Universidad de las Ciencias Informáticas
Facultad 3

Título: Propuesta de un Procedimiento para el Aseguramiento de la Calidad en los Proyectos Productivos de la Facultad 3.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autoras: Yamira Medel Filtres.

Ana Gladys Sánchez Ramos.

Tutora: Ing. Yenisleidy Rendón Vigil.

Junio 2008

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 _____.

Yamira Medel Viltres	Ana Gladys Sánchez Ramos	Yenisleidy Rendón Vigil
_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor

"La base de un desarrollo impetuoso en los años futuros debe basarse en una ciencia cada vez más desarrollada"

Fidel Castro Ruz

DATOS DE CONTACTO

Ing. Yenisleidy Rendon Vigil

Especialidad de graduación: Ingeniería Industrial
Categoría docente: Instructor
Años de graduado: 1
Correo: yenisleidyrv@uci cu

AGRADECIMIENTOS

A la Revolución, a Fidel y a la UCI, por dejarnos formar parte de este proyecto futuro.

A nuestra tutora Yenisleidy Rendón Vigil por habernos ayudado durante el transcurso de nuestra tesis.

A nuestros compañeros de aula que compartieron los cinco años de carrera junto a nosotras y nos apoyaron para terminar la carrera.

Agradecemos a nuestros padres por confiar y darnos apoyo en todo momento.

A nuestras familias por estar a nuestro lado cada vez que los necesitamos.

A todos nuestros profesores, por contribuir a nuestra formación profesional.

A todas aquellas personas que nos brindaron su ayuda cuando los necesitamos.

DEDICATORIA

Les dedico este trabajo con todo el amor del mundo a mi madre Isabel y a mi padre Manuel por estar siempre a mi lado y darme el apoyo necesario para seguir adelante aún en los momentos más difíciles.

A mi novio Juan Carlos por haberme dado todo su amor, todo su apoyo y por todos los momentos lindos que me ha regalado.

A mis queridos abuelos por estar siempre pendiente de mí y regalarme tanto cariño.

A mi amigas Yanisleidys lara y Yaima Ceballo por ser mis mejores amigas y que siempre han estado presentes en las buenas y en las malas para ayudarme.

A mi tía Deisy Sánchez por darme su apoyo incondicional cada vez que lo necesitaba.

A mis hermanos por estar siempre a mi lado y creer en mí.

A la familia de mi novio por ser tan generosos y cariñosos conmigo.

A mis compañeras de apartamento por ser mi familia aquí: Yanet Rábago, Yamira, Katia, Yanet Álvarez, Mare, Kenia, Annerys.

A mi compañera de tesis Yamira por su esfuerzo en la realización de este trabajo y por aguantarme durante estos cinco años.

A todos los que contribuyeron a que hoy esté haciendo estos agradecimientos, que no son más que la memoria de mi corazón.

Ana Gladys

DEDICATORIA

A mis padres por apoyarme y estar siempre a mi lado en todo momento porque sin ellos este triunfo no se hubiera realizado.

A mi esposo, mi compañero de carrera por su apoyo y comprensión en estos cinco años.

A mi hermana por ayudarme y apoyarme siempre.

A mis abuelos, Eva, Ana, Robert y Fulgencio por quererme tanto.

A mi abuelo Erasmo por creer siempre en mí.

A mi familia por sus consejos y brindarme todo su apoyo.

A la familia de mi esposo por brindarme su ayuda cada vez que la necesitaba.

A mis amigas y amigos por ayudarme y seguir a mi lado durante todo este tiempo.

A mi tía Naty y a su esposo por todo su amor y su apoyo.

A mis compañeras de cuarto durante toda la carrera: Yanet Rábago, Ana Gladys, Katia, Yanet Álvarez, Mare, Kenia, Annerys.

A mis amistades, compañeros de grupo y profesores que de una forma u otra me han ayudado a lo largo de mi carrera y en la realización de este trabajo.

A mi eterna compañera de tesis Ana Gladys por su empeño y dedicación en este trabajo.

A nosotras.

Yamira

RESUMEN

El presente trabajo se realizó en la Facultad 3 de la Universidad de las Ciencias Informáticas, con el objetivo de elaborar una propuesta de un procedimiento para el aseguramiento de la calidad en los proyectos productivos de la facultad.

En este trabajo se realizó la fundamentación teórica con el objetivo de estudiar el estado actual de la calidad del software, para ello se recogen algunos conceptos emitidos por diferentes autores, entre los que están: calidad, procedimiento, proceso, etc. Se realizó una caracterización sobre el proceso de producción de la Facultad 3 para después realizar un diagnóstico a los proyectos para determinar la situación actual en cuanto al aseguramiento de la calidad y para ello se emplearon diferentes técnicas como las entrevistas y encuestas.

Para cumplir el objetivo de esta investigación se elaboró la propuesta de solución, dicha propuesta tiene como objetivo asegurar la calidad en la producción de software de la Facultad 3 y se realizó basada en la metodología RUP por ser la metodología más utilizada en los proyectos productivos de la facultad.

El procedimiento propuesto en esta investigación, favorecerá el trabajo y el incremento de la eficiencia en la Facultad 3 en cuanto a proyectