existe un subconjunto de teorías, herramientas y métodos orientados a lo que se denomina la Calidad del Software que va dirigida y aplicada a cada uno de los componentes anteriormente mencionados.

Relacionado a la Ingeniería de Software se puede adoptar la definición de la norma ISO-8402, que luego se repite en otras (por ejemplo en ISO-14598): *"La totalidad de aspectos y características de un producto o servicio que tienen que ver con su habilidad para satisfacer las necesidades declaradas o implícitas"*

La definición más completa y acorde a la solución que se propone es la siguiente:

"Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario" (Pressman, 1998)

La Calidad de Software es por lo tanto, el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. Esta es medible y varía de un sistema a otro, de un programa a otro.

Es por eso que para lograr un producto excepcional debe existir el aseguramiento de la calidad, o sea, mantener un conjunto de actividades planificadas y sistemáticas necesarias para lograr que el software satisfaga las necesidades del cliente. Además es imposible lograr producir un

producto de calidad si la organización y el proyecto de la misma no funcionan bien. Como empresa debe existir un control gerencial de excelencia, donde se puedan establecer y cumplir los objetivos trazados, así como la implantación de lineamientos que deben ser llevados a cabo para que a su vez los proyectos puedan guiarse y ser dirigidos hacia una producción exitosa.

1.4. Calidad a nivel de la organización y a nivel de los proyectos software.

La función de la calidad a nivel de la organización y a nivel de los proyectos de software es ayudar a las organizaciones de línea y de proyecto a alcanzar sus objetivos, apoyándolos en la implementación de los métodos, herramientas y la formación asociada.

La ayuda que ofrece a la Organización la Función de Calidad se basa en:

- ✓ Comprobar que los proyectos disponen de la infraestructura y medios suficientes para poder cumplir sus objetivos.
- ✓ Proporcionar información objetiva a la organización de línea y de proyecto para poder tomar decisiones en relación al grado en que el producto va a satisfacer las expectativas de los clientes.
- ✓ Prevenir problemas que puedan afectar a los objetivos del proyecto.
- ✓ Asegurar la transferencia de experiencia entre los distintos componentes de la Organización.

Hasta que no se tiene claro el papel de la Función de Calidad para la Organización no se puede hacer una implantación efectiva. Es en este momento en el que surgen naturalmente los niveles adecuados de:

- ✓ Autoridad y Responsabilidad
- ✓ Independencia
- ✓ Auditorías.

1.4.1. Calidad a nivel de organización.

Dentro de este nivel la gestión de la calidad en organizaciones de software ha seguido dos líneas que pueden ser complementarias entre sí.

Por una parte, se propone aportar para la consolidación de la implantación, en el medio productivo del país, de las propuestas que se formulen por parte de entidades internacionales de estandarización para todas las organizaciones de producción o servicios. Principalmente, se ha impuesto en la práctica los estándares ISO a través de su serie de normas ISO 9000 para la gestión de calidad. En Cuba la adopción de los estándares ISO por parte de la industria del software está en sus primeros pasos.

Por otra parte, el mundo del software ha creado otras líneas de trabajo en la gestión de la calidad del software tomando las ideas básicas de la anterior, es decir, trabajar sobre los procesos de producción de software como medio de asegurar la calidad del producto software. Así, se comenzó en el SEI (Software Engineering Institute por sus siglas en inglés) de EE.UU. proponiendo un modelo de clasificación y mejora de los procesos empleados por las organizaciones de software denominado CMM. Su trabajo se centra en el estudio y clasificación de los distintos procesos involucrados en la producción de software bajo el enfoque de una serie de niveles de madurez. Sobre este modelo pionero, se han creado nuevos modelos que suponen tanto actualizaciones y variantes por parte del propio SEI.

En la Universidad de las Ciencias Informáticas como organización se trabaja en aras de mantener la calidad del proceso empleado en la producción de software como medio muy efectivo de asegurar un buen nivel de calidad en los productos. Para esto se ha creado en la Infraestructura Productiva de la misma la Dirección de Calidad de Software, que garantiza el crecimiento continuo de una producción de software con calidad; a través de la definición de procesos siguiendo las especificaciones de metodologías, estándares y modelos de desarrollo de software, brindando asesorías, entrenamiento, métodos de medición y servicios de verificación-validación a los diferentes entidades.

La Dirección de Calidad de Software de la UCI es un centro de referencia de calidad y órgano de certificación de software, conocido y acreditado a nivel nacional. Cuenta con especialistas altamente calificados y un alto por ciento de ellos certificados internacionalmente, con un laboratorio certificado según las normas ISO/IEC 17025 y con todos los procesos definidos e institucionalizados.

1.4.2. Calidad a nivel de proyecto.

En cada proyecto de desarrollo, el aseguramiento de la calidad del software supone la aplicación de los estándares de proceso definidos en las disposiciones que, al nivel de

organización, se han establecido, bien sea como un sistema de calidad bien definido, o bien mediante una serie de procedimientos perfectamente definidos.

En cualquier caso, la medición supone, junto a las actividades de verificación y validación (básicamente, pruebas de software y actividades de revisión y auditoría), una de las técnicas principales previstas en los estándares para el control y el aseguramiento de la calidad. Desde este punto de vista, la medición puede contribuir tanto en el control de los procesos y actividades como en el de los productos, para comprender la situación de los mismos o para controlar si cumplen los requisitos pedidos o un cierto nivel de calidad.

Por otra parte, existe una multitud de métodos de medición y de productos orientados a la evaluación de la calidad del software, o alguna de sus facetas. En este caso, el proyecto se orientará al análisis comparativo de las propuestas (tanto a nivel conceptual como en proyectos del mundo real). Este estudio consistirá en evaluar cómo contribuyen o se pueden usar para el aseguramiento de la calidad en un proyecto y qué característica miden realmente y si pueden encajar dentro de los anteriormente mencionados modelos de evaluación de calidad de software y dentro de las actividades de aseguramiento de calidad.

El proyecto Convenio Cuba – Venezuela debido a sus características específicas, cuenta con un pequeño grupo de calidad que tiene el objetivo de controlar todos los procesos que se desarrollan en el mismo, o sea, asegurar la calidad del trabajo, a través de la vigilancia, prevención, comprobación y valoración sistemática a lo largo del ciclo de vida del software, velando por que el producto final cumpla con los requerimientos establecidos por el cliente. Se realizan capacitaciones para el grupo de calidad para mantenerse actualizados en estos temas y hacer uso de los procedimientos más eficientes que se desarrollan actualmente para mejorar el proceso de desarrollo. Así de manera general, este proyecto se guía por los lineamientos establecidos por su organización en este caso la UCI y hace uso del mismo de acuerdo a sus necesidades y características particulares.

Como bien se conoce, el objetivo de un proyecto de software es hacer un buen producto, pero su proceso es bastante difícil y no siempre el resultado obtenido es el que se espera, ya sea por la existencia de malas prácticas o por el mal funcionamiento del proyecto de manera general. Muchas veces no son capaces de analizar con detenimiento las características que debe tener el producto para cumplir con lo estipulado por el cliente, de ahí que muchos son los factores que influyen en el proceso de creación de un software que tendrá la calidad requerida.

1.5. Factores relacionados con la calidad del proceso software.

Para obtener un producto con un alto nivel de calidad, se necesita que en su proceso de desarrollo se analicen una serie de factores que contribuyan de manera positiva en el mismo y que determinen el grado de aceptación que el cliente necesita, o sea, que tenga la máxima calidad.

Según Pressman estos factores se pueden clasificar en dos grandes grupos y son conocidos como Factores de calidad de McCall:

- ✓ Factores que pueden ser medidos directamente (defectos por puntos de función).
- ✓ Factores que solo pueden ser medidos indirectamente (facilidad de uso o mantenimiento).

Se centran en tres aspectos fundamentales de un producto software según McCall:

Operaciones del producto: características operativas

Corrección. ¿Hace lo que se quiere?

- ✓ El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.

Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?

- ✓ El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.

Eficiencia. ¿Se ejecutará en el hardware lo mejor que pueda?

- ✓ La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

Seguridad (Integridad). ¿Es seguro?

- ✓ El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.

Facilidad de Uso. ¿Está diseñado para ser usado?

- ✓ El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

Revisión del producto: capacidad de soportar los cambios

Facilidad de Mantenimiento. ¿Se puede corregir y localizar los fallos?

- ✓ El esfuerzo requerido para localizar y reparar errores.

Flexibilidad. ¿Se puede cambiar?

- ✓ El esfuerzo requerido para modificar una aplicación en funcionamiento.

Facilidad de Prueba. ¿Se puedo probar?

- ✓ El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

Transición del producto: adaptabilidad a nuevos entornos

Portabilidad. ¿Se podrá usar en otra máquina?

- ✓ El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.

Reusabilidad. ¿Se podrá reutilizar alguna parte del software?

- ✓ Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.

Interoperabilidad. ¿Se podrá hacerlo interactuar con otro sistema?

- ✓ El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas Informáticos.

Estos factores que inciden positivamente sobre cualquier producto hacen que el mismo tenga una gran ventaja en el mercado pues según Deming “La calidad aumenta la productividad”.

Todo esto conlleva a una reacción en cadena que permite producir más, vender más y complacer cada vez mejor las necesidades del cliente.

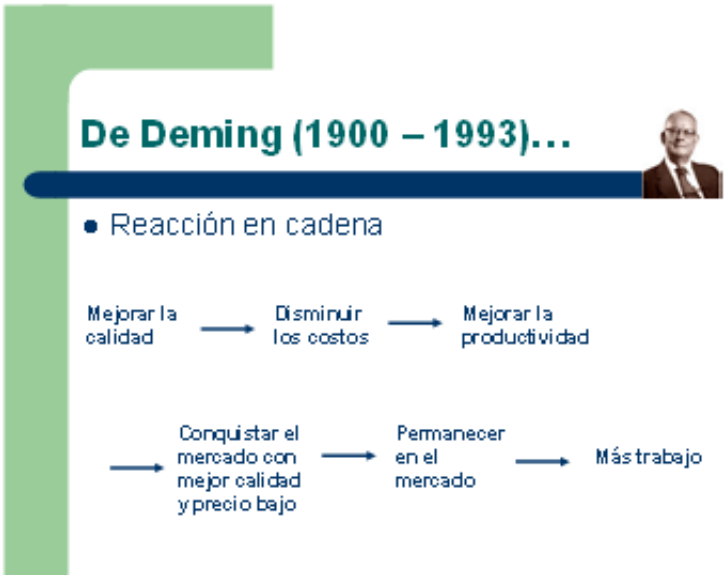

Figura 1 Reacción en cadena de Deming.

Según Deming (Figura 1) una mejora de la calidad contribuye a disminuir los costos ya que hay menos desperdicios, menor reproceso, menos errores y retrasos, se aprovecha mejor el tiempo de máquinas y solo se emplean los insumos necesarios. Esto a su vez mejora la productividad pues las horas-hombre y las horas-máquina no se malgastan, se aprovechan mejor y los recursos renovables no se deterioran permitiendo así que se conquiste el mercado con mejor calidad y precio bajo ya que se planifica y produce de acuerdo a las preferencias y requerimientos del cliente y se realiza una mejora continua en ese sentido. Esto hace que se tenga una buena permanencia en el mercado, los clientes satisfechos siempre volverán a comprar y recomendarán lo mismo a otros clientes, lo que esto hará que aumente el trabajo pues el mercado ha sido conquistado y el desarrollador está contento con lo que ha producido.

Esta teoría de Deming demuestra una vez más la importancia que tiene la calidad para obtener mayores niveles de producción, la Reacción en Cadena como él mismo la llama es un ejemplo eficaz de lo que se puede realizar para mejorar como empresa, de todo lo que se puede hacer para satisfacer las necesidades del mercado e implantar a nivel mundial un modelo de organización con una alta productividad de calidad.

Lo más importante para lograr estos objetivos es regirse siempre por estándares y modelos mundialmente probados y con su correcta aplicación se llegará a dominar a la competencia internacional. El uso de los mismos le dará al cliente una mayor seguridad para realizar su inversión y contribuirá a que otros confíen en dicha empresa, haciendo de la misma el centro de producción de software mayormente solicitado. De ahí que los modelos y estándares deben ser prioridad número uno para todos las organizaciones desarrolladoras de sistemas.

1.6. Modelos y estándares de calidad.

Los modelos de calidad son un conjunto de buenas prácticas para el ciclo de vida del software, son útiles para discutir, planificar y obtener altos índices de calidad en el desarrollo del mismo, además que tienen en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores y usuarios finales. Estos dicen (qué) hacer y no (cómo) hacerlo, pues depende de la metodología que se use y de los objetivos del negocio.

Los Estándares de Calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería de Software. Además suministran los medios para que todos los procesos se realicen de la misma manera y definen el rango en el que resulta aceptable el nivel de calidad que se alcanza en un determinado proceso, o sea, el nivel mínimo y máximo que puede alcanzar un indicador.

Estos modelos y estándares convierten a la calidad en algo concreto, de ahí su ventaja principal, ya que permiten definir, medir y planificar. Pero desafortunadamente aún no ha sido demostrada la validez absoluta de ninguno. Esto origina que existan múltiples Modelos y Estándares de Calidad. Cada uno de ellos es una herramienta que guía a las Organizaciones a la Mejora Continua y la Competitividad. (Díaz Pérez, y otros, Junio de 2007).

1.6.1. Modelo Integrado de Capacidad de Madurez (CMMI, siglas en inglés).

CMMI nació integrando tres modelos diferentes, con representaciones diferentes, pues se desarrolló con el objetivo de facilitar y simplificar los mismos de forma simultánea: CMM-SW (CMM for Software) que tenía una representación escalonada, SE-CMM (Systems Engineering Capability Maturity Model) con una representación continua y IPD-CMM (Integrated Product Development) que era un modelo mixto. Este nuevo modelo tenía dos representaciones, continua y escalonada. La visión continua de una organización mostrará la representación de nivel de capacidad de cada una de las áreas de proceso del modelo y la visión escalonada definirá a la organización dándole en su conjunto un nivel de madurez del 1 al 5.

CMMI es un modelo que facilita lineamientos para las empresas, organizaciones o áreas que deseen una mejora continua y efectiva en sus procesos de desarrollo de software. No propone crear ni establecer los procesos para el desarrollo de software, ni la manera de como producir o mantener un sistema, si no que sugiere los lineamientos y características que deben tener estos procesos.

Este modelo es un enfoque de mejora de procesos que proporciona a las organizaciones los elementos esenciales de la eficacia de los procesos. Se puede utilizar para guiar el proceso de mejora a través de un proyecto, una división, o de toda una organización. Además ayuda a integrar funciones tradicionalmente separadas de la organización, establecer los objetivos y las prioridades de la mejora del proceso, facilitar la orientación de procesos de calidad, y proporcionar un punto de referencia para la evaluación de los procesos actuales. (SEI, 2008)

Está organizado por niveles de madurez y áreas de proceso. Los 5 niveles definidos en CMMI para medir la capacidad de los procesos son:

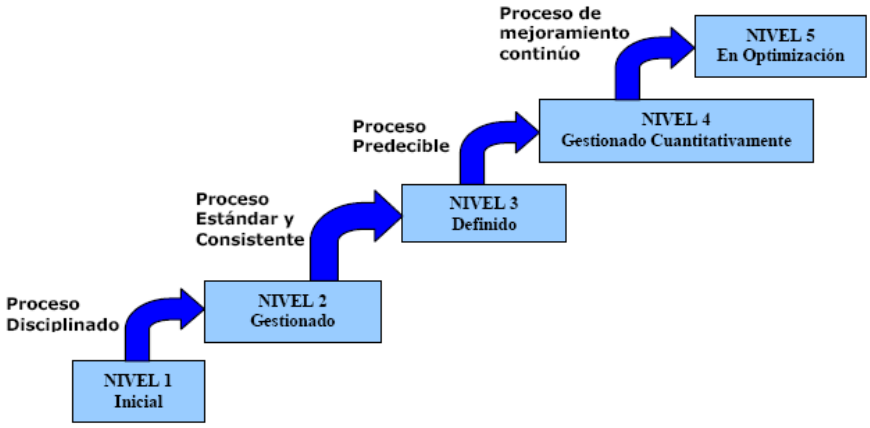

Figura 2 Niveles de CMMI.

- ✓ Nivel 1, Inicial: Describe organizaciones sin grado de madurez, en las que los proyectos salen adelante por el esfuerzo de las personas implicadas.
- ✓ Nivel 2, Gestionado: Se establecen procesos de gestión que permitan la repetición de buenas prácticas en los proyectos.
- ✓ Nivel 3, Definido: Implica la definición de procesos para toda la organización ya sea, formación, interrelación con otros grupos, ingeniería y equipo de mejora continua, además supone la extensión de los mismos a toda la organización y toda la actividad.
- ✓ Nivel 4, Cuantitativamente Gestionado: Mide y analiza el rendimiento de los procesos y proyectos de la organización.
- ✓ Nivel 5, Optimizado: Mejora continua, con la participación de todos los componentes de la organización, define objetivos cuantitativos de mejora, la implantación de los cambios necesarios y su impacto en la consecución de dichos objetivos. (Sánchez, 2005)

CMMI identifica 25 áreas de procesos (22 en la versión que no integra IPD). Vistas desde la representación continua del modelo, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte a las otras categorías.

Vistas desde la representación escalonada, se clasifican en los 5 niveles de madurez. Al nivel de madurez 2 pertenecen las áreas de proceso cuyos objetivos debe lograr la organización para alcanzarlo, ídem con el 3, 4 y 5.

Tabla 1. 1 Áreas de procesos de CMMI

Área de proceso	Categoría	N. Madurez.
Análisis y resolución de problemas.	Soporte	5
Gestión de la configuración.	Soporte	2
Análisis y resolución de decisiones.	Soporte	3
Gestión integral de proyecto.	G. Proyectos	3
Gestión integral de proveedores.	G. Proyectos	3
Gestión de equipos.	G. Proyectos	3
Medición y análisis.	Soporte	2
Entorno organizativo para integración.	Soporte	3
Innovación y desarrollo.	G. Procesos	5
Definición de procesos.	G. Procesos	3
Procesos orientados a la organización.	G. Procesos	3
Rendimiento de los procesos de la organización.	G. Procesos	4
Formación.	G. Procesos	3
Integración de producto.	Ingeniería	3
Monitorización y control de proyecto.	G. Proyecto	2
Planificación de proyecto.	G. Proyecto	2
Gestión calidad procesos y productos.	Soporte	2
Gestión cuantitativa de proyectos.	G. Proyectos	4
Desarrollo de requisitos.	Ingeniería	3
Gestión de requisitos.	Ingeniería	2
Gestión de riesgos.	G. Proyectos	3
Gestión y acuerdo con proveedores.	G. Proyectos	2
Solución técnica.	Ingeniería	3
Validación.	Ingeniería	3
Verificación.	Ingeniería	3

CMMI realiza un gran aporte en las mejoras de desarrollo de un proceso software:

- ✓ Desarrolla un marco de actuación para permitir el crecimiento de otras disciplinas.
- ✓ Nuevo énfasis sobre el producto así como sobre los procesos, incluyendo las interacciones con el cliente.
- ✓ Mayor importancia desde las fases iniciales, del análisis y la medición de los procesos empresariales.
- ✓ Cobertura de servicios, así como de sistemas.
- ✓ Especial énfasis sobre la capacidad de los procesos y madurez de la organización en su conjunto (no exclusivamente en el área de ingeniería del software).
- ✓ Mejor cobertura de la gestión de ingeniería integrada.
- ✓ Énfasis sobre las mejoras medibles y cuantificables para alcanzar los objetivos del negocio empresarial.
- ✓ Existe un nuevo enfoque de la formación. La educación y el entrenamiento adecuado para la mejora de la eficacia y de la eficiencia.
- ✓ Favorece el establecimiento de un ambiente adecuado para la gestión de los cambios dentro de la organización.
- ✓ Proporciona compatibilidad con los principios, requisitos y recomendaciones de la norma ISO 9000:2000.
- ✓ Sienta las bases para que las organizaciones del sector de desarrollo del software se encaminen hacia el ciclo de la mejora continua.

1.6.2. Normas ISO para la Calidad de Software.

El término ISO 9000 se refiere a una serie de normas universales que define un sistema de "Garantía de Calidad" desarrollado por la Organización Internacional de Normalización (ISO) y adoptado por 90 países en todo el mundo. Su objetivo es promover el intercambio de productos y servicios en todo el mundo y fomentar la cooperación mundial en las áreas intelectual, científica, tecnológica y económica. Las series de ISO 9000 son un grupo de 5 individuales, pero relacionadas, estándares internacionales de administración de la calidad y aseguramiento de calidad.

Estas son:

- ✓ ISO 9000: Cumple el papel de eje distribuidor del sistema. Define la filosofía general de las normas, los distintos tipos, niveles y pautas para la aplicación de las distintas normas. Describe los fundamentos de los sistemas de gestión de la calidad y contiene la terminología más utilizada en las normas de esta serie. Es decir que si alguien necesita conocer qué se entiende por "sistema de gestión de la calidad", "no conformidad", "producto", por ejemplo, debe referirse a esta norma.
- ✓ ISO 9001: Se aplica cuando la empresa debe responsabilizarse en las etapas del ciclo, es decir: diseño, desarrollo y elaboración.
- ✓ ISO 9003: Cubre las obligaciones de aseguramiento de calidad en las áreas de control final y pruebas. Es de limitada aplicación por lo que existen planes para su eliminación.
- ✓ ISO 9004-1/ ISO 9004-2: Establecen condiciones y pautas para guiar a las empresas en la implementación de su propio sistema de aseguramiento de calidad. Su desarrollo no es válido para certificación o registro.ISO. (Tecnomaestros, 2008)
- ✓ ISO/IEC 9126: Esta establecerá las condiciones para definir los objetivos de la calidad y las métricas.

1.6.3. Determinación de Capacidad de Mejora para Procesos Software (SPICE, siglas en inglés).

SPICE es el estándar internacional para la evaluación y mejora de procesos de software, ya que es aplicable a cualquier organización o empresa que quiera mejorar la capacidad de cualquiera de sus procesos. Es independiente de la organización, del ciclo de vida, metodología y tecnología. Realmente es un marco para métodos de evaluación, no un método o modelo en sí. Está alineado con el estándar ISO 12207.

Componentes de SPICE:

- ✓ Conceptos y Guía de Introducción.
- ✓ Modelo de Referencia para Procesos y Capacidad.
- ✓ Realización de una Evaluación.
- ✓ Guía de evaluación.

- ✓ Modelo de Evaluación y Guía de uso.
- ✓ Guía de Calificación de Evaluadores.
- ✓ Guía de Uso para la Mejora de Procesos.
- ✓ Guía para Determinación de Capacidad de Proveedores.
- ✓ Vocabulario.

El modelo de referencia de SPICE describe los procesos que una organización puede realizar para comprar, suministrar, desarrollar, operar, mantener y soportar el software, así como los atributos que caracterizan la capacidad de estos procesos. Está compuesto por 2 dimensiones: Procesos y Capacidad. La primera dimensión contiene los procesos que se van a evaluar, o sea, correspondientes al ciclo de vida del software y está dividido en categorías dependiendo del tipo de actividad al cual se aplican:

Cliente-Proveedor (CUS)

Esta categoría está formada por procesos relacionados directamente con el cliente, soportan el desarrollo y la transición del software al cliente y permiten la correcta operación y uso del producto o servicio. Los principales procesos son:

Adquisición de productos software y/o servicios, Establecimiento de contratos, Identificar las necesidades del cliente, Realizar auditorías y revisiones conjuntas, Entrega e instalación del software, Mantenimiento del software, Proporcionar servicio al cliente, Valorar la satisfacción del cliente.

Ingeniería (ENG)

Está compuesta por procesos que directamente implementa, especifica o mantienen el producto, su relación con el sistema y su documentación. Los procesos que abarcan son:

Análisis y diseño de requerimientos del sistema, Análisis de requerimientos del software, Diseño del software, Construcción del software, Integración y pruebas del software, Integración y pruebas del sistema, Mantenimiento del software y del sistema.

Soporte (SUP)

En esta los procesos dan soporte a cualquier otro ya sean:

Documentación, Gestión de la configuración del software, Garantía de calidad, Resolución de problemas, Realizar revisiones conjuntas.

Gestión (MAN)

Está formada por procesos utilizados en la gestión de cualquier tipo de proyecto o proceso en el ciclo de vida del sistema, ya sea:

Gestionar el proceso, Gestionar el proyecto, Gestionar la calidad, Gestionar los riesgos.

Organización (ORG).

En esta categoría se analizan los procesos que establecen los objetivos de negocio de la empresa.

Alineamiento de la organización, Establecimiento del proceso, Evaluación del proceso, Mejora del proceso, Gestión de recursos humanos, Infraestructura, Reutilización.

La dimensión de Capacidad define una escala de medida para determinar la capacidad de cualquier proceso y consta con 6 niveles.

- ✓ Nivel 0, Incompleto: Es un fracaso general el tratar de utilizar las prácticas bases a los procesos. Ya que no es fácil identificar las salidas de los procesos o el trabajo de los productos, o sea, no hay evidencia del logro de los resultados esperados.
- ✓ Nivel 1, Realizado: Las prácticas de los procesos son ejecutados generalmente. La ejecución de las prácticas dependerá del conocimiento y esfuerzo personal. Se identifica algunos procesos. El proceso es comprendido y los productos son realizados correctamente.
- ✓ Nivel 2, Gestionado: La ejecución de las prácticas bases en los procesos son planificadas y seguidas. La primera distinción entre el nivel 1 y el 2 es que la ejecución de los procesos está planificada y administrada y progresan hacia un proceso bien definido.
- ✓ Nivel 3, Establecido: Las prácticas bases son ejecutadas de acuerdo a una versión adaptada del estándar, procesos aprobados bien definidos y documentados.
- ✓ Nivel 4, Previsible: Mediciones detalladas de rendimiento o ejecución son alcanzadas y evaluadas. Es conocido el rendimiento de los procesos y es posible su predicción. La calidad de las prácticas es cuantitativamente conocida.

- ✓ Nivel 5, Optimizado: Son establecidos en forma cuantitativa procesos y metas eficientes, basados en los objetivos de la organización. En forma continua los procesos se van mejorando mediante la retroalimentación obtenida por los resultados de procesos definidos y por ideas pilotos y tecnologías novedosas. (Díaz Pérez, y otros, Junio de 2007)

Ventajas

- ✓ Ofrece una base para una detallada evaluación del estado actual del proceso de una organización.
- ✓ Por su gran nivel de descomposición de los procesos e indicadores, proporciona evaluaciones objetivas y con resultados repetibles, especialmente cuando es realizada por evaluadores entrenados y calificados.
- ✓ Como es un estándar internacional se pueden realizar comparaciones a nivel mundial entre evaluaciones en contextos similares.

Desventajas

- ✓ Se requiere de un gran esfuerzo para realizar las evaluaciones.
- ✓ No es práctico ni fácil de aplicar.
- ✓ Tiene solamente lineamientos para un mecanismo de evaluación.

Por tanto la calidad de todos los componentes integrados en el proceso de desarrollo del software no mejora necesariamente por el simple hecho de adoptar un estándar, pues es necesario que el proceso de adopción conlleve una gestión del cambio adecuada. Es necesario tener un estándar como objetivo y referencia del proceso de desarrollo del software y el modelo seleccionado no es tan importante como el compromiso de mejora. Aun así, queda clara la necesidad de la aplicación de los métodos y estándares de calidad.

Si bien es necesario regirse y adherirse a un estándar mundialmente definido, es primordial el uso de aspectos medibles que proveerán a un equipo de desarrollo de la información requerida para convertir el proceso en un producto final con calidad, así las mediciones demostrarán desde un principio la factibilidad y el control de los recursos utilizados en la creación del software. A esto se le conoce como métricas y son de vital importancia ya que el uso de las mismas provee los datos de las características del producto de manera cuantitativa.

1.7. Métricas de Software.

El área de las mediciones de software, es una de las áreas en la Ingeniería de Software donde se ha investigado desde hace más de 30 años, se conoce también como métricas de software. Existe una confusión en la utilización de los términos métricas de software y mediciones de software, sin embargo, en la literatura los términos: métrica, medida o medición son usados como sinónimos (Zuse, 1995).

Varios investigadores han tratado de desarrollar una única métrica que proporcione una medida más completa para evaluar la complejidad de un software. Todas las que se han propuesto tienen un punto de vista diferente, y aunque se sabe que es real la necesidad de medir y controlar la complejidad del software, es muy difícil tener un solo valor de estas métricas de calidad. Pese a estas dificultades, se conoce que las métricas son esenciales y más si se quiere obtener un producto de software con calidad.

Las métricas de software incluyen varias actividades como son:

- ✓ Estimación de costo y el esfuerzo.
- ✓ Medición de la productividad.
- ✓ Acumulación de datos.
- ✓ Realización de modelos y mediciones de la calidad.
- ✓ Elaboración de modelos de seguridad.
- ✓ Evaluación y modelos de desempeño.
- ✓ Valoración de las capacidades y de la madurez.
- ✓ Administración por métricas.
- ✓ Evaluación del método y herramientas.

Debido a que la aplicación de las métricas tiene la ventaja de caracterizar, evaluar, predecir y mejorar el proceso, el producto y el entorno de la empresa es necesario dedicarle la mayor atención. La medición es parte de algo más amplio, es un enfoque de la organización hacia la mejora, es una actividad que requiere esfuerzo y preparación. Pero a pesar de los beneficios que pueda brindar la aplicación de las métricas no indican que todo esté bien, pues las mismas

no son absolutas, simplemente proporcionan comprensión del proceso software y no se puede identificar, explicar o predecir todo.

Una vez aplicadas las mediciones no se debe estar conforme en el resultado que se obtiene a lo largo del desarrollo del software, para eso se necesita de una actividad de revisión y control sistemático y minucioso conocida como Revisión Técnica Formal.

1.8. Revisiones técnicas formales (RTF).

Como todo producto, un software también necesita ser controlado y revisado minuciosamente para evitar que sea entregado con defectos. Las revisiones son un estudio realizado por varias personas para indicar las mejoras en el producto, indicar las partes que no es necesario mejorar y conseguir un trabajo técnico más homogéneo. Las revisiones técnicas deben centrarse solamente en el producto para detectar si este hace lo que realmente debe hacer y una revisión formal proporciona una información fiable sobre cuestiones técnicas, puesto que las informales se realizan de manera constante.

Las RTF son una actividad colectiva de garantía de la calidad del software lo que permite ampliar la mirada sobre lo que se revisa. Estas agrupan en sí diversos mecanismos de revisión y permiten establecer un marco común para la definición de distintas etapas de revisión en el ciclo de vida del software, este mecanismo no sólo está pensado para las etapas tempranas del ciclo de vida sino que también puede -y debe- ser utilizado en períodos como el de prueba y el de mantenimiento.

La orientación de la RTF debe ser muy específica, en el sentido que cada RTF se debe centrar en una parte muy bien delimitada del software total. Por ejemplo, en lugar de intentar revisar un diseño completo, se hacen inspecciones para cada módulo o pequeño grupo de módulos. Al limitar el centro de atención de la RTF la probabilidad de descubrir errores es mayor. Las revisiones técnicas pueden llevarse a cabo después del desarrollo de las especificaciones, del análisis y diseño y del Test de prueba. (Pressman, 2005)

Las revisiones se basan en verificación y validación, esto implica la tarea de asegurar que el sistema está conforme a especificaciones y acorde a las necesidades del cliente. Cuando se verifica, se hace contra especificaciones del software, o sea, determinar si se ha construido de forma correcta. Cuando se valida, es contra requisitos de usuario, aquí se revisa si se ha construido el producto correcto.

Las técnicas de revisión pueden ser estáticas, aquí se analizan y chequean los documentos de requisitos, los diagramas de diseño, código fuente, entre otros y también pueden ser dinámicas, que son pruebas que se aplican sobre implementación real (solo pueden usarse cuando ya se tiene código ejecutable).

Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida, independientemente de cualquier formato que se elija, dicha reunión debe acogerse a las siguientes restricciones:

- ✓ Deben convocarse normalmente entre 3 y 5 personas.
- ✓ Se debe preparar por adelantado, pero sin que requiera más de 2 horas de trabajo a cada persona.
- ✓ La duración de la reunión debe ser menor de 2 horas.

Objetivos de las RTF:

- ✓ Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.
- ✓ Verificar que el software bajo revisión alcanza sus requisitos.
- ✓ Garantizar que se siguen los estándares definidos.
- ✓ Conseguir un desarrollo uniforme.
- ✓ Facilitar la gestión de los proyectos, o sea, que sean más manejables.
- ✓ Encontrar errores, no corregirlos.

Como todo procedimiento, siempre se necesita de pasos específicos para llevar a cabo ciertas actividades. En primer lugar se seleccionan los revisores de forma que se cubran todos los ámbitos (futuro mantenimiento, ajuste a los estándares de la organización, usuarios y corrección y calidad del producto).

A partir de aquí, los pasos que se siguen son:

1. El productor del software solicita una revisión al jefe del proyecto.
2. El jefe de proyecto contacta con un jefe de revisión que:

- ✓ Evalúa la disponibilidad del producto, genera copias del material del producto y las distribuye a dos o tres revisores.

3. Los revisores y el jefe de revisión revisan el producto (una o dos horas). El jefe de revisión elabora una agenda para la reunión.

4. Se reúnen el jefe de revisión, los revisores (uno actuará como secretario) y el productor.

- ✓ La RTF comienza con una explicación de la agenda (jefe de revisión) y una breve introducción (productor).

- ✓ Posteriormente se discuten las "pegas" que cada revisor ha encontrado. El secretario tomará las notas oportunas.

- ✓ La duración de la reunión debe ser inferior a dos horas.

5. Al final decidirán si:

- ✓ Aceptan el producto (no se hacen modificaciones).

- ✓ Rechazan el producto (existen errores graves).

- ✓ Aceptan el producto modificándolo (corregir errores).

6. Todos los revisores firman la conformidad de los resultados de la revisión.

Ventajas:

- ✓ Más efectivo que las pruebas para encontrar errores.

- Encuentra la causa del error en lugar del síntoma.

- ✓ Programadores saben que sus programas serán revisados.

- Programas más fáciles de entender y programadores más cuidadosos.

- ✓ Anula el efecto de "puntos ciegos" (programador pasa reiteradamente sobre el error sin verlo).

- ✓ Es posible imponer estándares de codificación con facilidad.

- ✓ Explicar el código hace que el programador lo entienda cada vez mejor (los profesores saben esto).

- ✓ Reduce dramáticamente tiempo de pruebas.
- ✓ Sirven para promover la seguridad y la continuidad, ya que varias personas se familiarizarían con partes del software que de otro modo no hubieran visto nunca.
- ✓ Sirven como campo de entrenamiento, permitiendo que los ingenieros más jóvenes, puedan observar los diferentes enfoques de análisis, diseño e implementación del software.

Desventajas:

- ✓ Problemas de personalidad
- ✓ Persona con buenas ideas no se expresa.
- ✓ Persona con malas ideas se expresa mucho.
- ✓ Algunas personas odian discutir o estar en desacuerdo.
- ✓ Fácil de perderse en cosas triviales (punto y coma).
- ✓ Es agotador (pierde efectividad después de un par de horas o menos).
- ✓ Dificultades para convencer a gerentes de beneficios económicos. (Pressman, 2005).

Para que se realice una revisión con calidad hay que establecer una serie de directrices que conducirán a los revisores para llegar a un consenso final. Es por eso que: se debe revisar el producto y no al productor, o sea, se deben destacar los errores educadamente y con un tono constructivo. Fijar una agenda y mantenerla es otro punto importante pues es conveniente seguir un plan de trabajo concreto y sin divagaciones.

Está establecido que una revisión no es una sesión de resolución de problemas, es por eso que se deben enunciar áreas de problemas pero no intentar resolverlos. El tomar notas por escrito puede ayudar que las declaraciones o la asignación de prioridades puedan ser comprobadas por el resto de los revisores.

Otras directrices que influyen en la realización de una buena revisión técnica formal son: desarrollar una lista de comprobación para cada producto que haya de ser revisado, disponer recursos y una agenda para las RTF, llevar a cabo un buen entrenamiento de todos los revisores, así como, repasar las revisiones que se van realizando, para descubrir problemas en el propio proceso de revisión. (Pressman, 2005)

Después de realizado un completo chequeo se obtiene la Lista de sucesos de revisión que identifica las áreas problemáticas dentro del producto y sirve como guía para la corrección de errores por parte del productor. También se obtiene el Informe sumarial de revisión que es donde se recoge el producto que se ha revisado, los revisores y los problemas detectados, así como las conclusiones finales.

De este modo se puede afirmar que un software en el que se controle la calidad no puede dejar de lado un proceso de revisión formal que garantice de manera eficaz, la detección de cualquier problema que pueda afectar la integridad y calidad del producto. Además se debe dejar claro que para poder tener un resultado positivo en esta actividad todo dependerá de las personas que la realicen y de cómo será dirigida.

Son varias las acciones encaminadas a obtener un sistema robusto y que cumpla con las exigencias y necesidades del cliente, por lo que la aplicación de estándares, las métricas, las revisiones, el control, el aseguramiento y la planificación siempre van a ser necesarios para cumplir con dichos objetivos y todos dirigidos a desarrollar un producto de calidad. Por tanto, para que esto pueda cumplirse de manera satisfactoria se necesita de un proceso de Gestión de la Calidad que permita tener una visión general para manejar, medir y mejorar los procesos internos de las diferentes áreas de la empresa.

1.9. Gestión de la Calidad del software.

La Gestión de la calidad según la (ISO 9000) es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación, el control, el aseguramiento (garantía) y la mejora de la calidad, en el marco del sistema de calidad y que se aplica a nivel de empresa. Se centra en dos niveles de trabajo:

Primero: El nivel de entidad u organización, donde se trata de crear y gestionar una infraestructura que fomente la Calidad de los productos de software mediante la adecuación y mejora de las actividades y procesos involucrados en su producción e, incluso, en su comercialización y en la interacción con los usuarios / clientes.

Segundo: El nivel de proyecto, donde los estándares que la infraestructura organizativa prevé para las distintas actividades de desarrollo y mantenimiento de software. Estos estándares deben ser adaptados a las características concretas del proyecto y de su entorno para ser aplicados en proyectos del mundo real.

1.9.1. Planificación de la Calidad de Software.

De acuerdo a la bibliografía consultada es la parte de la Gestión de la Calidad que se encarga de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado. En relación a la planeación de la calidad la ISO 9000:2000 establece que es la parte de la gestión de la calidad que se enfoca para establecer objetivos de calidad y especificar procesos operacionales necesarios y recursos relacionados para cumplir dichos objetivos.

La norma ISO/IEC 90003:2004 plantea que:

"La planificación de la calidad facilita el modo de adaptar la planificación del sistema de gestión de la calidad a un proyecto específico, producto o contrato. La planificación de la calidad puede incluir referencias genéricas y/o proyecto / producto / contrato específico de procedimientos, como apropiados. La planificación de la calidad debería ser revisada de nuevo junto con el progreso del diseño y desarrollo, y los elementos, en cada fase, deberían ser completamente definidos al comienzo de dicha fase"

Los objetivos fundamentales son:

- ✓ Establecer un sistema de gestión de calidad
- ✓ Documentar un sistema de gestión de calidad.
- ✓ Implementar un sistema de gestión de calidad.
- ✓ Mantener un sistema de gestión de calidad.
- ✓ Continuamente mejorar la efectividad de un sistema de gestión de calidad.
    - Identificar los procesos.
    - Identificar aplicación de procesos.
    - Determinar la secuencia de procesos.
    - Determinar la interacción de procesos para determinar criterios y métodos.
    - Asegurar la disponibilidad de recursos y la información para monitorear, medir y analizar procesos.
- ✓ Gestionar procesos.

- ✓ Asegurar control de subcontratistas.
- ✓ Identificar control de procesos subcontratados.

Los aspectos a considerar en la Planificación de la Calidad de Software son: Modelos/Estándares de Calidad de Software a utilizar, Costos de la Calidad de Software, Recursos humanos y materiales necesarios, etc. El Plan de Aseguramiento de la Calidad de Software definido por la norma IEEE 730 de 1984 establece los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad.

En la Planificación de la Calidad de Software se debe determinar: (1) Rol de la Planificación, (2) Requerimientos de la Calidad de Software, (3) Preparación de un Plan de Calidad de Software, (4) Implementación de un Plan de Calidad de Software y (5) Preparar un Manual de Calidad. Dicho plan es específico para cada proyecto ya que debe seguir las directrices del Manual de Calidad y las normas de los clientes.

Según (Scalone, 2006), la Planificación de la Calidad de Software a nivel de proyectos debería considerar lo siguiente:

4. Inclusión de los planes de desarrollo.
5. Los requisitos de calidad relacionados con los productos y/o procesos.
6. Los sistemas de gestión de calidad adaptando y/o identificando los procesos e instrucciones específicos, apropiados para el ámbito del manual de calidad y algunas exclusiones expuestas.
7. Los procesos de proyectos específicos e instrucciones, tales como, especificación de pruebas del software detallando los planes, diseños, casos de pruebas y procesos para la unidad, integración, sistemas y pruebas de aceptación.
8. Los métodos, modelos, herramientas, convenios de lenguajes de programación, bibliotecas, marcos de trabajo, y otros componentes reutilizables para ser usados en los proyectos.
9. Los criterios para el comienzo y el final de cada fase o etapa del proyecto.
10. Los tipos de análisis y otras verificaciones y actividades de validación para ser llevadas a cabo.

Según Humphrey (1989) un plan de calidad puede tener la siguiente estructura:

1. Introducción al producto: una descripción del producto, su objetivo en el mercado y expectativas de calidad del producto.
2. Planes del producto: fechas críticas de acuerdo a la liberación del producto y responsabilidades del producto respecto de su distribución y servicio.
3. Descripciones del proceso: Procesos de desarrollo y servicios que serían utilizados en el desarrollo y en la administración.
4. Objetivos de Calidad: objetivos y planes de calidad del producto, los cuales incluyen la identificación de los atributos de calidad del producto.
5. Manejo del riesgo: principales riesgos que pueden afectar la calidad del producto.

Esta información es fundamentalmente recogida en un plan de calidad. El Plan de Calidad define los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad.

1.9.2. Control de la Calidad de Software.

Según Pressman el control de calidad es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados. Este incluye un bucle de realimentación (feedback) del proceso que creó el producto. La combinación de medición y realimentación permite afinar el proceso cuando los productos de trabajo creados fallan al cumplir sus especificaciones. Este enfoque ve el control de calidad como parte del proceso de fabricación.

Según la norma ISO 9000:2000, es la parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad. El control de la calidad del software, es la técnica y actividad de carácter operativo, utilizado para satisfacer los requisitos relativos a la calidad, centrado en dos objetivos fundamentales: mantener bajo control un proceso y eliminar las causas de los defectos en las diferentes fases del ciclo de vida. Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la Calidad del Software es la “Prueba de Software”.

Se llama prueba de software, según Pressman (5) al proceso en el que se ejecuta un sistema con el objetivo de detectar fallos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades. La prueba exitosa es aquella que descubre defectos. El “caso de prueba bueno” es aquel que tiene alta probabilidad de detectar

un defecto aún no descubierto. El “caso de prueba exitoso” es aquel que detecta un defecto aún no descubierto.

Una estrategia tradicional de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. Cuando se construye un software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Estas las realiza el usuario final en lugar del responsable del desarrollo de sistema.

El diseño de casos de prueba para el software o para otros productos de ingeniería puede requerir tanto esfuerzo como el propio diseño inicial del producto. Cualquier producto de ingeniería puede probarse de una de estas 2 formas: (1) prueba de caja negra y (2) prueba de caja blanca.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para este tipo de prueba, se deben definir todos los caminos lógicos y desarrollar casos de prueba que ejerciten la lógica del programa.

De manera general todas estas son actividades a implementar para evaluar la calidad de los productos desarrollados.

1.9.3. Aseguramiento de la Calidad de Software.

El aseguramiento de la calidad debe proveer la gestión con una adecuada visibilidad en el proceso de software que está siendo usado y en los productos que están siendo construidos Este incluye las revisiones y las auditorías de los productos y de las actividades para verificar que ellas cumplen con los procedimientos y estándares y suministra los resultados al jefe de proyecto y otros administradores. (Schulmeyer, 1997)

Según Pressman *"La garantía de calidad del software (SQA: "Software Quality Assurance") es un diseño planificado y sistemático de acciones que se requieren para asegurar la calidad del software"*

Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después. El aseguramiento de la calidad del software engloba: (1) un enfoque de gestión de calidad, (2) métodos y herramientas de Ingeniería del Software, (3) revisiones técnicas formales aplicables en el proceso de software, (4) una estrategia de prueba multiescala, (5) el control de la documentación del software y de los cambios realizados, (6) procedimientos para ajustarse a los estándares de desarrollo del software y (7) mecanismos de medición y de generación de informes. El mismo tiene asociado 2 constitutivos diferentes: los Ingenieros de Software que realizan el trabajo técnico y un grupo de SQA (Software Quality Assurance) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

Las actividades del grupo de SQA son: (1) Establecimiento de un plan de SQA para un proyecto, (2) Participación en el desarrollo de la descripción del proceso de software del proyecto, (3) Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido, (4) Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software, (5) Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido, y (6) Registrar lo que no se ajuste a los requisitos e informar a sus superiores. (Pressman, 2005)

Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y; ayuda a recopilar y analizar las métricas del software.

Cuando se habla de software nos referimos a la disciplina de recopilar y analizar datos basándonos en mediciones reales de software, así como a las escalas de medición. El término producto se utiliza para referirse a las especificaciones, a los diseños y a los listados del código. Los valores de las métricas no se obtienen sólo por mediciones. Algunos valores de métricas se derivan de los requisitos del cliente o de los usuarios y, por lo tanto, actúan como restricciones dentro del proyecto.

Las medidas de Calidad del Software deben comenzar desde la especificación y terminar con la implementación, implantación y mantenimiento o post-implantación. Debe aplicarse a lo largo de todo el proceso de Ingeniería de Software. Básicamente, la medición es una fase normal de

cualquier actividad industrial Sin mediciones es imposible perseguir objetivos comerciales normales de una manera racional.

Existen métricas a nivel Proyecto, Proceso y Producto respectivamente. Las métricas a recabar dependen de los objetivos del negocio en particular. Los desarrolladores tienen a la vez objetivos comunes como, respetar el presupuesto y respetar los plazos, minimizar las tasas de defectos antes y después de la entrega del producto e intentar mejorar la calidad y la productividad. Las métricas deben ayudar a la evaluación de las representaciones del modelo lógico y físico, deben tener la capacidad de intuir sobre la complejidad del diseño y construcción; y deben ayudar en el diseño de casos de prueba.

1.9.4. Mejora de la Calidad de Software.

Según la Norma ISO 9000:2000, la mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad. Es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y Auditorías, a efectuar mejoras en la calidad del software.

Una Auditoría de Calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos.

Los pasos para la realización de una Auditoría de la Calidad del Software son: (1) Identificación del Producto de software que se pretende auditar, (2) Determinar los Requisitos aplicables, (3) Relevar la información necesaria para el cálculo de las métricas de los requisitos aplicables, (4) Evaluar la Calidad del Software usando las métricas respecto de los requisitos establecidos en el paso 2 y determinar su cumplimiento, (5) Establecer las acciones correctivas respecto de los requisitos evaluados y (6) Elaborar un Informe Final.

El Informe de Auditoría de la Calidad es un medio formal para comunicar los objetivos de la Auditoría, el cuerpo de las normas de Auditoría, el alcance de la Auditoría, y los hallazgos y conclusiones. Refleja los objetivos, alcances, observaciones, recomendaciones y conclusiones del proceso de evaluación, relacionados con las áreas de informática. Debe incluir suficiente información para que sea comprendido por los destinatarios esperados y facilitar las acciones correctivas.

Existen esquemas recomendados respecto de los informes de auditorías con los requisitos mínimos aconsejables respecto a estructura y contenido. Los puntos esenciales de un Informe de Auditoría son: (1) Identificación del Informe, (2) Identificación del Cliente, (3) Identificación de la Entidad auditada, (4) Objetivos de la Auditoría Informática, (5) Normativa aplicativa y excepciones, (6) Alcance de la Auditoría, (7) Conclusiones: Informe corto de opinión , (8) Resultado: Informe largo y otros informes, (9) Informe previo, (10) Fecha del Informe, (11) Identificación y firma del Auditor, (12) Distribución del Informe y (13) Conclusiones.

Los resultados que se obtienen en una auditoría son documentados y remitidos al líder del proyecto auditado, a la entidad auditora, y cualquier organización externa identificada en el plan de Auditoría. Por último, se puede decir que la Auditoría de la Calidad del Software puede traer aparejado la certificación del software, lo cual permite mejorar el nivel de competitividad de la empresa con sus respectivas consecuencias.

Para implementar un programa de mejoras es necesario definir procesos, decidir qué se quiere mejorar, definir qué medidas serán necesarias recoger, cómo y dónde tomarlas, gestionarlas mediante herramientas, utilizarlas para la toma de decisiones y reconocer las mejoras. Cuando el proceso a mejorar es el de desarrollo del software, es importante definir qué objetivos se quieren alcanzar, para reducir el número de medidas y, en consecuencia, el coste de recopilarlas y el impacto sobre la actividad de producción de software.

1.10. Conclusiones parciales.

- ✓ El análisis del estado del arte permitió definir al modelo de calidad CMMI como el más apropiado para ser aplicado en el proyecto Convenio Cuba – Venezuela, debido a que este no solo busca el éxito en un proyecto en particular, sino que sea mantenido para todos los demás de la organización, con costos cada vez menores y permitiendo aumentar las ganancias de la empresa.
- ✓ Se determinó además la aplicación de los estándares de la norma ISO 9000 para el establecimiento de los objetivos de calidad y las métricas, además para la gestión de la configuración de software y para el aseguramiento de la calidad.
- ✓ Se estudiaron los factores relacionados con la calidad, lo que permitió conocer las características que debía tener un producto para que tuviera niveles adecuados de calidad.

- ✓ El análisis de la Gestión de la Calidad posibilitó conocer la existencia de un conjunto de actividades dirigidas a desarrollar un producto con calidad, entre ellas el aseguramiento como parte de este proceso.
- ✓ El estudio de las métricas fundamentales aplicadas al proceso de desarrollo de software a nivel nacional e internacional permitió identificar un conjunto de las mismas para el proyecto Convenio Cuba – Venezuela, encaminadas a dar cumplimiento a las metas de planificación, control y seguimiento, y mejoramiento de la calidad del producto de software.

CAPÍTULO 2: PROPUESTA DE UNA ESTREATEGIA DE ASEGURAMIENTO DE LA CALIDAD

2.1. Introducción.

Una de las principales fases dentro de la elaboración de un proyecto es el Aseguramiento de la Calidad del Software (SQA) y su importancia se debe a que es un modelo sistemático y planeado de todas las acciones necesarias para proveer la confianza adecuada, según los requerimientos técnicos establecidos, de cada producto e ítem del proyecto.

La actividad del aseguramiento de calidad es un proceso que incluye la verificación de que los estándares sean aplicados correctamente. Como se sabe, en proyectos pequeños esto se puede realizar por el equipo de desarrollo, pero en proyectos grandes, un grupo específico se debe dedicar a este rol.

En este capítulo se presenta en su primera parte una propuesta sobre el chequeo y revisión que debe realizar el grupo de aseguramiento de la calidad de un proyecto en los planes que se generan en cada fase del proceso de desarrollo de software que contribuirán con el aseguramiento de la calidad, específicamente en este caso para el proyecto Convenio Cuba-Venezuela. Luego de esto se propone un Plan de Aseguramiento de la Calidad para el proyecto y por último se presentan varias actividades de revisión para el chequeo de la administración, de la documentación, de la adherencia a los estándares y de las revisiones y las auditorías. Este documento pretende entregar la pauta general del proceso que debe seguir una Gerencia de SQA para la creación de un software con la calidad requerida.

2.2. Chequeo de los planes del proyecto para cada fase de desarrollo.

Para que exista un buen aseguramiento de la calidad en un proyecto existen 3 principales puntos de vista que son la base fundamental del modelo de SQA.

- ✓ Se definen los planes de acuerdo a estándares y normas.
- ✓ Se realizan los procedimientos de acuerdo a los planes.
- ✓ Se implementan los productos de acuerdo a estándares y normas.

Un procedimiento define cómo una actividad será llevada a cabo. Los procedimientos son definidos en los planes, como por ejemplo el Plan de Gestión de Configuración de Software.

Un producto es un entregable al cliente, ya sea, código, manual de usuario y documentos técnicos. Un estándar será utilizado dentro del proyecto para que el producto final tenga uniformidad, ya sea, en el código fuente, en el diseño, en sus documentos, etc.

SQA debe planear qué chequeos debe hacer tempranamente en el proyecto, el criterio de selección más importante para asegurar la calidad es el riesgo. Las áreas comunes de riesgo en el desarrollo de software son: complejidad, capacidad y experiencia del personal, procedimientos manuales y la madurez organizacional. Es por esto que el personal encargado del aseguramiento de la calidad debe concentrarse en estos aspectos que tienen gran influencia en la calidad del producto, por lo que deben chequear lo más rápido posible:

- ✓ El proyecto debe ser propiamente organizado, con un ciclo de vida apropiado.
- ✓ Los miembros del equipo de desarrollo deben tener tareas y responsabilidades bien definidas.
- ✓ Los planes documentados deben ser implementados.
- ✓ La documentación y los estándares de código deben ser seguidos y aplicados.
- ✓ Los estándares, prácticas y convenciones deben estar adheridas al proyecto.
- ✓ Los datos de métricas deben ser reunidos y usados para mejorar los productos y procesos.
- ✓ Las revisiones y auditorías deben tomar lugar y ser propiamente dirigidas.
- ✓ Las pruebas deben ser especificadas y rigurosamente llevadas a cabo.
- ✓ El proyecto debe usar las herramientas, técnicas y métodos apropiados.
- ✓ El software debe ser almacenado en librerías controladas.
- ✓ El software debe ser almacenado de manera segura.
- ✓ El personal debe ser propiamente entrenado.
- ✓ Los riesgos del proyecto deben ser minimizados.

La administración del proyecto es la encargada de la organización de las actividades de SQA y de la definición de los roles de SQA y su asignación al personal que ocupará dicho rol. En el proyecto, diferentes grupos tienen sus propios requerimientos de SQA, pues la administración necesita saber que el software es construido de acuerdo a los planes y a los procedimientos que se han definido. Los desarrolladores también necesitan saber que su trabajo es de buena calidad y realizado de acuerdo a estándares establecidos.

Una administración efectiva del proyecto depende de un ciclo de control a lo largo de la producción del software, este comienza con los requerimientos de usuario y los estándares para crear los planes, ya sean: Plan de Desarrollo de Software, Plan de Gestión de Configuración de Software, Plan de Validación y Verificación de Software y el Plan de Aseguramiento de la Calidad de Software. Los planes son usados para controlar la creación del producto. Una vez construido el software, este se monitorea a través de los planes, hasta lograr después de varios ciclos de comprobación que el producto sea entregado con la calidad requerida.

El personal de SQA hace una crítica contribución en el control y monitoreo de las actividades en la dirección del proyecto ya sean:

- ✓ Escribir el Plan de Aseguramiento de la Calidad de Software.
- ✓ Revisar el Plan de Desarrollo de Software, Plan de Gestión de Configuración de Software y Plan de Validación y Verificación de Software.
- ✓ Aconsejar a la dirección del proyecto en cómo aplicar los estándares y los planes.
- ✓ Aconsejar a los desarrolladores en cómo aplicar los estándares para las tareas de producción.

2.2.1. Revisión en la fase de Inicio.

El personal de aseguramiento de la calidad debe estar involucrado desde la primera oportunidad en el proyecto de desarrollo de software. Deben revisar los requerimientos de usuario de acuerdo a los estándares leyendo todo el documento y asegurarse que el mismo contenga paso a paso todo lo que el usuario quiere que se haga, o sea, debe contener todos los requerimientos de usuarios bien conocidos. Para confirmar de manera exhaustiva, SQA buscará pruebas de las actividades de captura de requisitos, tales como: Encuestas y Entrevistas.

SQA debe chequear que el usuario haya declarado todas las limitaciones que quiere imponer en el software, como: portabilidad, disponibilidad y usabilidad, así como también deben

aparecer las interfaces externas descritas con sus limitaciones de diseño. Los requerimientos de usuario deben ser verificables.

La aceptación del Plan de Prueba define el alcance y el enfoque hacia la validación, por lo que SQA debe chequear que el plan permitirá que los requerimientos sean validados. Este plan es documentado en el Plan de Validación y Verificación de Software.

Al final de la fase de requerimientos de usuarios se realiza una revisión en la que el SQA debe formar parte y chequear que se realicen los procedimientos adecuados. Esta revisión decidirá si hay algunos requerimientos que no pueden ser aplicables, pero que no serán eliminados, pues puede que en un futuro los desarrolladores los necesiten.

Este documento tiene una fuerte influencia en todo el desarrollo y es muy rentable si algún problema es identificado y corregido lo más temprano posible.

Luego de la revisión final del los requerimientos de usuario los desarrolladores deben producir 4 planes, que ya han sido mencionados anteriormente, pero que se realizan después de dicha revisión, estos son: Plan de Desarrollo de Software, Plan de Gestión de Configuración de Software, Plan de Validación y Verificación de Software, Plan de Aseguramiento de la Calidad de Software; este último es producido por el personal de SQA y los demás también son revisados por los mismos, para comprobar que están de acuerdo a los estándares.

2.2.1.1. Plan de Desarrollo de Software.

Cada proyecto debe hacer un Plan de Administración de Proyecto de Software, por lo que el personal de SQA debe cerciorarse que éste contenga bien declarado los objetivos del proyecto y chequear que todos los planes son consistentes con dichos objetivos. Además de revisar que el Plan de Desarrollo de Software describa todas las actividades de la fase de requerimientos de usuario y que contenga un estimado del costo total del proyecto.

El personal de SQA verificará que se define el enfoque del ciclo de vida del proyecto apropiado y que ha sido correctamente ajustado al mismo. Se debe chequear también que el plan sea suficiente para minimizar los riesgos y que sobre todo sea un plan realista. Todos los planes tienen suposiciones viables, dependencias a eventos externos y están sujetos a limitaciones, de ahí que sea necesario que lo anteriormente expuesto esté bien direccionado en el análisis de los riesgos. Por último, las métricas son importantes para la evaluación del proyecto, por lo que se verificará que cada una sea apropiada para el mismo.

2.2.1.2. Plan de Gestión de Configuración de Software.

En este documento se definen los procedimientos de control del proyecto antes de que comience la producción. El personal de SQA necesita chequear que los mismos están propiamente definidos y llevados a cabo, o sea, verificar que el desarrollo del software es correctamente controlado. Se debe verificar que el plan esté completo, que existan procedimientos de control de cambios, procedimientos de almacenamiento, mecanismos de seguimiento de los cambios, etc.

Varios son los documentos que son las primeras salidas de esta fase y estos deben ser lo suficientemente flexibles para ser usados en todo el proyecto. Los aspectos claves que deben ser dirigidos son:

- ✓ Identificación de Documento.
- ✓ Almacenamiento de Documento.
- ✓ Documento de Control de Cambio.

Toda la documentación relevante del proyecto debe ser identificada de manera única, ya sea por el nombre de la compañía, el nombre del proyecto, el tipo de documento, el nombre del documento y el número de versión del mismo. Cada página del documento debe ser marcada con un identificador. El personal de SQA debe verificar que todos los documentos del proyecto hayan sido correctamente redactados.

Durante la vida del proyecto se generarán múltiples documentos, por lo que SQA debe chequear mediante auditorías físicas que una copia de cada uno de ellos debe estar almacenada en una biblioteca de archivos.

2.2.1.3. Plan de Validación y Verificación de Software.

Las técnicas de verificación incluyen revisión, pruebas formales y traceo, la revisiones intermedias son usadas para verificar el producto y luego para aclarar los problemas antes de realizar una revisión formal.

El personal de SQA debe examinar el plan y chequear que los procedimientos de revisión están bien definidos, que estén involucradas las personas correctas, y que se haya realizado la cantidad suficiente de revisiones; así como chequear que el enfoque de verificación es suficientemente riguroso para asegurar la corrección del documento de requerimientos de software. Los métodos formales para especificación y verificación de software son a menudo necesarios cuando las cuestiones de protección y de seguridad están involucradas.

El Plan de Validación y Verificación de Software necesita definir cómo los requerimientos de usuarios serán llevados a requerimientos de software, por lo que el personal de SQA tiene que chequear la matriz de trazabilidad y es muy importante que esos procedimientos de traceo sean fáciles de usar.

También se debe crear un plan de prueba de aceptación tan pronto como el documento de requerimientos de usuario esté disponible. El nivel de detalle requerido depende del proyecto, todos los planes deben definir el enfoque de validación de requerimientos de usuario, por lo que SQA debe chequear que el plan está suficientemente detallado para permitir la estimación del esfuerzo requerido para las pruebas de aceptación.

2.2.2. Revisión en la fase de Elaboración.

En esta fase un reconocido método de diseño de software debe ser adoptado y aplicado consistentemente. El mismo debe apoyar la construcción de un modelo físico que defina cómo el software trabaja.

Al principio de esta fase, SQA debe chequear que:

- ✓ Un método de diseño apropiado será usado.
- ✓ Los desarrolladores han usado dicho método anteriormente, o recibirán entrenamiento.
- ✓ El método de diseño puede ser apoyado por herramientas CASE.
- ✓ El nivel más bajo del diseño arquitectónico ha sido acordado.

El diseño arquitectónico debe ser revisado capa por capa. Estas revisiones intermedias ayudan a:

- ✓ Prevenir que el diseño se convierta muy complejo de probar.
- ✓ Chequear que la viabilidad de cada componente principal ha sido probado.
- ✓ Garantizar que el diseño será fiable, mantenible y seguro.

En el documento de diseño arquitectónico SQA debe chequear que se definan las funciones, entradas y salidas de los principales componentes del software. El documento debe contener definiciones de las estructuras de datos de los componentes de interfaz, y se debe definir el flujo de control entre componentes, así como incluir una referencia de la matriz de trazabilidad de los requerimientos de software a los componentes. Este documento le da a cualquier

integrante del proyecto visibilidad del sistema como un todo. Por tanto SQA tiene la responsabilidad de garantizar que los diseñadores hagan el mismo, fácil de entender.

2.2.2.1. Plan de Desarrollo de Software.

En esta fase el Plan de Desarrollo de Software es un poco más largo y complejo. El personal de SQA debe chequear que el plan contiene:

- ✓ Una relación de actividades de codificación, de integración y de pruebas.
- ✓ Que define un camino mostrando el tiempo requerido para la terminación de la fase.
- ✓ Actividades esenciales como las revisiones.
- ✓ Estimaciones realistas del esfuerzo requerido para cada actividad.

Actividades de SQA programadas como las auditorías.

2.2.2.2. Plan de Gestión de Configuración de Software.

La principal salida de esta fase es el código, por lo que este plan estará extendido a cubrir:

- ✓ Identificación de código.
- ✓ Almacenamiento de código.
- ✓ Control de cambio de código.

El personal de SQA debe revisar para confirmar que todos los ítems del software, la documentación, el código fuente, el código objeto, el ejecutable, los archivos de datos y las pruebas de software están cubiertos por el plan. En este se debe definir un estándar para todo el código fuente y SQA debe realizar auditorías físicas para verificar que los ítems tangibles del software se encuentran en su correcta ubicación.

El sistema de librerías del software también es algo básico en este plan y este debe ser examinado para comprobar que el acceso a las mismas es controlado y comprobar también que el software no puede perderse a través de actualizaciones simultáneas.

Debe quedar plasmada una copia de seguridad (backup) del plan. Todas las versiones del sistema deben ser retenidas y en auditorías se debe comprobar el almacenamiento y los registros de dichos backup.

Buenos procedimientos de cambio son esenciales, SQA debe examinarlos y estimar el tiempo total necesitado para procesar un cambio a través de la administración de configuración de sistema. Esta vez debe ser pequeño comparado con el tiempo requerido para hacer los trabajos técnicos necesarios. Si el proceso de cambio consume mucho tiempo entonces hay una alta probabilidad de explotar en períodos de Stress, ya cuando la entrega se está acercando y cuando esto sucede aumenta la presión en los desarrolladores. Por tanto el personal de SQA debe chequear que esta configuración pueda manejar el volumen de cambio esperado y si es necesario hacerlo más eficiente.

También se debe monitorear la ejecución de los procedimientos que se describen en el plan, así como, examinar las tendencias de la ocurrencia de problemas para hacer significativas comparaciones e ir evaluando la calidad del producto. SQA debe clasificar los problemas en:

- ✓ Error de usuario.
- ✓ Error de documentación de usuario.
- ✓ Error de código.
- ✓ Error de diseño.
- ✓ Error de especificación de requerimientos.

2.2.2.3. Plan de Validación y Verificación de Software.

Las técnicas usadas para verificar el diseño detallado son similares a las de la fase anterior. Como antes, revisiones intermedias son usadas para acordar el diseño capa por capa y la última revisión es la inspección de código. Esto es realizado después de que los módulos han sido satisfactoriamente compilados pero antes de las pruebas de unidad, este tipo de revisiones antes de dichas pruebas suelen ser muy rentables. El personal de SQA debe asegurarse de que este vital paso no se pierda. Las métricas de inspección deben incluir:

- ✓ Líneas de código por módulo.
- ✓ Complejidad ciclomática por módulo.
- ✓ Número de errores por módulo.

Se debe chequear que los valores de estas métricas están en los límites definidos en los estándares de diseño y de código.

El plan de integración de prueba debe ser producido tan rápido como el documento de la fase esté disponible y debe definir los enfoques de integración. Posteriormente del plan de integración se debe aumentar el uso de integración de software y disminuir el uso de pruebas de software que sustituyen componentes que son integrados más tarde.

2.2.3. Revisión en la fase de Construcción.

SQA debe chequear que el software esté documentado de la misma forma en que ha sido producido. A menudo la documentación se aplaza porque los desarrolladores están bajo presión, y esto constituye un grave error. SQA puede asegurar que la documentación y la producción sean concurrentes por la revisión de software a medida que se vaya produciendo. Si SQA retrasa su revisión, los desarrolladores deben aplazar por escrito la documentación hasta justo antes de que SQA esté listo.

El diseño detallado debe revisarse capa-por-capa. Estas revisiones deben incluir a los directores técnicos y a los diseñadores interesados. SQA debe asistir a algunas de estas reuniones de revisión, sobre todo cuando se refieren a la aplicación de la calidad, fiabilidad, mantenimiento y requisitos de seguridad. SQA también debe comprobar que la segunda parte del documento es una extensión lógica de la estructura definida en el documento de Elaboración, y que existe una sección para cada componente de software.

2.2.3.1. Codificación.

SQA debe inspeccionar el código. Esta actividad puede ser parte de un proceso de inspección utilizada por el personal de desarrollo, o puede ser una actividad independiente del aseguramiento de la calidad. Se debe verificar que los estándares de codificación han sido observados. La calidad del código debe ser evaluada con la ayuda de métricas estándares como el módulo de longitud, la complejidad y el tipo de comentario. Herramientas estáticas de análisis deben utilizarse para apoyar la recopilación de datos y métricas de evaluación.

2.2.3.1. Reutilización.

La reutilización de software de un proyecto a otro es cada vez más común. SQA debe asegurarse de que el software reutilizado ha sido desarrollado de acuerdo a normas aceptables. El software puede ser confiable en un entorno operacional, pero no en otro. SQA debe tratar con suma cautela que la calidad del software reutilizable es probada a través de un exitoso uso operativo. SQA debe comprobar que el desarrollo y mantenimiento de estándares y registros del software estén aptos para su reutilización.

El software reutilizado es comprado con frecuencia fuera de la plataforma. Este software es usualmente llamado "software comercial". SQA debe comprobar que la certificación de la calidad del software comercial proveedor cumpla con los estándares del proyecto.

2.2.3.2. Prueba.

El software que no ha sido probado es muy poco probable que funcione. El personal de SQA juega un rol vital en proporcionarles a la administración y a los usuarios que las pruebas se han realizado correctamente. Para esto SQA debe chequear que las actividades de prueba:

- ✓ Son apropiadas para el grado más crítico del software.
- ✓ Cumplan con la verificación y aceptación de las pruebas.
- ✓ Sean propiamente documentadas en el Plan de Validación y Verificación de Software.

SQA lleva esto a cabo revisando las especificaciones de pruebas, observando como son ejecutadas y participando en algunas pruebas seleccionadas, normalmente no serán capaces de participar en todas las pruebas pero en el plan de aseguramiento de la calidad se debe dejar bien claro en cuáles participarán u observarán. Como a partir de las pruebas se mide el progreso del software, se deben definir programas de métricas que incluyen medidas como:

- ✓ Número de fallos.
- ✓ Número de fallos por cada prueba.
- ✓ Número de fallo por cada ciclo de prueba.

Se debe chequear que se hayan realizado las pruebas de unidad, primeramente las de caja blanca ya que aseguran que el software está haciendo el trabajo de la manera que se quiere y

luego las pruebas de caja negra para comprobar su funcionalidad, ya sea para chequear la ocurrencia de errores como por ejemplo entradas inválidas de datos.

Los procedimientos de las pruebas deben ser escritos, así que estas pueden ser ejecutadas por cualquier persona que no sea desarrolladora del software, SQA debe confirmar que estos se explican de manera entendible para quien vaya a ejecutar dichas pruebas. También se deben asegurar que las pruebas que detectaron errores deben ser realizadas nuevamente después de que la corrección se haya hecho. Por último se debe chequear que los autores del software no pueden modificar componentes que ya hayan sido almacenados una vez revisados correctamente y que no necesiten cambios.

2.2.3.3. Revisiones Técnicas Formales.

Cada proyecto debe revisar antes de la entrega del software, para comprobar si está listo para la fase de Transición. SQA debe participar en el proceso de revisión y hacer recomendaciones sobre la base de:

- ✓ Los resultados de las pruebas.
- ✓ Los resultados de las auditorías.
- ✓ Análisis de los problemas pendientes.

SQA debe realizar una auditoría física del software, mediante la verificación de la configuración de la lista de componentes antes de la transición. La lista debe contener todos los componentes liberados del software y estar especificada en el plan del proyecto.

SQA debe realizar una auditoría funcional del software mediante la verificación de los requerimientos de software contra la matriz de trazabilidad de los componentes de software.

Una segunda actividad es comprobar que el Plan de Validación y Verificación de Software contiene pruebas para cada requisito de software, y que cada una de estas se han llevado a cabo. Las actividades de la auditoría funcional deben comenzar lo más temprano posible, tan pronto como las especificaciones de prueba están disponibles y SQA no aprobara el paso a la etapa de Transición con problemas principales pendientes.

2.2.3.4. Plan de Desarrollo de Software.

La confianza del cliente en el software aumenta cuando la fase de transición está libre de problemas. Esto es más probable cuando las actividades de instalación y las pruebas de aceptación son cuidadosamente planeadas. El software es instalado y las pruebas de aceptación se ejecutan. SQA debe comprobar que estas dos actividades se han ensayado en la fase de Elaboración. Estos ensayos deben permitir la estimación exacta del esfuerzo en la fase de Transición.

Cuando el software es para ser incorporado en un sistema más amplio u operado en un entorno diferente, la transición del software puede ser mucho más complicada. Las pruebas de aceptación comprueban que el software se integra correctamente con el objetivo del entorno. Los problemas siempre ocurrirán cuando el software sea ejecutado en su entorno por primera vez. Los cambios serán necesarios y el Plan de Desarrollo de Software debe permitir los mismos.

SQA debe chequear las estimaciones realistas de las pruebas y reparar el trabajo, así como verificar si hay nuevos problemas para que sean diagnosticados y rápidamente corregidos. El documento debe dejar bien claro que usuarios o clientes, desarrolladores y SQA deberán asistir a las pruebas de aceptación.

2.2.3.5. Plan de Gestión de Configuración de Software.

Este plan debe definir los procedimientos de administración de configuración para el producto que se entregue en su entorno operacional. SQA debe comprobar que este plan:

- ✓ Identifique los elementos a entregar.
- ✓ Defina los procedimientos para el almacenamiento y la copia de seguridad de los productos terminados.
- ✓ Defina los procedimientos de control de cambios.
- ✓ Defina los procedimientos de reporte de problemas.

2.2.3.6. Plan de Validación y Verificación de Software.

En esta fase el plan se expande a la inclusión de pruebas de diseño, casos de pruebas, procedimientos de pruebas y resultados de pruebas para las pruebas de unidad, de integración y de sistemas. El documento debe contener:

- ✓ Aceptación de pruebas de diseño.
- ✓ Aceptación de casos de pruebas.
- ✓ Aceptación de procedimientos de pruebas.

SQA debe confirmar que existe una prueba de diseño para cada requerimiento y que los casos de pruebas deberán estar marcados para usarse en aceptación provisional o final.

2.2.4. Revisión en la fase de Transición.

Después de la construcción del software, este es instalado usando los procedimientos definidos. SQA debe monitorear dicho proceso.

Las pruebas de aceptación necesarias para una aceptación provisional, ahora se ejecutan. SQA debe comprobar que los usuarios o clientes estén presentes como testigos en dichas pruebas, y que cada una de ellas debe ser firmada por los desarrolladores y los usuarios o clientes.

Después de esto es necesario realizar una reunión para revisar el rendimiento del software y decidir si el mismo puede ser provisionalmente aceptado, esto puede generar dos resultados:

- ✓ Rechazo del software.
- ✓ Prueba de aceptación condicional del software.

Este último es el más común, la aceptación se hace condicional bajo las acciones definidas en dicha reunión.

El documento de esta fase es una salida obligatoria y el personal de SQA debe inspeccionarlo, la primera revisión debe ser antes de la entrega, pues en esta etapa el documento contiene secciones listando los entregables, procedimientos de instalación y de construcción. La segunda inspección será realizada al final de la fase cuando son adicionadas las secciones donde se describen los resultados de las pruebas de aceptación, los reportes de problemas de software, los pedidos de cambios en el software y los resultados de la modificación del software.

Luego de la instalación es necesario darle mantenimiento al software, ya que el mismo puede ser arruinado por un pobre mantenimiento. SQA debe monitorear la calidad de software en esta fase para que no sea degradada, por lo que deberá chequear que:

- ✓ La configuración de software está adecuadamente gestionada.
- ✓ El código y la documentación se mantienen actualizados.

El desarrollo de los planes debe cubrir el período hasta la aceptación final y todos los proyectos deben tener un hito para dicha aceptación, por tanto, antes que este llegue, SQA verificara si:

- ✓ Todas las pruebas de aceptación han sido exitosamente completadas.
- ✓ Todo el historial de documentos del proyecto está listo para la entrega.

Después de la aceptación final, debe definirse una organización para el mantenimiento la cual debe ser revisada para comprobar que está propiamente distribuida y que se hará de la manera correcta.

2.3. Plan de Aseguramiento de la Calidad propuesto.

2.3.1. Introducción.

El aseguramiento de la calidad del software (SQA) es un patrón que contiene acciones planificadas y sistemáticas dirigidas a asegurar la calidad del producto. Con éste se desea lograr una planificación de las actividades que se realizarán durante el proceso de desarrollo del software de modo que se obtengan resultados satisfactorios. Se pretende además una planificación lo más real posible y que se ajuste a las características del proyecto, teniendo en cuenta el mejor aprovechamiento de recursos y esfuerzos para su desarrollo.

2.3.2. Propósito.

Se persigue describir cómo se asegurará la calidad del producto de forma general, lo artefactos, herramientas y procesos a través de los cuales se pretende lograr dicho propósito.

2.3.3. Alcance.

Este plan es realizado a partir del plan que se propone por la universidad pero que ha sido modificado de acuerdo a las características del proyecto Convenio Cuba Venezuela por lo cual solo se puede aplicar en él, en todos los módulos que posee.

2.3.4. Definiciones, Acrónimos y Abreviaturas.

- ✓ SQA: Software Quality Assurance (Aseguramiento de la Calidad del Software).
- ✓ SQAP: Software Quality Assurance Plan (Plan de Aseguramiento de la Calidad del Software).

2.3.5. Referencias.

Tabla 1. 2 Planes de referencia para conformar el SQAP

Código	Título
01PDS	Plan de Desarrollo de Software.
02PP	Plan de Pruebas.
03PI	Plan de Iteraciones.
04DAS	Documento de arquitectura de software.
05PACS	Plan de Administración de Configuración de Software.

2.3.6. Resumen.

En el plan que se propone se especifica detalladamente qué se deberá hacer en cada fase para cumplir con dicho objetivo. Para facilitar el trabajo y hacerlo organizadamente se han creado algunas reglas para el grupo de SQA que permitirán de alguna manera resumir las actividades que aquí se describen:

1. Establecimiento de un plan de SQA para el proyecto.
2. Participación en el desarrollo de la descripción del proceso de software del proyecto.
3. Revisión de las actividades de ingeniería de software para verificar su ajuste al proceso de software definido.
4. Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software.
5. Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
6. Registrar lo que no se ajuste a los requisitos e informarlo a los superiores.

2.3.7. Objetivos de calidad.

Los objetivos que se plantearon en el proyecto para el área de calidad son:

- ✓ Asegurar la calidad del trabajo, a través de la vigilancia, prevención, comprobación y valoración sistemática en el proyecto, a lo largo del ciclo de vida del mismo, velando por que el producto software cumpla con los requerimientos establecidos por el cliente.
- ✓ Mediante la gestión de riesgos identificar posibles errores antes de que se conviertan en puntos fatales o problemáticos.
- ✓ Mantener el trabajo sobre la base de los diferentes estándares y normas internacionales existentes.
- ✓ Velar y asegurar el desarrollo e implantación de un sistema informático que soporte las decisiones estratégicas del Cliente dentro del marco legal establecido.
- ✓ Velar porque los errores de un flujo de trabajo no continúen para otro en el modelado del sistema de tal manera que al finalizar la construcción del software exista la menor cantidad de errores posibles logrando de esta forma un producto de mayor calidad.
- ✓ Corresponder y hacer cumplir con los lineamientos de calidad establecidos por Calisoft para los proyectos productivos de la UCI.
- ✓ Lograr que el equipo de calidad cuente con el personal capacitado con el conocimiento y las habilidades necesarias para realizar las tareas y actividades encaminadas a lograr la calidad del proyecto.
- ✓ Colaborar con que la Gestión de Configuración y demás procesos de soporte sean desarrollados de tal manera que satisfaga las necesidades de la evolución del producto software y los procesos de producción del mismo.

2.3.8. Gestión.

2.3.8.1. Organización.

Tabla 1. 3 Relación de Roles del proceso de Aseguramiento de la Calidad

Rol	Descripción	Requerimientos Mínimos	Nombre y Apellidos
Responsable de Calidad	-Es una persona orientada al detalle. Asegura que la aplicación producida se ajusta a las especificaciones y está razonablemente libre de errores. -Proporciona una metodología para realizar las pruebas. -Coordina las pruebas de calidad interna, las pruebas de aceptación del cliente y pilotos de conjunto con el Laboratorio de Certificación. -Evalúa los resultados que se obtienen de las pruebas de calidad.	-Metodología RUP y UML. -Calidad de Software. -Ingeniería de Software. -Conocimientos básicos sobre el negocio.	
Diseñador de Pruebas	-Diseña los casos de prueba. -Evalúa y documenta el resultado de las pruebas realizadas al software. -Define listas de chequeo.	-Metodología RUP y UML. -Pruebas de Software.	
Revisor Técnico	-Chequea que los artefactos generados se ajusten a las pautas y lineamientos establecidos para su confección.	-Metodología RUP y UML.	
Probador	-Ejecuta las pruebas diseñadas. -Anota los resultados obtenidos.	-Conocimientos del negocio. -Habilidades mínimas de computación.	

2.3.9. Tareas y Responsabilidades.

Tabla 1. 4 Lista de tareas y responsables

Tarea de SQA	Precondición Al finalizar la fase:	Pos Condición Antes de la fase:	Responsable	Comentarios
Definición del procedimiento de chequeos técnicos.			Jefe de Calidad	
Definición de métricas a usar en el proyecto.			Jefe de Calidad	
Planificación del procedimiento del chequeo técnico.			Jefe de Calidad	
Elaboración de las Listas de Chequeo.	Definición de los artefactos que se realizarán en el proyecto y metodología a seguir.		Ingeniero de Prueba	
Elaboración del plan de pruebas por módulos.			Ingeniero de Prueba	
Elaboración de los casos de pruebas que se desea aplicar.			Ingeniero de Prueba	
Ejecución de las pruebas planificadas.			Probador y programador implicado del módulo	
Ejecución de las listas de chequeo.			Revisor técnico	
Puesta en práctica del procedimiento de chequeos técnicos.			Equipo de Calidad	

2.3.10. Documentación.

Para la elaboración del plan de aseguramiento de la calidad se ha basado en los siguientes documentos rectores del proyecto:

- ✓ Plan de desarrollo de software.
- ✓ Plan de pruebas.
- ✓ Plan de iteraciones.
- ✓ Documento de arquitectura de software.
- ✓ Plan de administración de configuración.

2.3.11. Métricas.

Las métricas que se definen *(Anexo 2)* están encaminadas a entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo realizado a nivel de proyecto. Estas permitirán descubrir y corregir problemas potenciales en etapas tempranas del software, así como también, aportarán argumentos en la toma de medidas preventivas que son necesarias para mejorar el proceso de revisiones. Se propone el uso de métricas para el cumplimiento de las metas de planificación, control y seguimiento, y mejoramiento.

- ✓ Esfuerzo promedio por líneas de código fuente (ELC).
- ✓ Eficiencia del inspector en la fase de preparación (EIP).

Estas dos primeras métricas están dirigidas al cumplimiento de la meta de planificación y se cuestiona cuánto cuesta y cuánto consume el proceso de revisión.

- ✓ Promedio de defectos detectados por líneas de código fuente (DELC).
- ✓ Productividad de la revisión promedio (PR).
- ✓ Razón de preparación promedio de los inspectores (RPP).
- ✓ Densidad de defectos (DD).
- ✓ Promedio de líneas revisadas en cada revisión (PLCR).
- ✓ Porcentaje de re-inspección, sólo para inspecciones (PRI).
- ✓ Eficiencia del inspector en la fase de preparación (EIP).

- ✓ Cantidad total de líneas revisadas (TLCR).

Este segundo grupo de métricas están dirigidas al control y seguimiento y se cuestiona cuál es la calidad del software inspeccionado, en qué medida el personal técnico sigue los procedimientos establecidos para las revisiones y cuál es el estado del proceso de inspección.

Por último para el cumplimiento de la meta de mejoramiento se cuestionará cuál es la productividad del proceso de revisión

- ✓ Efectividad de eliminar los defectos en una revisión (EED) o la efectividad de Eliminar los defectos de la fase j en la revisión i (EEDi,j).
- ✓ Promedio de defectos detectados por líneas de código fuente (DELC).
- ✓ Productividad de la revisión promedio (PR).
- ✓ Razón de preparación promedio de los inspectores (RPP).
- ✓ Promedio de líneas revisadas en cada revisión (PLCR).
- ✓ Densidad de defectos (DD).
- ✓ Esfuerzo promedio por defectos detectados (EDE).

2.3.12. Estándares y Guías.

Tabla 1. 5 Listado de estándares y guías a usar en el proyecto.

Estándar	Ubicación	Comentario
Codificación en j2ee	Repositorio del Proyecto	Estándar de codificación
Codificación en .Net	Repositorio del Proyecto	Estándar de codificación
IEEE Std 1063-2001	Repositorio del Proyecto	Manual de usuario
IEEE 828-1998	Repositorio del Proyecto	Gestión de la configuración
IEEE 730-1998	Repositorio del Proyecto	Para el aseguramiento de la calidad de software
ISO/IEC 9126	Repositorio del Proyecto	Para definir los objetivos de calidad y métricas
ISO 9003	Repositorio del Proyecto	Dirigidas al control final y las pruebas.
ISO 9001	Repositorio del Proyecto	Para las etapas del ciclo de diseño, desarrollo y elaboración.

ISO 1901-2002	Repositorio del Proyecto	Revisiones y auditorías
RUP	Repositorio del Proyecto	Para el proceso de desarrollo de software
LMC (emitidos por la DCS)	Repositorio del Proyecto	Se aplica durante el proceso de desarrollo de software
CMMI v1.2	Repositorio del Proyecto	Para el aseguramiento de la calidad, gestión de configuración, gestión de requisitos.

2.3.13. Plan de Revisiones y Auditorías.

2.3.13.1. Tareas de Revisiones y Auditorías.

Se requiere como mínimo que se realicen revisiones e inspecciones ya sea en los requerimientos, en el análisis, en el diseño, o sea, se recomienda realizarlas al terminar cada etapa de desarrollo. Se debe aclarar que las revisiones son discusiones sobre artefactos propuestos y las inspecciones se realizarán sobre artefactos terminados. Queda estipulado de la siguiente manera:

- ✓ Revisiones de artefactos propuestos al final de cada fase.
- ✓ Inspecciones de artefactos terminados al final de cada fase.

Entre estos artefactos se encuentran los documentos como: el Plan de Desarrollo de Software , Plan de Gestión de Configuración de Software y Plan de Verificación y Validación de Software, así como toda la documentación que se generará por fase, los diagramas, el código fuente, ejecutables, etc.

Se realizan auditorías para la Gestión de la Configuración. La auditoría de configuración de software tiene un carácter complementario y se preocupa de si:

- ✓ Se ha hecho el cambio especificado en la UCI.
- ✓ Se han incorporado modificaciones adicionales.
- ✓ Se ha llevado a cabo o no una RTF.

Se realizan todas las auditorías que sean necesarias:

- ✓ Auditorías en proceso: son aleatorias y se envía un aviso con anticipación. Su propósito será revisar el trabajo actual que se realiza en el proyecto.

- ✓ Auditorías funcionales: se realizarán para verificar que el producto que se entrega satisface los requerimientos especificados.
- ✓ Auditorías físicas: se realizarán para verificar que realmente se entreguen el software físico y su documentación designados para ser entregados.

Se deberá realizar un plan de auditoría a la gestión de configuración de cada módulo, para comprobar que se hayan realizado todos los cambios que fueron planificados y su correcta documentación.

Tabla 1. 6 Plan de Auditorías a la Gestión de Configuración

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

2.3.13.2. Cronograma.

El cronograma de las tareas y actividades de calidad se define partiendo de la organización, definición de tareas y responsabilidades y contando con los Planes de Iteración y Desarrollo del equipo de software.

Tabla 1. 7 Plantilla del cronograma

Módulo	Fecha Inicio	Fecha Fin
Módulo Presentación		
Módulo Contratación		
Módulo de Seguimiento		
Módulo Financiamiento		
Módulo de Pago de contravalor		
Módulo de Misiones		
Módulo de Administración		
Pruebas de Integración		
Solución a las no conformidades detectadas		

2.3.13.3. Organización y Responsabilidades.

Tabla 1. 8 Relación de los responsables por cada tarea.

Organización	Responsable
Definición del procedimiento de chequeos técnicos	Responsable de Calidad
Definición de métricas a usar en el proyecto	Responsable de Calidad
Planificación de procedimiento de chequeo técnico	Responsable de Calidad
Elaboración de las listas de chequeo	Revisor Técnico
Elaboración del plan de pruebas por módulos	Ingeniero de prueba
Elaboración de los casos de pruebas que se desea aplicar.	Ingeniero de prueba
Ejecución de las listas de chequeo	Revisor Técnico
Puesta en práctica del procedimiento de chequeos técnicos	Equipo de Calidad
Ejecución de casos de prueba	Ingeniero de prueba

2.3.13.4. Resolución de Problemas y Actividades de Corrección.

El máximo responsable de SQA deberá designar a una persona del grupo del proyecto, que tenga las capacidades para resolver el problema. Es necesario comunicar al jefe de proyecto, el tiempo estimado para la corrección del error y chequear si produce cambios en la planificación. Una vez resuelto el problema, se debe notificar al líder de SQA del proyecto para corroborar la solución. En el caso que el mismo no se solucione en un tiempo adecuado, se debe concretar una reunión con el encargado de la tarea, el jefe de proyecto y el líder de SQA.

2.3.13.5. Herramientas, Técnicas y Metodologías.

Las herramientas, técnicas o metodologías especificadas que serán usadas para llevar a cabo las actividades de revisión y Auditorías en este plan estarán basadas en listas de chequeo y Procedimiento de las RTF.

Las listas de chequeo cuentan con varios puntos, los cuales serán clasificados antes de ser aplicados, para esto se debe tener en cuenta que puede ser: muy importante o menos importante, además para las evaluaciones del punto puede ser calificado en: (Preciso, Correctos, Incorrectos, Ambiguos) además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.

El procedimiento de las RTF tiene como propósito llevar la calidad a todos los artefactos elaborados por el equipo del proyecto y será aplicado a todos los artefactos generados en el mismo.

Las listas de chequeo que serán utilizadas en cada revisión están almacenadas en el Expediente de Proyecto y las mismas son:

- ✓ Lista de chequeo para Requisitos.
- ✓ Lista de chequeo para Análisis.
- ✓ Lista de chequeo para Diseño.
- ✓ Lista de chequeo para Implementación.
- ✓ Lista de chequeo para Despliegue.
- ✓ Lista de chequeo para Caso de Uso del Sistema.
- ✓ Lista de Chequeo para Modelo de Procesos.
- ✓ Lista de chequeo para el Manual de Usuario.

2.3.14. Pruebas y Evaluación.

Las pruebas posibilitan no solo la detección de errores de la aplicación, verificando el cumplimiento de los requisitos funcionales y no funcionales, errores ortográficos, abreviaturas, la funcionalidad de vínculos, botones, la ayuda, sino también verifican la correspondencia de la documentación con la aplicación, dígase especificaciones de casos de uso del sistema, manuales de usuario y de instalación. Se verifica además, la lógica interna del programa, pues

un buen resultado de la implementación no solo depende de la codificación sino también de las pruebas. La realización de un proceso adecuado facilita que no aumenten los costos y tiempo del proyecto, que disminuyan los riesgos asociados y además que se eleve el nivel de calidad del producto, de esta forma, una vez que el mismo llegue a manos del cliente, va a tener la menor cantidad de errores dando como resultado la satisfacción del usuario final.

Para las pruebas se realiza un Plan de Pruebas. En ese documento se describe de forma detallada cómo el Equipo de Calidad realizará todas sus pruebas. Su planificación, flujos de trabajo, herramientas y funcionamiento de los roles, deben quedar muy bien definidos.

Es elaborado a partir del propuesto por RUP, sólo que ha sido adaptado a las condiciones específicas del proyecto de forma general. Por tanto, su uso es muy particular del proyecto.

2.3.15. Herramientas, Técnicas y Metodologías.

- ✓ SADT (Técnicas de Análisis y Diseño Estructurado).

Está constituido por un lenguaje gráfico y métodos de administración.

- ✓ SSA (Análisis Estructurado de Sistemas).

Se compone principalmente de diagramas de flujo de datos (DFD), diccionarios de datos, representaciones lógicas de procedimientos y técnicas de estructuración de almacenamiento de datos.

- ✓ TSP (Proceso de Software en Equipo).
- ✓ Utilización de los conceptos de RUP.
- ✓ Utilización de normas internacionales.

2.3.16. Resolución de Problema y Acción Correctiva.

Los errores o incongruencias que se detecten durante las revisiones se reportarán en un documento de No Conformidades habilitado para el caso *(Anexo 3)*. Este documento se el hace llegar al equipo de desarrollo y al líder del proyecto para solucionar los problemas.

2.3.17. Gestión de Configuración.

En el desarrollo de software los cambios, debidos principalmente a modificaciones de requisitos y fallos, son inevitables.

- ✓ Se debe llevar un control y registro de los cambios con el fin de reducir errores.
- ✓ Evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o a las tareas realizadas por otros miembros del equipo de proyecto.
- ✓ La gestión de configuración se debe realizar durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que éste deja de utilizarse.
- ✓ Se debe facilitar el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.
- ✓ Deberá permitir el control del sistema como producto global a lo largo de su desarrollo.
- ✓ Se deberán obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de mejora de la organización.
- ✓ Se deberá mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan.
- ✓ Los elementos de configuración software como: ejecutables y código fuente, modelos de datos, modelos de procesos, especificaciones de requisitos y pruebas, deberán ser controlados.

La gestión de Configuración del Software será llevada a cabo por un equipo de trabajo conformado por: Jefe de Proyecto, Jefe de Calidad, Jefe de módulo y Jefe de Gestión de la Configuración; se hará el uso de herramientas de control como el Subversión y Trac como herramienta de planificación.

2.3.18. Registros de Calidad.

Los registros de calidad son documentos que guardan información específica y relacionada a un procedimiento o instrucción de trabajo. Éstos comprueban que el proyecto cumple sus procedimientos y normas establecidas.

Los tipos de registros que se guardarán serán:

- ✓ Actas de reuniones efectuadas.
- ✓ Listas de Chequeo aplicadas.
- ✓ Casos de Prueba.
- ✓ Los Informes de No Conformidades.
- ✓ Listas de errores encontrados en un artefacto.
- ✓ Órdenes de trabajo para los módulos del proyecto.
- ✓ Resúmenes de Resultados de las pruebas.
- ✓ Correos de trabajo y comunicación de intercambio de trabajo.
- ✓ Todos los registros serán debidamente guardados en el repositorio del proyecto.

2.3.19. Entrenamiento.

Los cursos de capacitación del grupo de calidad los imparte la facultad 3 y el cronograma depende de ellos, todos los miembros del equipo de calidad deben tener el perfil de calidad lo que los obliga a seguir los diferentes cursos que se definieron dentro del perfil.

2.4. Actividades para el chequeo del Plan de Aseguramiento de la Calidad.

Con el objetivo de lograr que el Plan de Aseguramiento de la Calidad sea bien implantado en el proyecto a desarrollar, se han diseñado un conjunto de actividades para medir en qué grado se cumple cada una de las mismas y confirmar que el producto final sea entregado con elevados niveles de calidad.

Estas actividades fundamentalmente deben chequear la administración, documentación, la adherencia a los estándares, la revisión, el testeo del producto así como las auditorías que se le apliquen al mismo.

Actividades para el chequeo de la administración.

Actividad	
Criterios de Entrada	-- Documento con la especificación de la estructura gerencial del equipo de SQA.
Revisión	- Examinar estructura gerencial de la organización encargada del SQA.

	Identificar tareas de cada integrante de la gerencia - Definir responsabilidades a cada integrante de la gerencia
Criterios de Salida	- Estructura de la administración del departamento SQA revisada.

Actividades para el chequeo de la documentación.

Actividad	
Criterios de Entrada	-- Plan de Documentación
Revisión	Revisión y análisis del plan de documentación. - Buscar discrepancias. - Discutir discrepancias con el líder del proyecto.
Criterios de Salida	Documentación revisada

Actividades para el chequeo de la adherencia a los estándares.

Actividad	
Criterios de Entrada	-- Documentación, diseño, código, comentarios, casos de prueba, métricas.
Documentación	- Monitorear adherencias de los documentos a los estándares
Diseño	- Monitorear adherencias del diseño a los estándares.
Codificación	- Monitorear adherencias de la codificación a los estándares.
Comentarios	- Monitorear adherencias de los comentarios a los estándares.
Prueba	- Monitorear adherencias de las pruebas a los estándares. - Monitorear adherencia de las pruebas a las prácticas definidas.
Métricas	- Revisar la métrica definida.
Conformidad	- Monitorear la conformidad que existe en el sistema.
Criterios de Salida	- Proceso de Documentación revisado. - Proceso de Diseño revisado. - Proceso de Codificación revisado. - Proceso de Comentarios Revisado. - Proceso de Pruebas revisado. - Métricas definidas revisadas. - Conformidad revisada.

Actividades para el chequeo de las revisiones y las auditorías.

Actividad	
Criterios de Entrada	-- Código, requerimientos de las revisiones.
Revisión	Revisar el propósito de cada revisión. - Participar en revisiones de código. - Examinar argumentos de revisión y auditoría. - Verificar que el mecanismo de revisión sea acorde al tipo de proyecto. - Identificar los requerimientos mínimos para las revisiones.
Criterios de Salida	- Proceso de Revisión y Auditoría revisadas.

2.5. Conclusiones parciales.

- ✓ Se propuso una serie de actividades de chequeo y revisión para el Plan de Desarrollo de Software, el Plan de Gestión de la Configuración de Software y el Plan de Validación y Verificación de Software del proyecto para que estos se desarrollen de la manera correcta y beneficien el proceso de producción, ya que todas las actividades se rigen por estos planes.
- ✓ Se desarrolló un Plan de Aseguramiento de la Calidad para el proyecto que dirigirá de la manera correcta a los desarrolladores a crear un producto final con adecuados niveles de calidad, aplicando las métricas, estándares, metodologías y herramientas apropiadas.
- ✓ Se propusieron varias actividades para hacer cumplir el plan de aseguramiento satisfactoriamente dirigidas a la administración, a la documentación, a la adherencia a los estándares y a las revisiones y auditorías.
- ✓ Se logró desarrollar de manera general una estrategia que aplicada correctamente contribuirá a obtener un producto con niveles adecuados de calidad.

CAPÍTULO 3: VALIDACIÓN DE LA ESTRATEGIA PROPUESTA MEDIANTE EL CRITERIO DE ESPECIALISTAS.

3.1. Introducción.

El objetivo de este capítulo es analizar y discutir los criterios especializados y los juicios críticos expresados por los especialistas seleccionados acerca de la validez y adecuación de propuesta a través del instrumento utilizado, para evaluar el cumplimiento del objetivo general de nuestro trabajo.

El criterio de especialistas es un instrumento rápido y eficaz por el potencial que posee para conformar, valorar y enriquecer criterios, concepciones, modelos, estrategias, metodologías, etc. En este caso para valorar la propuesta de estrategia de Aseguramiento de la Calidad para el proyecto Convenio Cuba–Venezuela se utilizó la encuesta *(Anexo 4).* Este instrumento, que es una técnica de adquisición de información de interés con un cuestionario previamente elaborado, permitió conocer la opinión y valoración de los especialistas seleccionados en una muestra sobre el asunto dado.

Las respuestas se escogieron de modo especial y se determinaron del mismo modo las posibles variantes de respuestas estándares, lo que facilitó la evaluación de los resultados.

3.2. Nivel de competencia de los especialistas.

Para la selección de los especialistas que harían la valoración de la estrategia propuesta se escogieron un conjunto de indicadores generales que permitieron obtener información más rica y actualizada. Éstos fueron: la labor que desempeñan actualmente (con énfasis en especialistas que desempeñan roles relacionados con el aseguramiento de la calidad), la calificación profesional, los años de experiencia en el tema de calidad y la categoría docente y científica; cuidando que existiera una mayor representatividad de líderes de proyecto en aquellos indicadores que se consideran más significativos dentro de los más generales para la obtención de resultados.

De los especialistas, tres trabajan en la Dirección Central de Calidad de Software de la Universidad. Un especialista tiene más de 35 años de trabajo en la producción de Software desempeñando su actividad profesional en el Instituto Central de Investigaciones Digitales, ICID. Cuatro de ellos desempeñan el rol de especialistas de calidad en proyectos de exportación.

De acuerdo a la encuesta aplicada se obtuvo el siguiente resultado con respecto a la autovaloración del nivel de competencia de cada uno de los especialistas (Figura 3):

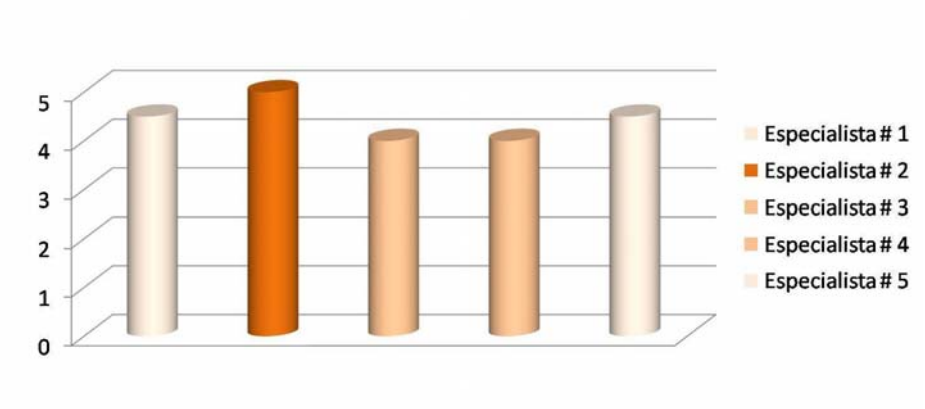

Figura 3 Nivel de autovaloración de los especialistas en una escala del 1 al 5

Para conocer tres aspectos fundamentales como son los conocimientos teóricos del tema, la experiencia obtenida en la actividad práctica y las certificaciones que pudieran tener en esta área, la encuesta proporcionó la siguiente información (Figura 4):

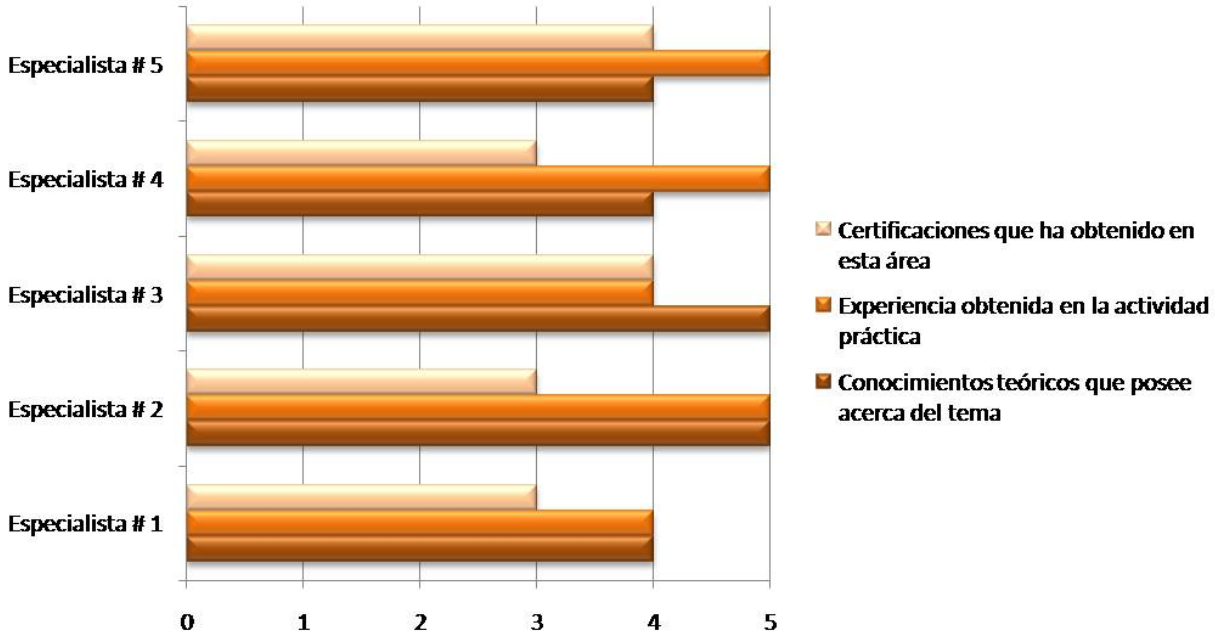

Figura 4 Nivel de Competencia de los especialistas

En la misma se evidencia que en una escala del 1 al 5, el promedio de conocimiento teórico que poseen los especialistas es de un 4,4. En cuanto a la experiencia en la actividad práctica el promedio es de 4,6 y el aspecto con menor resultado es el de certificaciones obtenidas en esta área con un promedio de 3,4. Estos datos demuestran un alto nivel de competencia de los

mismos, lo que da alto valor a sus criterios con respecto a cada uno de las preguntas realizadas en la encuesta.

3.1. Resultados del criterio de especialistas.

En la encuesta aplicada se formulan 9 nueve preguntas dirigidas a valorar las actividades de chequeo de los planes que se generan en el proceso de desarrollo de de software, los objetivos de calidad, las métricas y estándares propuestos para este proyecto, la tareas de revisiones y auditorias, así como las actividades para el chequeo del cumplimiento del Plan de Aseguramiento de la Calidad.

La respuesta a las preguntas han sido graficadas de manera que se puedan comprender mejor los resultados (Figura 4 - 13).

Tabla 1.9 Evaluación por preguntas de los especialistas

	Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
Pregunta # 1	1	1	3		
Pregunta # 2	2		2	1	
Pregunta # 3		3	2		
Pregunta # 4	2	2	1		
Pregunta # 5	1	4			
Pregunta # 6	2	1	2		
Pregunta # 7	2		1	2	
Pregunta # 8	2	1	2		
Pregunta # 9	2	1	2		

En la respuesta a la primera pregunta de la encuesta acerca del proceso de revisión de los planes del proyecto para cada fase del proceso de desarrollo de software los especialistas evaluaron lo siguiente (Figura 5):

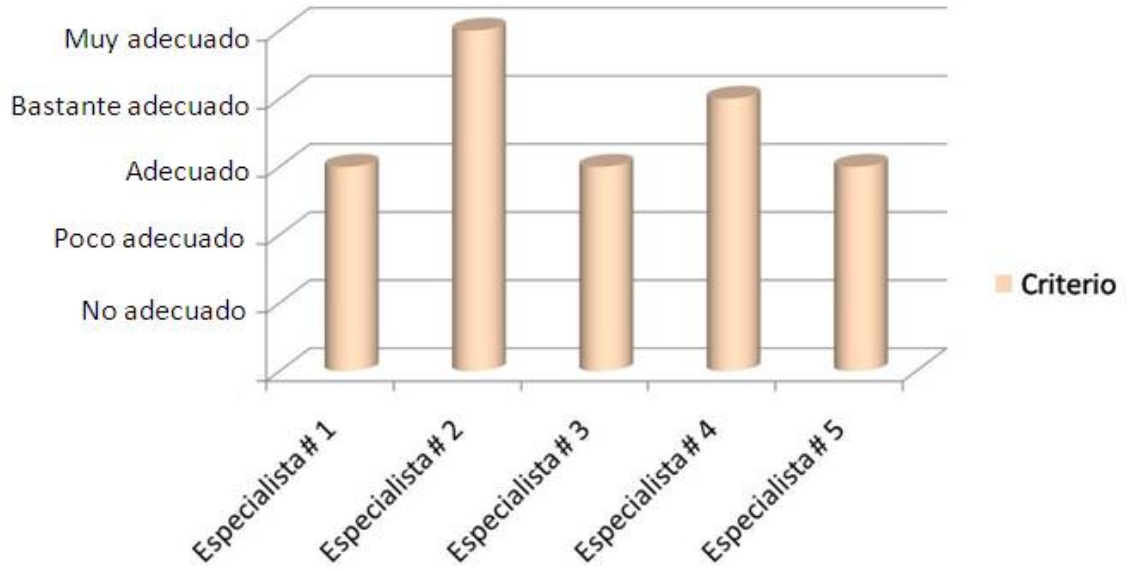

Figura 5 Criterio de evaluación de la pregunta 1 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Adecuado las actividades de revisión de los planes del proyecto que se proponen como se aprecia en la Figura 6.

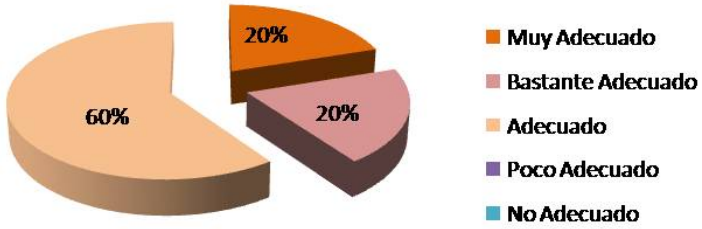

Figura 6 Por ciento del criterio de evaluación de los especialistas para la pregunta 1.

En la respuesta a la segunda pregunta de la encuesta acerca de si los objetivos de Calidad especificados en el Plan de Aseguramiento estaban bien definidos los especialistas evaluaron lo siguiente (Figura 7):

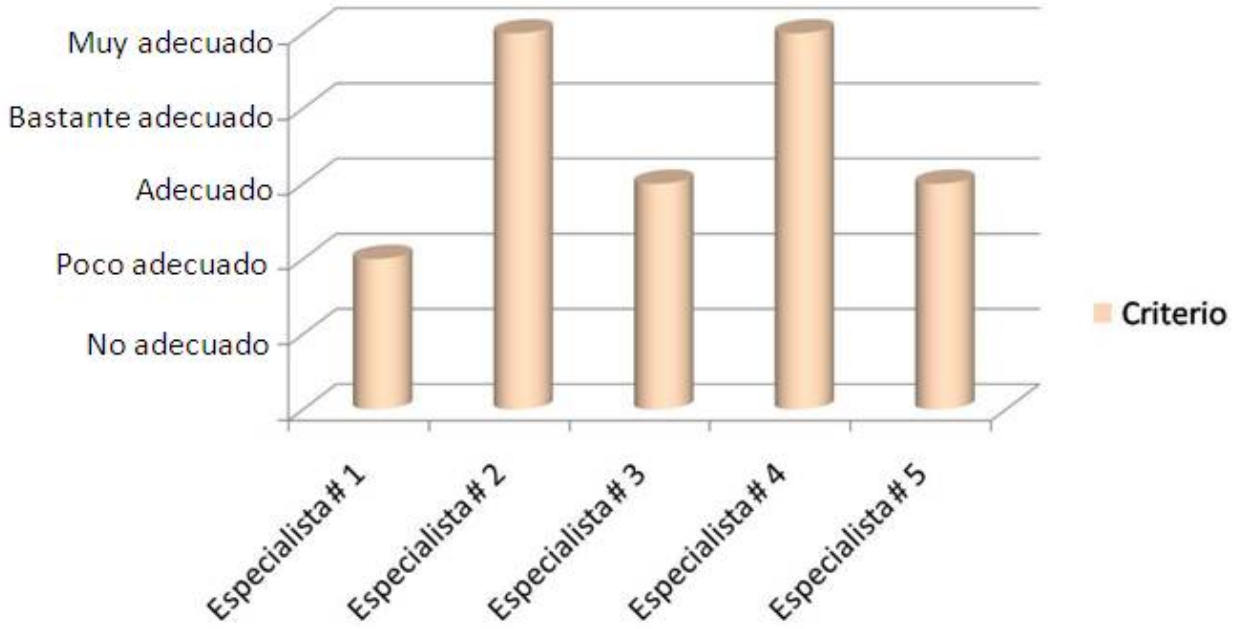

Figura 7 Criterio de evaluación de la pregunta 2 por especialista.

El 80 % de los especialistas valora entre Muy Adecuado y Adecuado que se proponen los objetivos de Calidad especificados en el Plan de Aseguramiento. Con respecto a una valoración de poco adecuado se tuvo en cuenta y fueron tomados en consideración estos criterios en la propuesta realizada (Figura 8).

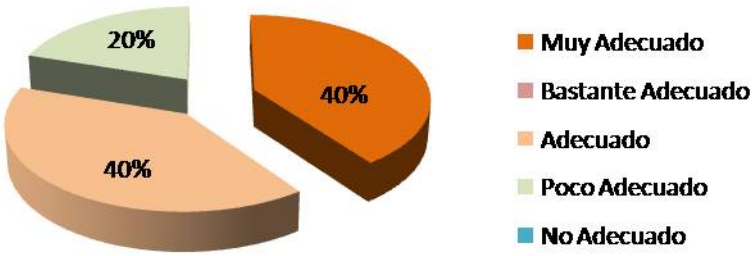

Figura 8 Por ciento del criterio de evaluación de los especialistas para la pregunta 2.

En la respuesta a la tercera pregunta de la encuesta sobre las métricas propuestas para medir la planificación, el control, el seguimiento y el mejoramiento del proceso de desarrollo de software los especialistas evaluaron lo siguiente (Figura 9).

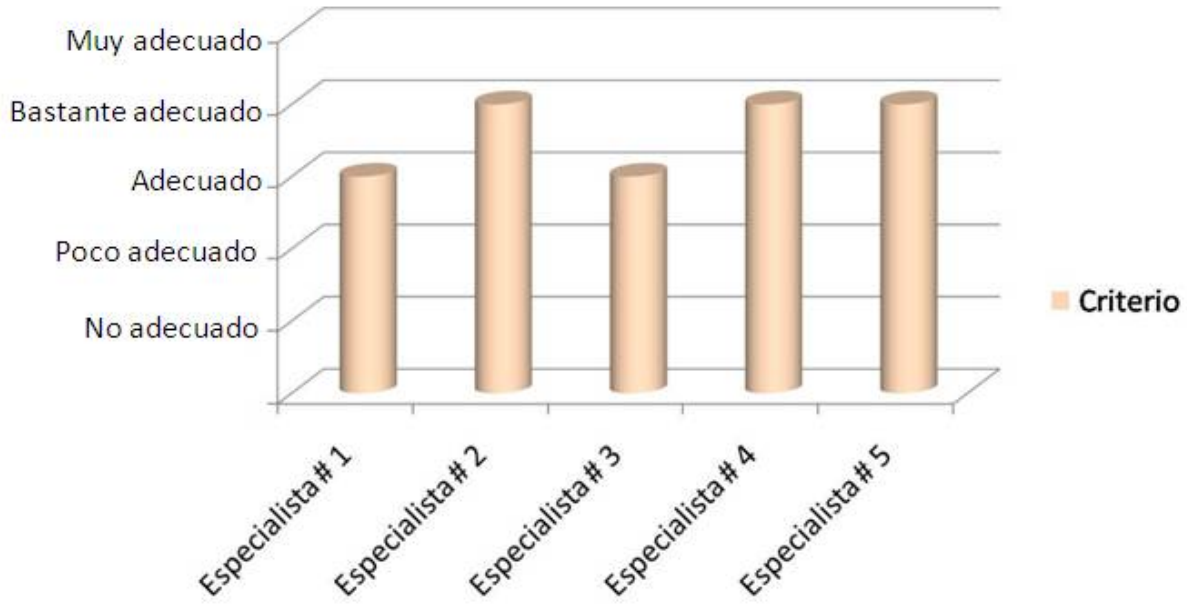

Figura 9 Criterio de evaluación de la pregunta 3 por especialista.

El 100 % de los especialistas valora entre Bastante Adecuado y Adecuado las métricas propuestas para medir la planificación, el control, el seguimiento y el mejoramiento del proceso de desarrollo de software (Figura 10).

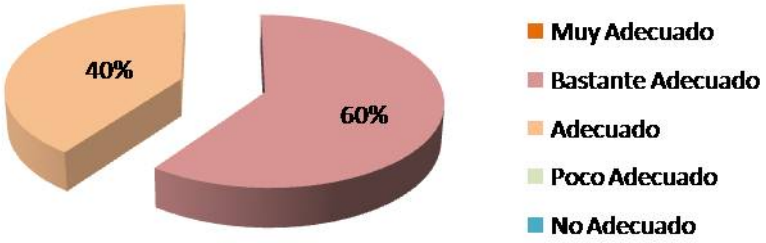

Figura 10 Por ciento del criterio de evaluación de los especialistas para la pregunta 3.

En la respuesta a la cuarta pregunta de la encuesta sobre los estándares y guías propuestos los especialistas evaluaron lo siguiente (Figura 11):

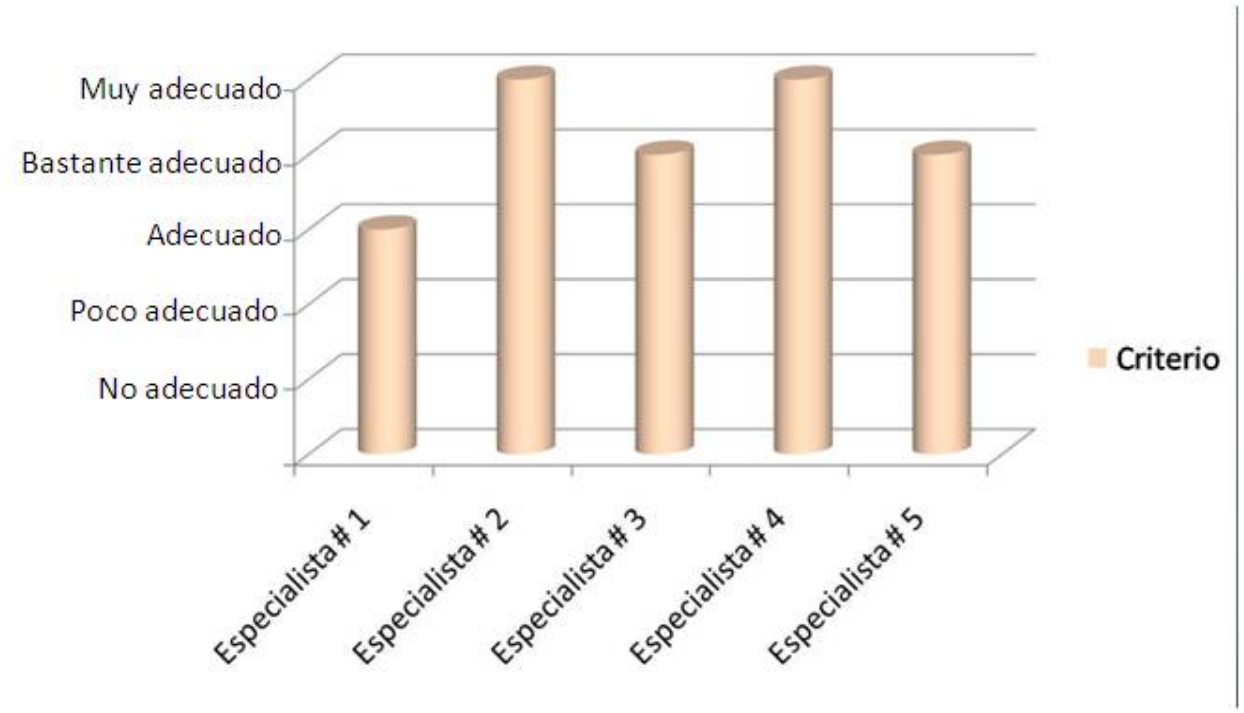

Figura 11 Criterio de evaluación de la pregunta 4 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Adecuado los estándares y guías propuestos. (Figura 12)

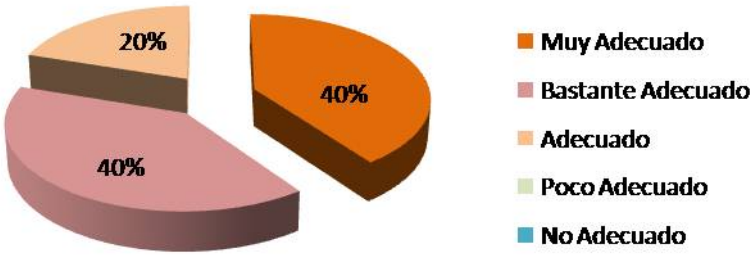

Figura 12 Por ciento del criterio de evaluación de los especialistas para la pregunta 4.

En la respuesta a la quinta pregunta de la encuesta sobre la conveniencia de utilizar las listas de chequeo como herramientas para evaluar los procesos de desarrollo los especialistas evaluaron lo siguiente (Figura 13):

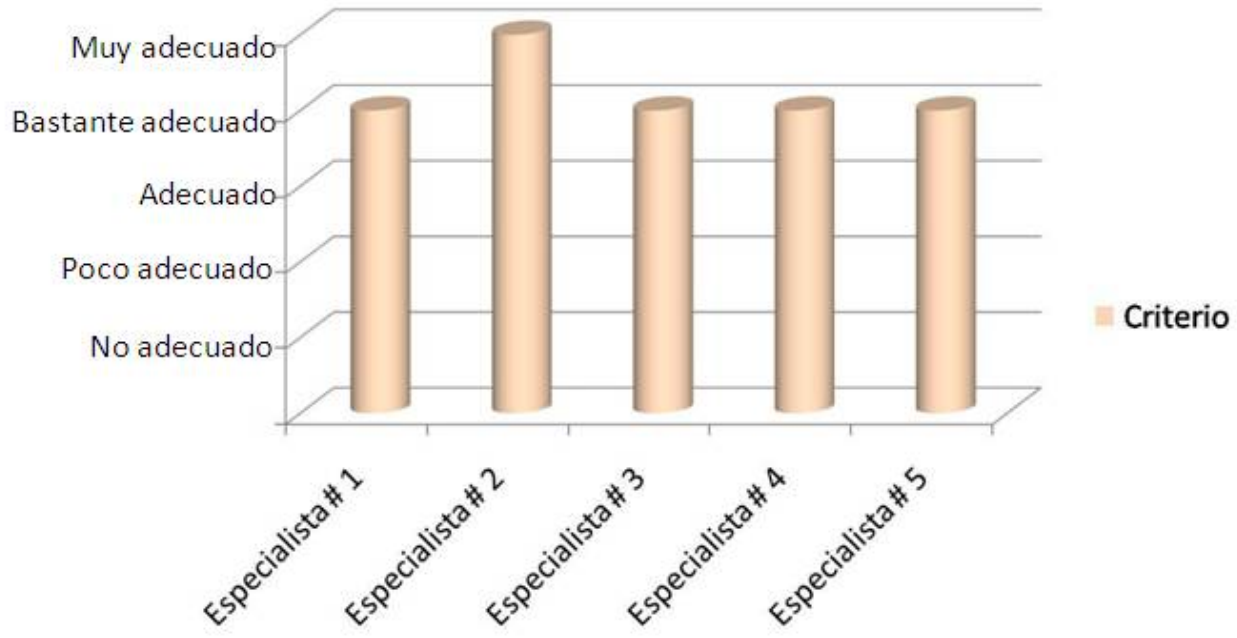

Figura 13 Criterio de evaluación de la pregunta 5 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Bastante Adecuado la utilización de las listas de chequeo como herramientas para evaluar los procesos de desarrollo (Figura 14).

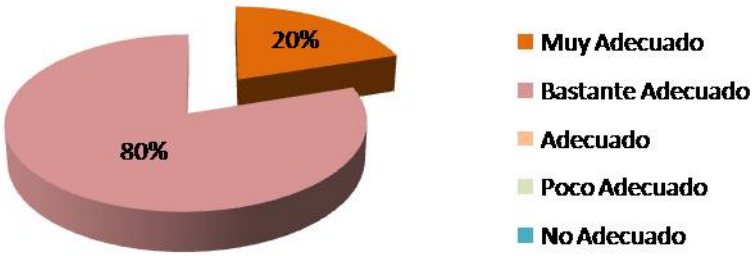

Figura 14 Por ciento del criterio de evaluación de los especialistas para la pregunta 5.

En la respuesta a la sexta pregunta de la encuesta acerca de las tareas de Revisiones y Auditorías propuestas los especialistas evaluaron lo siguiente (Figura 15):

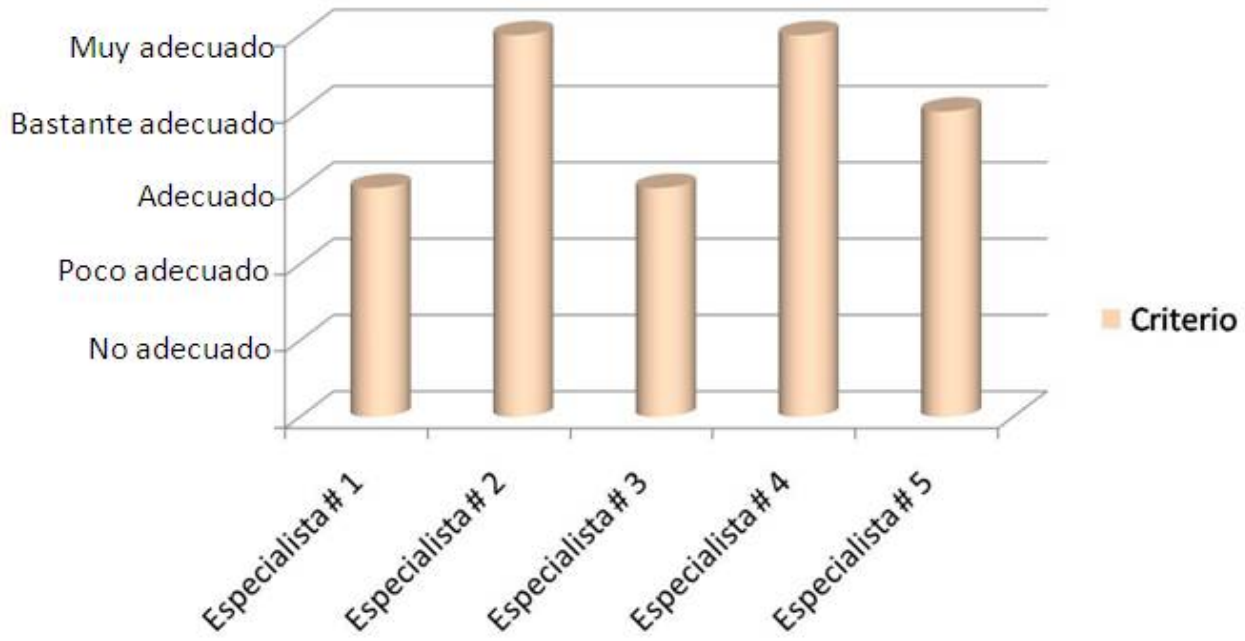

Figura 15 Criterio de evaluación de la pregunta 6 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Adecuado las tareas de Revisiones y Auditorías propuestas (Figura 16).

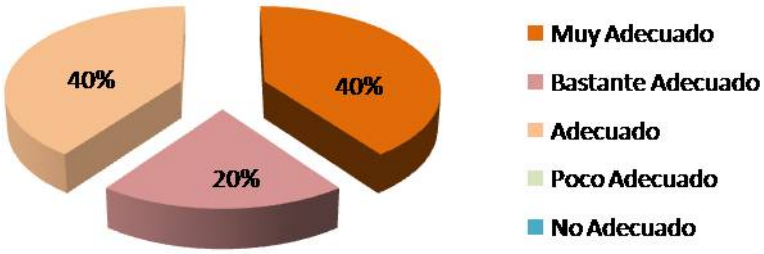

Figura 16 Por ciento del criterio de evaluación de los especialistas para la pregunta 6.

En la respuesta a la séptima pregunta de la encuesta acerca las actividades de Gestión de la Configuración propuestas en el Plan de Aseguramiento los especialistas evaluaron lo siguiente (Figura 17):

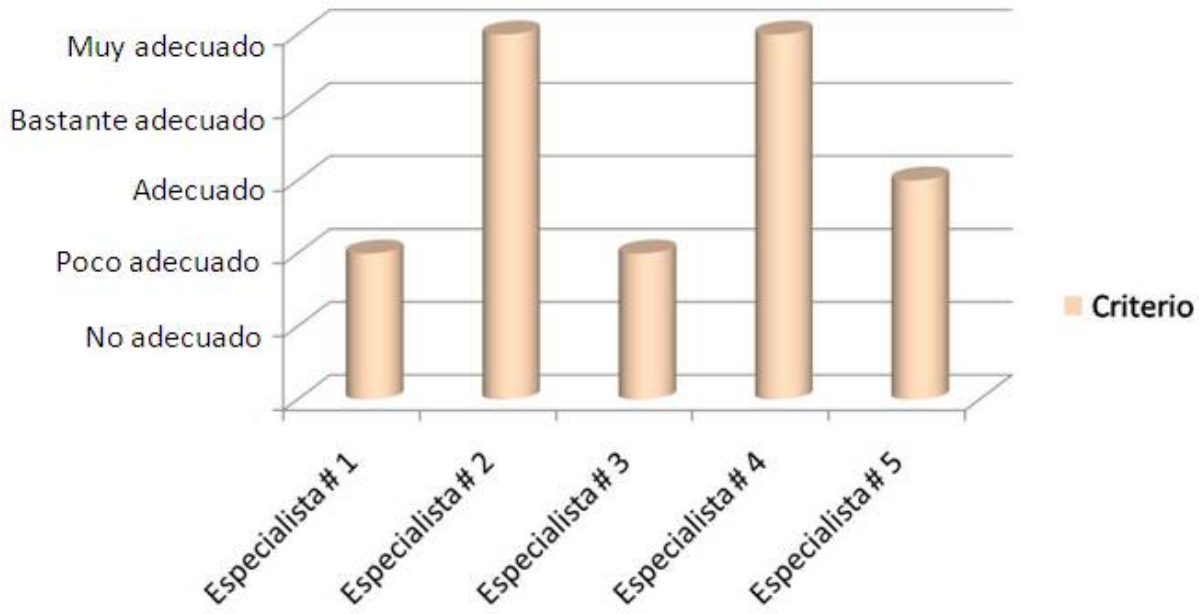

Figura 17 Criterio de evaluación de la pregunta 7 por especialista.

El 60 % de los especialistas valora entre Muy Adecuado y Adecuado las actividades de Gestión de la Configuración propuestas en el Plan de Aseguramiento. Con respecto a dos valoraciones de Poco Adecuado se tuvieron en cuenta y fueron tomados en consideración estos criterios en la propuesta realizada (Figura 18).

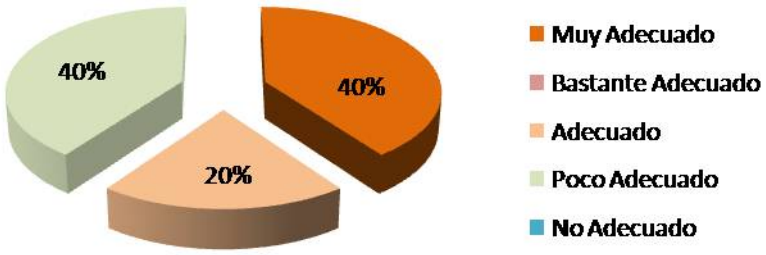

Figura 18 Por ciento del criterio de evaluación de los especialistas para la pregunta 7.

En la respuesta a la octava pregunta de la encuesta acerca de la utilidad que pueda tener la aplicación de este Plan de Aseguramiento de la Calidad de acuerdo a los objetivos trazados los especialistas evaluaron lo siguiente (Figura 19):

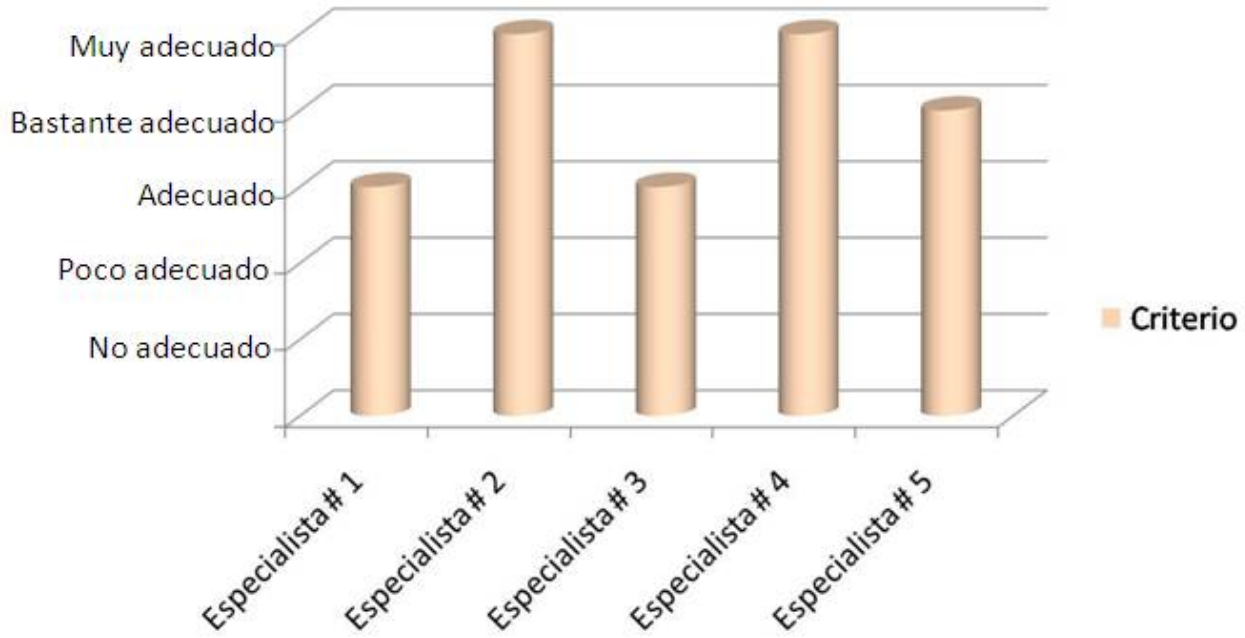

Figura 19 Criterio de evaluación de la pregunta 8 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Adecuado la utilidad que pueda tener la aplicación de este Plan de Aseguramiento de la Calidad de acuerdo a los objetivos trazados (Figura 20).

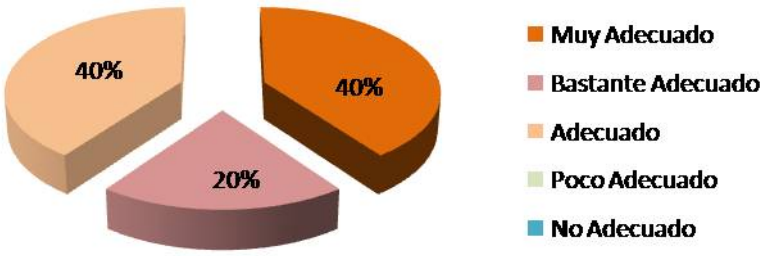

Figura 20 Por ciento del criterio de evaluación de los especialistas para la pregunta 8.

En la respuesta a la novena pregunta de la encuesta acerca de las actividades para el chequeo del cumplimiento del Plan de Aseguramiento de la Calidad propuesto los especialistas evaluaron lo siguiente (Figura 21):

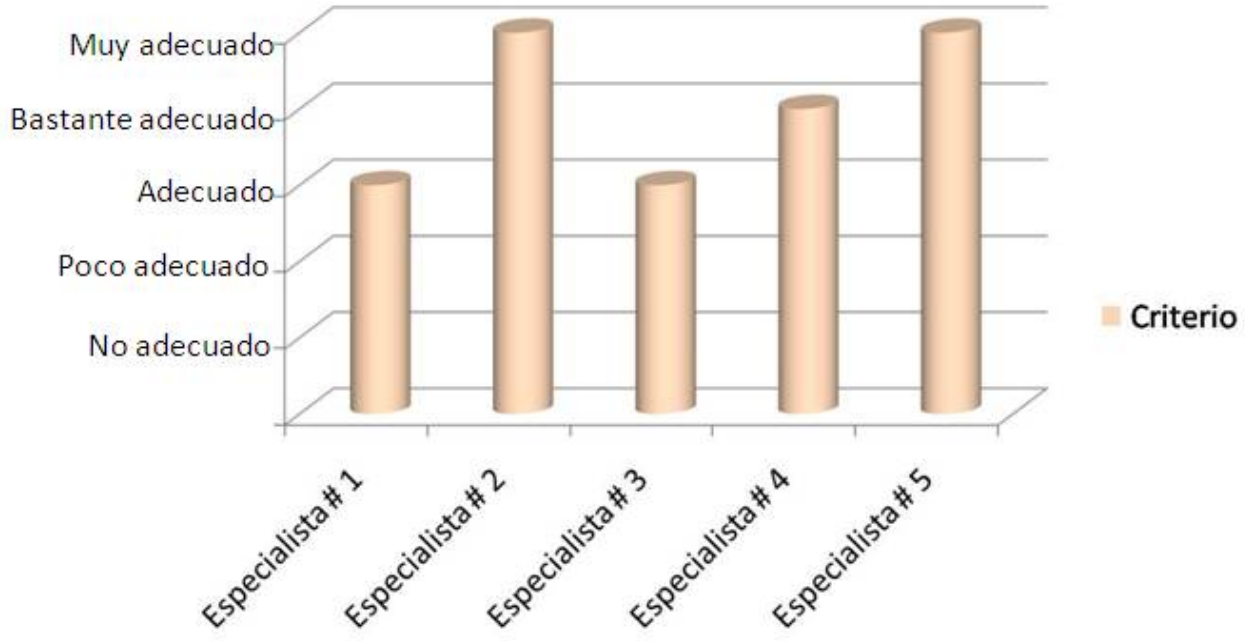

Figura 21 Criterio de evaluación de la pregunta 9 por especialista.

El 100 % de los especialistas valora entre Muy Adecuado y Adecuado las actividades para el chequeo del cumplimiento del Plan de Aseguramiento de la Calidad propuesto (Figura 22).

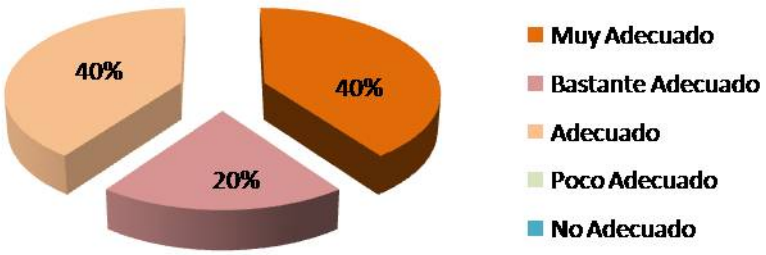

Figura 22 Por ciento del criterio de evaluación de los especialistas para la pregunta 9.

En las siguientes gráficas se aprecia el criterio de cada uno de los especialistas encuestados en los aspectos que se ponen a su consideración a través de las nueve preguntas que se formulan.

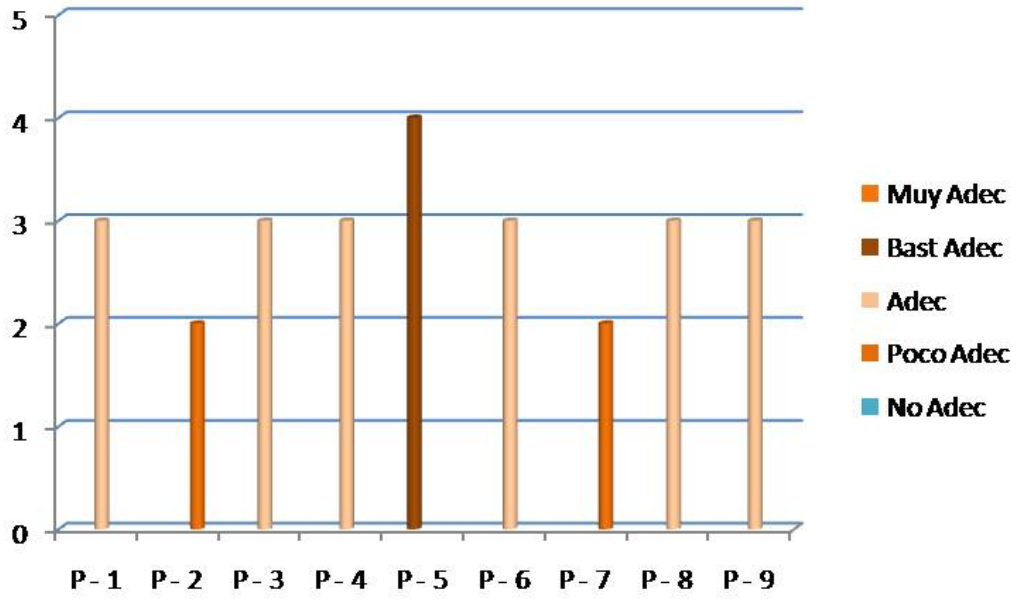

Figura 24 Criterio por preguntas del especialista 1.

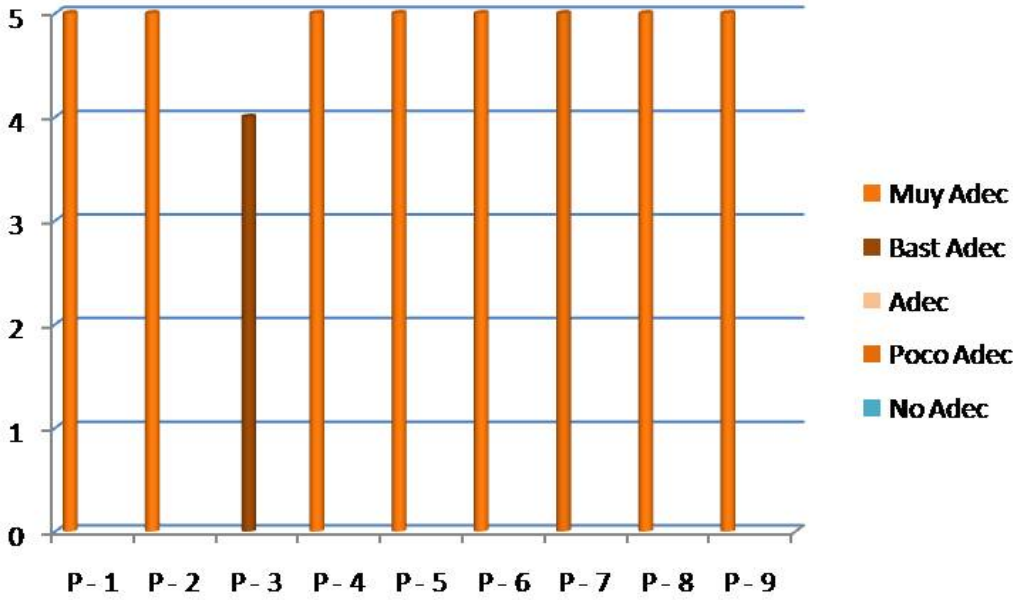

Figura 25 Criterio por preguntas del especialista 2.

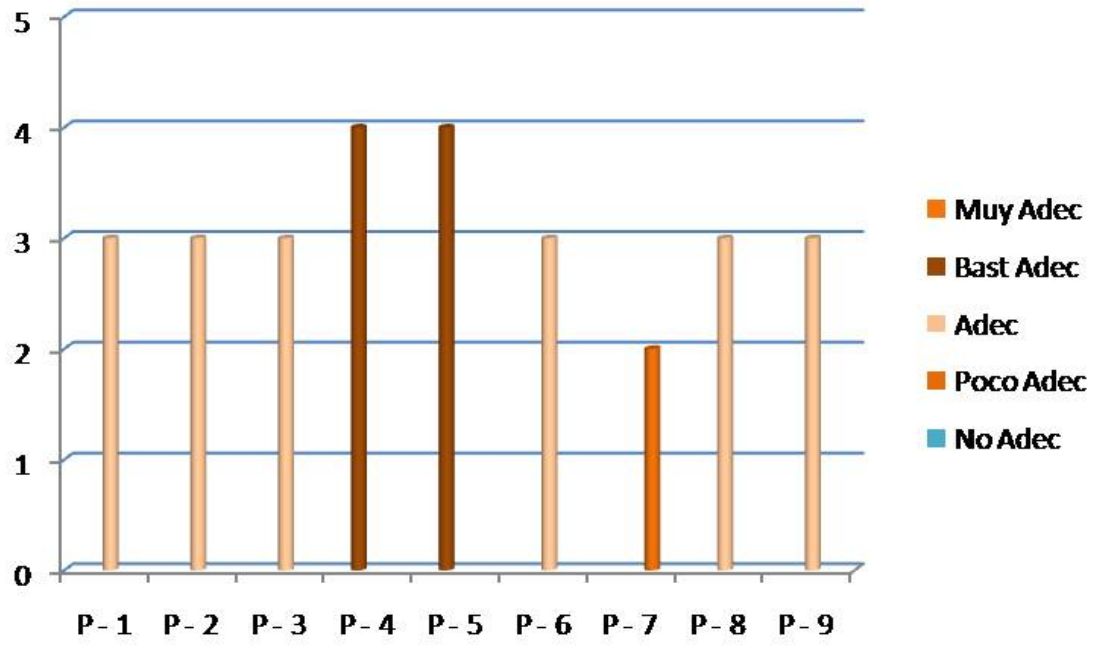

Figura 26 Criterio por preguntas del especialista 3.

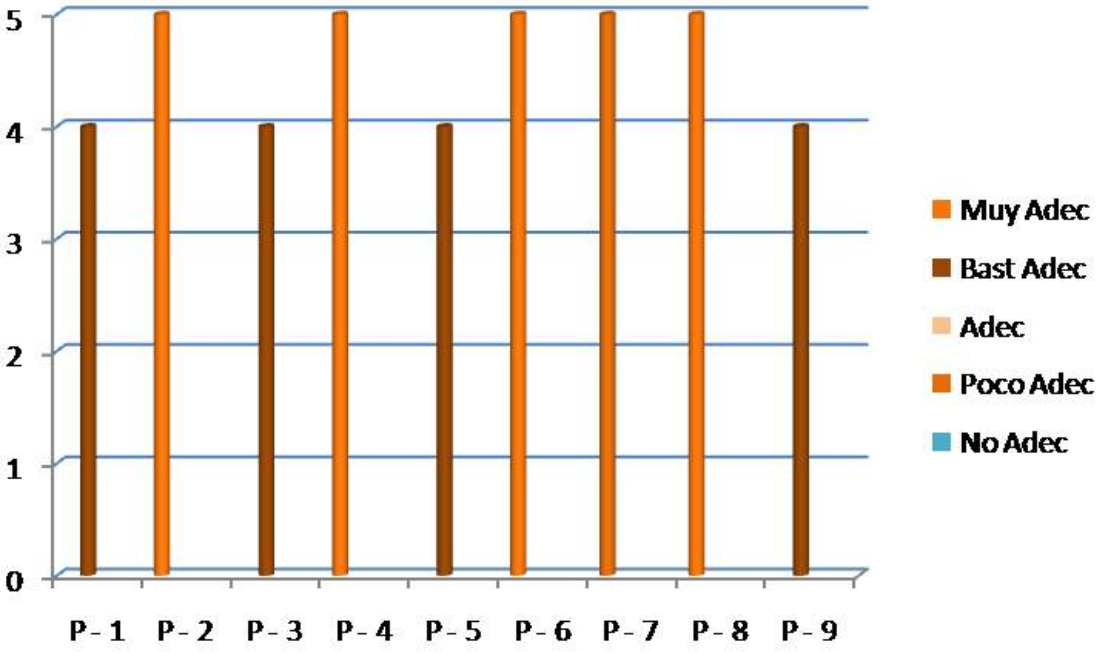

Figura 27 Criterio por preguntas del especialista 4.

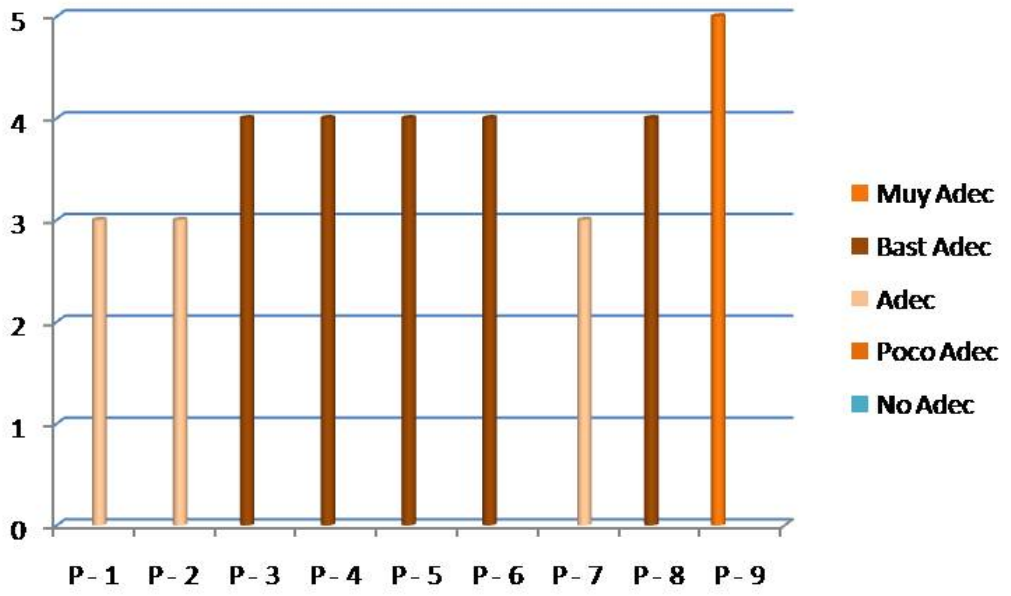

Figura 28 Criterio por preguntas del especialista 5.

De manera general como se observa en la Figura 29 el 93 % de los criterios corresponden a los indicadores que van desde Adecuado hasta Muy Adecuado. No se expresaron criterios de No Adecuado por ninguno de los especialistas encuestados.

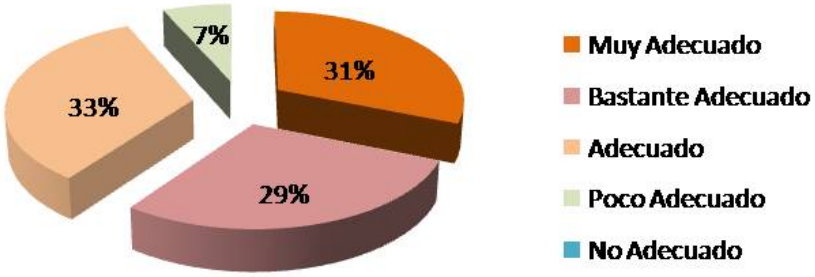

Figura 29 Porciento por criterios de los especialistas

3.2. Conclusiones parciales

- ✓ La aplicación de este instrumento fue muy significativa, pues el 93 por ciento de los criterios se encontraban desde Adecuado hasta Muy Adecuado por cuanto permitió corroborar la idea de que si se aplica apropiadamente esta estrategia se podría obtener un producto con adecuados niveles de calidad.
- ✓ Permitió corregir y perfeccionar el sistema de acciones propuesto, pues todos estos criterios fueron incorporados al cuerpo del trabajo y sirvieron en la elaboración final de la estrategia, la cual quedó satisfactoriamente validada.
- ✓ Se concluyó que los criterios de evaluación, no adecuado, poco adecuado, adecuado, bastante adecuado y muy adecuado muestran el nivel de calidad que puede tener el producto final y con esto dar respuesta a la interrogante de cómo saber cuando se puede determinar un nivel adecuado de calidad.

CONCLUSIONES

Con la realización de este trabajo de diploma se arribó a las siguientes conclusiones:

- ✓ Se determinaron las deficiencias existentes actualmente en los proyectos de la facultad a través del estudio de las debilidades y fortalezas de los mismos.
- ✓ Se logró seleccionar el modelo de calidad a utilizar, así como las métricas y estándares internacionales adecuados para la realización de la estrategia propuesta con el análisis del estado del arte.
- ✓ Se definió un conjunto de actividades de chequeo y revisión para hacer más rigurosa la aplicación correcta del plan de aseguramiento de la calidad.
- ✓ La estrategia propuesta cumple con las demandas del proyecto. La misma fue validada satisfactoriamente por los especialistas, con un 93 % de los criterios desde Adecuado hasta Muy Adecuado.
- ✓ Se desarrolló una estrategia de aseguramiento de la calidad para el proyecto convenio Cuba - Venezuela de acuerdo a las necesidades y características específicas del mismo, proponiéndose una solución factible para resolver los problemas existentes de manera que el producto final sea entregado con niveles de calidad adecuados.

RECOMENDACIONES

Tomando como punto de partida los resultados obtenidos con la realización de este trabajo de diploma, se hacen las siguientes recomendaciones:

- ✓ Aplicar la propuesta realizada para las futuras fases del proyecto Convenio Cuba – Venezuela.
- ✓ Complementar la estrategia para las demás actividades de la Gestión de la Calidad.
- ✓ Capacitar a todo el personal de aseguramiento de la calidad del proyecto Convenio Cuba - Venezuela, en los diferentes aspectos presentes en la estrategia para que puedan dar cumplimiento de manera efectiva a las actividades orientadas en la propuesta.

BIBLIOGRAFÍA

1. **Bedini, M. A. (2007).** Calidad de Software ¿Meta Inalcanzable?

2. **Capability Maturity Model Integration (CMMI), 2006.***Version 1.2 "Improving processes for better products" (*CMU/SEI-2006-TR-008) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

3. **Centro de Investigación Postgrado y Extensión UTPVirtual**. [Online] http://web.unvi.utp.ac.pa/bibliotecavirtual/files/Calidad_en_Inge_858.ppt.

4. **Davis, A.M. 201. 1995.** *Principles of Software Development.* s.l. : MacGraw-Hill, 1995.

5. **2002.** Departamento de Ciencias Matemáticas e Informática. [Online] Enero 2002. http://dmi.uib.es/~bbuades/calidad/calidad.PPT.

6. **Dergarabedian, C. (2008).** *http://tecnologia.infobaeprofesional.com*. Obtenido de http://tecnologia.infobaeprofesional.com/notas/25157-Normas-de-calidad-en-software-una-llave-que-abre-mercados.html

7. **Díaz Pérez, Heney and Delgado Alón, Lídice. Junio de 2007.** *Propuesta de Gestión de Calidad Interna para el proyecto Registro y Notarías [Trabajo de Diploma para optar por el título de Ingeniero Informático].* La Habana : s.n., Junio de 2007.

8. **Dymon M., Kenneth. 1997.** *Una guía de CMM.* 1997.

9. **El Emam, K. y otros (Ed.). 1998.** *SPICE. The Theory and Practice of Software Process Improvement and Capability Determination.* s.l. : IEEE Computer Society., 1998.

10. **Escuela Superior de Ingeniería Informática**. [Online] http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/PPI-t5_1.ppt .

11. **Facultad de Ingenieria Universidad de la República de Uruguay.** [Online] 2008. http://www.fing.edu.uy/inco/cursos/gestsoft/ppts/GS04.PPT.

12. **Febles Estrada, A. (2003)**. *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software [Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas].* Ciudad Habana.

13. **Fernández Carrasco, Oscar M, García León, Delba and Beltrán Benavides, Alfa. 1995.** Biblioteca Virtual de Salud en Cuba. *Biblioteca Virtual de Salud en Cuba.* [Online]

septiembre 1995. [Cited: diciembre 10, 2007.] http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.

14. **Fernández Carrasco, Oscar M., García León, Delba and Beltrán Benavides, Alfa. 1995.** Un enfoque actual sobre la calidad de software. [Online] Septiembre 1995. http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.

15. **Humphrey. W. S. 1989** *Managing the Software Process. SEI Series in Software Engineering.* AddisonWesley. 1989.

16. **Informática, D. d. (2000).** *Control de Calidad en los Sistemas.*

17. **Lovelle, Juan Manuel Cueva. 1999.** GIDIS (Grupo de I+D en Ingeniería de Software). [Online] Octubre 21, 1999. http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF.

18. **Mendoza, G. M. (2007).** Métricas Internas de la Calidad del Producto de Software. Universidad Autónoma de Querétaro.

19. **Moore, J.W. 1998.** *Software Enginnering Standards. A User's Road Map.* s.l. : IEEE Computer Society, 1998.

20. **Moreno García, María N, García Peñalvo, Francisco J and Polo Martín, María José.** Nuevo Portal Público de la Universidad del País Vasco. [Online] http://www.sc.ehu.es/jiwdocoj/remis/docs/mmg-2000.ppt#257,4,1.Introducción).

21. **Navarro, J. 2005.** Escuela de Ingeniería Pontifica Universidad Católica de Chile. [Online] 2005. http://www2.ing.puc.cl/~jnavon/IIC2142/Clases2005/11.IntroTesting.pdf.

22. **Oviedo, Ana Isabel.** LIS (Laboratorio Integrado de Sistemas). [Online] http://siona.udea.edu.co/~aoviedo/Arquitectura%20de%20Software/RTF.htm.

23. **Paulk M.A. 1994.** *A comparison of ISO 9001 and the Capability Maturity Model for Software.* 1994.

24. **Pérez Estévez, I. (2002).** *Métricas para el control de proyectos[Trabajo de Diploma para optar por el Título Ingeniero Informático].*

25. **Pressman, Roger S. 2005.** *Un enfoque práctico.* La Habana : Félix Varela, 2005. Vol.1.

26. **Pressman, Roger S. 1998.** *Un enfoque práctico.* 1998.

27. **Proyecto OEA/GTZ. 1983.** *Manual de Gestión de la Calidad Total a la Medida.* Washington, D.C : s.n., 1983.

28. **Rubio, Gabriel Buades. 2003.** Departamento de Ciencias Matemáticas e Informática. [Online] Mayo 15, 2003. http://dmi.uib.es/~bbuades/calidad/calidad.PPT.

29. **S. H. Kan. 1995.** *Metrics and Models in software Quality Engineering.* s.l. : Addison-Wesley , 1995.

30. **Sánchez, José Luis. 2005.** Consulting, Technology and Outsorcing Services at Accenture. [Online] 2005. http://www.accenture.com/Countries/Spain/About_Accenture/Newsroom/News_Releases/2005/CoritelCMMI.htm.

31. **Scalone, Lic.Fernanda. 2006.** *Estudio comparativo de los modelos y estándares de calidad de software.* Buenos Aires : Facultad Regional Buenos Aires, 2006.

32. **Schulmeyer, Gordon. 1997.** *G. Handbook of Software Quality Assurance.* Prentice Hall : s.n., 1997.

33. **SEI. 2008.** Software Engineering Institute. [Online] 2008. http://www.sei.cmu.edu/cmmi/general/index.html.

34. **SPICE Project File Server 1998.** *"SPICE Phase 2 Trials Interim Report. V 1.00". .* s.l. :

35. **Tecnomaestros. 2008.** Tecnomaestros, Estandares de Calidad ISO para Desarrollo de Software . [Online] 2008. http://tecnomaestros.awardspace.com/estandares_iso.php.

36. **Universidad Rey Juan Carlos. 2007.** Grupo de Investigación KYBELE. [Online] 2007. http://kybele.escet.urjc.es/.

37. **Vásquez Lema, Msc Marcelo. 2007.** La Calidad, el concepto actual. [Online] 2007. http://www.gestiopolis.com/canales8/ger/concepto-de-calidad-y-como-debe-ser-manejado-por-funcionariosy-gerentes.htm.

38. **Zuse H. (1995).** A Framework of Software Measurement. Berlin. Walter de Gruyter.

ANEXOS

Anexos 1

Auditoría realizada a los proyectos productivos de la Facultad 3. Lista de chequeo empleada.

Proceso de Gestión de Software

Proyecto

- ✓ Se realiza la gestión de recursos humanos *(Plantilla Recursos Humanos).*
- ✓ Se realiza la gestión de recursos materiales *(Plantilla Recursos Materiales).*
- ✓ Se obtiene la Visión del Proyecto *(Plantilla Visión del Proyecto).*
- ✓ El equipo conoce la Visión del Proyecto.

Trabajo en equipo

- ✓ Se definen subgrupos de trabajo.
- ✓ Se asignan roles en el equipo.
- ✓ Se emplea el siguiente método para asignar roles en el equipo:
    - Arbitrario.
    - No existe conocimiento previo sobre las habilidades del miembro para desempeñar dicho rol.
    - Existe conocimiento previo sobre las habilidades del miembro para desempeñar dicho rol.
    - Encuesta (Se recogen las principales habilidades de los miembros, y según los resultados se realiza la asignación).
    - Mutuo acuerdo (La asignación se realiza mediante un consenso entre los miembros).
    - Otro método (¿Cuál?).

- ✓ Se realizan reuniones de equipo con la siguiente frecuencia:
  - Diariamente.
  - Semanalmente.
  - Mensualmente.
  - Otra (¿Cuál?).
- ✓ No se realizan reuniones de equipo.
- ✓ Se ventilan las dudas que pueda tener algún integrante del equipo.
- ✓ Se tienen en cuenta los criterios de los miembros del equipo.
- ✓ Se controla el trabajo de cada miembro del equipo *(Plantilla Lista de Tareas).*
- ✓ Se evalúa el resultado del trabajo de cada miembro del equipo.

Planificación

- ✓ Se planifican las fases y las iteraciones (Diagrama de Gantt, Planilla Iteration Plan).
- ✓ Se establecen puntos de chequeo para verificar el estado del proceso (Planilla Evolución del Proyecto).
- ✓ Se establecen objetivos para las fases y las iteraciones (Planilla Iteration Plan).
- ✓ Se realizan revisiones a los artefactos que se van obteniendo durante las iteraciones (Listas de Comprobación y Chequeo).
- ✓ En caso de encontrar errores ¿qué se hace?
- ✓ Se realiza control de tiempo y esfuerzo para los miembros del equipo.
- ✓ Se realiza un balance de la carga de trabajo a los miembro del equipo *(Planilla Balance de Carga).*
- ✓ Se replanifica sobre la base de los datos obtenidos *(Diagrama de Gantt).*

Gestión de Configuración

- ✓ Se planifica la Gestión de Configuración del sistema (Procedimiento Planificación de la GCS, Plantilla Plan de G. de Configuración).

- ✓ Se identifican los elementos de configuración del sistema (Procedimiento Identificación de los ECS).
- ✓ Se realiza Control de Versiones sobre los elementos de configuración (Procedimiento Control de Versiones).
- ✓ Se usa la herramienta Visual Source Safe.
- ✓ Se usa la herramienta Subversion.
- ✓ Se usa otra herramienta (¿Cuál?).
- ✓ Cuando aparece la necesidad de un cambio en el sistema:
  - Se realiza un proceso para tomar una decisión acerca de él (Procedimiento Gestión de los cambios).
  - Se documenta el cambio (Plantilla Solicitud de Cambio).
  - Se le comunica al resto del equipo el cambio.
  - ¿Cómo?
  - Se documenta qué elementos fueron afectados con el cambio (Plantilla Configuration Management).
- ✓ Una vez implementado el cambio se prueba.
  - Se crean líneas base del proyecto (Plantilla Configuration Management).
  - Se tiene control sobre los releases (Plantilla Configuration Management).
  - Se obtienen métricas sobre el estado de la configuración (Plantilla Configuration Management).

Trabajo personal

- ✓ Se registra el tiempo personal de trabajo *(Plantilla Registro de Tiempos).*
- ✓ Se registran los defectos *(Plantilla Registro de Defectos).*
- ✓ Se lleva control de los compromisos *(Plantilla Lista de Compromisos).*

Modelos de calidad

- ✓ Se realiza estudio de los modelos de calidad existentes.
- ✓ Se sigue algún modelo de calidad en específico *(¿Cuál?).*
- ✓ El Proyecto se rige por algún plan de calidad *(Plantilla Plan Aseguramiento de la Calidad).*

Proceso de Ingeniería de Software

Metodología

- ✓ Se dominan las metodologías (Procedimiento Desarrollo de Software).
  - Rational Unified Process (RUP).
  - Extreme Programming (XP).
  - Microsoft Solution Framework (MSF).
  - Métrica 3.0.
  - Otra (¿Cuál?).
- ✓ No se domina ninguna metodología.
- ✓ Se utiliza la siguiente metodología para el desarrollo del proyecto.
  - Rational Unified Process (RUP).
  - Extreme Programming (XP).
  - Microsoft Solution Framework (MSF).
  - Otra (¿Cuál?).
- ✓ No se utiliza ninguna metodología.

Vocabulario

- ✓ Se establece un estándar para la nomenclatura de los artefactos *(Estándares de codificación).*
- ✓ Se establecen plantillas para los documentos *(Plantillas (Todas)).*
- ✓ Se definen los artefactos a utilizar *(Artefactos RUP).*
- ✓ Se crea el Glosario de Términos para el proyecto *(Plantilla Glosario).*
- ✓ Se establecen estándares de código *(Estándares de codificación).*

Requisitos

- ✓ Se describen los requisitos del sistema *(Plantilla Requisitos del Sistema).*
- ✓ Se utilizan metodologías para la captura de requisitos *(Plantillas Requerimientos, Procedimiento Captura de Requisitos).*
- ✓ Se verifican y validan los requisitos *(Plantilla Gestión de Requisitos).*
- ✓ Se construyen prototipos de interfaz.

Arquitectura

- ✓ Se identifican los riesgos (Plantilla Lista de Riesgos).
- ✓ Se gestionan los riesgos (Plantilla Risk Management Plan).
- ✓ Se definen patrones de arquitectura (Patrón de Arquitectura Layer).
- ✓ Se tienen en cuenta los riesgos que puedan derivarse de la arquitectura propuesta.
- ✓ Se priorizan los casos de uso para la arquitectura.
- ✓ Se hacen pruebas para verificar si la arquitectura definida es viable.
- ✓ Se realiza estudio de las posibles herramientas a utilizar en el proyecto.
- ✓ ¿Cuáles herramientas se utilizan para el desarrollo del producto?
  - Visual Studio .NET.
  - Java.
  - Borland C++.
  - Borland Delphi.
  - PHP.
  - Otras (¿Cuáles?).

Métricas

- ✓ Se definen métricas.
- ✓ Sobre el estado de las tareas.
- ✓ Sobre tiempo dedicado a las tareas por parte de los desarrolladores.
- ✓ Sobre los defectos.

Anexos 2

Definición de las métricas para mediciones.

1. Efectividad de eliminar los defectos en una Revisión:

$$EED = \frac{DE_i}{DL} * 100$$

$$DL = DE_i + DEP$$

$$DEP = \sum_{k=i+1}^{n} DE_k$$

DE: cantidad de defectos detectados durante la revisión,

DEP: cantidad de defectos encontrados posterior a la revisión, es decir la cantidad de defecto encontrados en las n-i restantes revisiones que se indican en el plan de revisiones y auditorías del proyecto. También puede calcularse este valor considerando los defectos detectados en las revisiones efectuadas hasta el momento (k = m, con i < m < n) en que se desea analizar la métrica, pero serán resultados parciales que pueden cambiar al finalizar el producto.

DL: cantidad total de defectos presentes en el producto, cuando éste ha sido terminado y se entrega al cliente para su operación.

Si se representan en un gráfico los valores resultantes de la métrica de efectividad se puede analizar qué revisiones de las realizadas al Proyecto de Software resultan poco efectivas y así valorar la posibilidad de incluirla o no en la misma fase del Desarrollo del Proyecto o en otros cuyas características sean similares al analizado o tomar cualquier otra decisión que contribuya al mejoramiento del Proceso de Revisiones.

Ahora bien, esta medida da una vista global de la efectividad de la revisión, pero en ocasiones no basta con esta información y es necesario profundizar para conocer cuáles tipos de errores no han sido detectados y que por tanto conspiran contra la efectividad de dicha Revisión, para ello se propone la métrica siguiente.

2. Efectividad de eliminar los defectos de la fase j en la revisión i.

Permite a los directivos conocer la efectividad de las revisiones en cuanto a la cantidad de defectos que pertenecen a una fase y que son encontrados oportunamente. Por tanto, se puede analizar la cantidad de defectos, por ejemplo de la fase de requisitos, que se han propagado hasta la implantación del sistema o hasta cualquier otra fase de desarrollo del proyecto.

$$EED_{i,j} = \frac{DE_{i,j}}{DL_j} * 100$$

DEi,j: cantidad de defectos detectados durante la revisión i, correspondientes a la fase j.

DLj: cantidad total de defectos presentes en el producto correspondientes a la fase j.

El comportamiento de estas dos primeras métricas puede ser representado en un único modelo que de una idea global de la eficiencia de las Revisiones y de cada una de las fases consideradas en ellas. Por ejemplo, se pudiera plotear los valores de la efectividad de diferentes revisiones realizadas a un proyecto y compararlas entre sí.

3. Densidad de defectos.

$$DD = \frac{DT}{TP}$$

DT: cantidad total de defectos encontrados en el producto.

TP: tamaño del producto, puede ser estimado en líneas de código (en miles) o en puntos de función.

Da una medida de la proporción de defectos del producto con respecto a su tamaño. De la forma en que está definida debe ser evaluada al finalizar el producto y puede ser aprovechada como experiencia para proyectos futuros, no obstante en las etapas tempranas de desarrollo del proyecto esta métrica puede ser utilizada considerando la estimación preliminar de líneas de código o puntos de función que haya sido considerada en la planificación del proyecto.

4. Porcentaje de reinspección (sólo para inspecciones).

$$PRI = \frac{CR}{CI} * 100$$

CR: cantidad de disposiciones de reinspecciones, que son indicadas por el equipo de inspección si entiende que los defectos son demasiados o muy complejos como para considerar satisfactoria la inspección al producto que se realiza. En estos casos se evalúa de mal el producto inspeccionado y se indica realizar nuevamente una revisión cuando hayan sido corregidos los defectos.

CI: cantidad total de inspecciones.

5. Cantidad total de líneas revisadas.

$$TLCR = \sum_{k=1}^{n} LCR_k$$

LCRk: cantidad de líneas de código revisadas en la revisión k.

n: número total de revisiones.

6. Promedio de líneas revisadas en cada revisión.

$$PLCR = \frac{TLCR}{n}$$

7. Productividad de la revisión promedio.

$$PR = \frac{TLCR}{\sum_{k=1}^{n} TDurac_k}$$

TDurack: tiempo de duración de la revisión k.

8. Razón de preparación promedio de los inspectores.

$$RPP = \frac{TLCR}{\sum_{k=1}^{n} \frac{TPrep_k}{TInsp_k}}$$

TPrepk: tiempo de preparación de la revisión k.

TInspk: total de inspectores que participaron en la revisión k.

La Razón de preparación promedio de los inspectores en una revisión es un indicador de la cantidad de líneas que inspeccionó un inspector del total de líneas inspeccionadas durante la fase de preparación, de las revisiones que se le realizaron al proyecto, y puede emplearse para la planificación de futuras revisiones a proyectos con características similares al analizado. La fase de preparación es la fase de detección de defectos.

9. Eficiencia del inspector en la fase de preparación (EIP).

Esta medida permite comparar lo planificado con lo real en cuanto al tiempo de preparación invertido por los inspectores en la fase de preparación. Además se puede hacer análisis sobre el desempeño de los inspectores y saber qué inspectores son los más eficientes.

Esta métrica también puede ser analizada para los proyectos, de forma que se tenga en cuenta para la planificación de proyectos que se desarrollen con posterioridad, en cuyo caso el modelo reflejaría la relación de horas planificadas y reales para cada proyecto.

10. Esfuerzo promedio por líneas de código fuente.

$$ELC = \frac{\sum_{k=1}^{n} Esf_k}{TLCR}$$

$$Esf_k = TPrep_k + TPart_k * TDurac_k + TRetrab_k$$

Esfk: esfuerzo de la revisión k.

TPartk: total de participantes de la revisión k.

TRetrabk: tiempo de retrabajo o de corrección de defectos de la revisión k.

11. Esfuerzo promedio por defectos detectados.

$$EDE = \frac{\sum_{k=1}^{n} Esf_k}{\sum_{k=1}^{n} DE_k}$$

12. Promedio de defectos detectados por líneas de código fuente.

$$DELC = \frac{\sum_{k=1}^{n} DE_k}{TLCR}$$

Da una idea de la cantidad de defectos encontrados en el proyecto, por lo tanto sirve para analizar la efectividad del proceso de revisión y además permite analizar el mejoramiento continuo de la calidad del proceso de desarrollo, pues los proyectos deben tender a tener cero defectos y a esto deben contribuir las Revisiones. Se puede utilizar un modelo similar al utilizado para mostrar la métrica de densidad de defectos.

Anexo 3

Plantilla del Documento de No Conformidades

CCV

Informe de No Conformidades

Versión 1.0

[Documento oficial donde se registrarán todos los errores y/o sugerencias encontradas durante las pruebas o las revisiones efectuadas]

[FORMA DE USO:

- *la casilla No. debe ser llenada forma consecutiva*
- *el Elemento evaluado es el objeto real. Ejemplo: documentación, aplicación, etc.*
- *el Aspecto correspondiente es la parte de ese Elemento evaluado. Ejemplo: formato, redacción, ortografía, contenido; interfaz X, etc.*
- *en la Descripción de la No Conformidad, se profundizará en el aspecto.*
- *en el Tipo se debe definir si es Importante o Sugerencia la no conformidad tratada.*
- *en Responsable de Solución debe hacerse referencia al mismo, así como su firma de conocimiento de la misma.*
- *en Versión se debe referenciar el número de versión que se revisa.]*

<Número de registro>

Fecha:

Etapa de la Detección:

No.	Elemento evaluado	Aspecto correspondiente	Descripción de la No Conformidad	Tipo	Responsable de Solución	Versión

Anexos

[El uso de los anexos es fundamental, ya que cuando se encuentre el error debe tirársele una foto (utilizar el Print Screen) para ayudar a la mejor comprensión del problema.

Estos se asociarán por el No. Correspondiente a la No Conformidad en el pie de la figura insertada]

Anexo 4

Encuesta aplicada a especialistas.

Encuesta a especialistas para someter a sus criterios la propuesta de un Plan de Aseguramiento de la calidad para el proyecto Convenio Cuba – Venezuela de la Facultad # 3.

Estimado(a) Profesor:

La presente encuesta forma parte de la aplicación del Método de Valoración de Especialistas. Con este fin solicitamos su valiosa colaboración, y de antemano le aseguramos, que sus opiniones se tendrán en cuenta para aplicación del Plan de Aseguramiento de la Calidad en este proyecto.

Muchas Gracias.

Nombre y Apellidos: _____

Fecha de graduación: _____ Puesto de trabajo actual: _____

Calificación profesional: Ingeniero___Licenciado en Educación.__Master___Doctor___

Años de experiencia como especialista de calidad: _____

Categoría Docente: Prof. Instructor___ Prof. Asistente____ Prof. Auxiliar____ Prof. Titular____ Prof. Adjunto_____

1. Seleccione en una escala del 1 – 5 el valor que corresponda con el grado de conocimientos que usted posee acerca del tema de investigación que desarrollamos (estrategias de aseguramiento de la calidad en proyectos de software), considerando 1 como no tener ningún conocimiento y 5 el de pleno conocimiento de la problemática tratada.

2. Valore el grado de influencia que cada una de las fuentes que le presentamos a continuación ha tenido en su conocimiento y criterios sobre el tema que se investiga.

FUENTES DE ARGUMENTACIÓN	Grado de Influencia de Cada Fuente		
	ALTO	MEDIO	BAJO
Conocimientos teóricos que posee acerca del tema.			
Su experiencia obtenida en la actividad práctica			
Certificaciones que ha obtenido en esta área			

La propuesta del Plan de Aseguramiento de la Calidad se encuentra adjunta a esta encuesta. Para su análisis y mejor comprensión se le informa que la misma consta de 3 partes fundamentales, la primera parte es sobre las actividades de revisión de los planes de proyecto para cada una de las fases de desarrollo de software, la segunda parte es el plan de Aseguramiento de la calidad propuesto y una tercera con las actividades que se proponen para el chequeo del cumplimiento dicho plan. Se asume la evaluación referenciada en criterios.

3. Le pedimos su criterio sobre las actividades de revisión de los planes del proyecto que se proponen en cada fase del proceso de desarrollo de software.

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Proceso de revisión de los planes del proyecto que se proponen en cada fase del proceso de desarrollo de software.					

4. Le pedimos su criterio para ver si los objetivos de Calidad especificados en el Plan de Aseguramiento están bien definidos

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Los objetivos de Calidad especificados en el Plan de Aseguramiento son:					

5. Le pedimos su criterio sobre las métricas propuestas para medir la planificación, el control, el seguimiento y el mejoramiento del proceso de desarrollo de software.

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Las métricas propuestas para medir la planificación, el control, el seguimiento y el mejoramiento del proceso de desarrollo de software son:					

6. Le pedimos su criterio sobre los estándares y guías propuestos

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Los estándares y guías propuestos son:					

7. Le pedimos su criterio sobre si es conveniente utilizar las listas de chequeo como herramientas para evaluar los procesos de desarrollo

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Las listas de chequeo como herramientas para evaluar los procesos de desarrollo son:					

8. Le pedimos su criterio sobre las tareas de Revisiones y Auditorías propuestas.

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Las tareas de Revisiones y Auditorías son:					

9. Le pedimos su criterio sobre las actividades de Gestión de la Configuración propuestas en el Plan de Aseguramiento.

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Las actividades de Gestión de la Configuración propuestas en el Plan de Aseguramiento son:					

10. Le pedimos su criterio en cuanto a la utilidad que pueda tener la aplicación de este Plan de Aseguramiento de la Calidad de acuerdo a los objetivos trazados

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
La utilidad que puede tener la aplicación del Plan de Aseguramiento de la Calidad es:					

11. Le pedimos su criterio sobre las actividades para el chequeo del cumplimiento del Plan de Aseguramiento de la Calidad propuesto.

	CRITERIO DE ESPECIALISTAS				
Pregunta	C1 muy adecuado	C2 bastante adecuado	C3 adecuado	C4 poco adecuado	C5 no adecuado
Las actividades para el chequeo del cumplimiento del Plan de Aseguramiento de la Calidad propuesto son:					

Muchas gracias por su gentil colaboración.

Mayo de 2008.

GLOSARIO

Estrategia: Un curso de acción conscientemente deseado y determinado de forma anticipada, con la finalidad de asegurar el logro de los objetivos de la empresa. Normalmente se recoge de forma explícita en documentos formales conocidos como planes.

"Las estrategias son programas generales de acción que llevan consigo compromisos de énfasis y recursos para poner en práctica una misión básica. Son patrones de objetivos, los cuales se han concebido e iniciado de tal manera, con el propósito de darle a la organización una dirección unificada". H. Koontz. Estrategia, planificación y control (1991).

Plan: Se ha definido como un documento en que se constan las cosas que se pretenden hacer y forma en que se piensa llevarlas a cabo.

Documento que contempla en forma ordenada y coherente las metas, estrategias, políticas, directrices y tácticas en tiempo y espacio, así como los instrumentos, mecanismos y acciones que se utilizarán para llegar a los fines deseados. Un plan es un instrumento dinámico sujeto a modificaciones en sus componentes en función de la evaluación periódica de sus resultados.

Cliente: Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos.

Defecto: Cualquier requerimiento, elemento de diseño o de implementación que si no es cambiado, causará un diseño, implementación, prueba, uso, o mantenimiento inapropiado del producto.

ISO: (International Organization for Standardization) Es la Organización Internacional para la Normalización; responsable para la normalización a escala mundial. ISO está formado por distintos comités técnicos, cada uno de los cuales es responsable de la normalización para cada área de especialidad. El propósito de ISO es promover el desarrollo de la normalización para fomentar a nivel internacional el intercambio de bienes y servicios y para el desarrollo de la cooperación en actividades económicas, intelectuales, científicas y tecnológicas. El resultado del trabajo técnico dentro de ISO se publica en forma final como normas internacionales.

IEEE: (Institute of Electrical and Electronics Engineers) Asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos,

ingenieros en electrónica, científicos de la computación e ingenieros en telecomunicación. Se creó en 1884.

Madurez: (Maturity, ISO 9126) Subcaracterística de fiabilidad, que indica la frecuencia con que ocurren los fallos.

Métrica: Medida o medición. La continúa aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa significativa y oportuna, junto con el uso de esas técnicas para mejorar el proceso y sus productos.

Proceso: (ISO-15504) Proceso de Software, es el proceso (o procesos), usado por una organización (o proyecto), para planificar, administrar, ejecutar, monitorear, controlar y mejorar sus actividades, relacionadas con el software.

Producto de Software: (IEEE-12207) Es el conjunto de programas de computadora, procedimientos, documentación y datos, asociados.

Revisión: Reuniones de un grupo definido de personas cuyo objetivo es encontrar errores en un artefacto de software. Con revisiones para testear requisitos, diseño, planes, manuales y software Participantes de los revisiones son: los autores que han escrito el artefacto; los revisores que tienen que detectar errores; el secretario que documenta los errores encontrados; el presentador que expone/explica el artefacto bajo testeo; el líder que dirige la reunión, elige la fecha para la reunión y invita a los participantes.

SEI: (Software Engineering Institute) Es un instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso Estadounidense en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, Es un referente en Ingeniería de Software por realizar el desarrollo del modelo SW-CMM (1991) que ha sido el punto de arranque de todos los que han ido formando parte del modelo que ha desarrollado sobre el concepto de capacidad y madurez, hasta el actual CMMI.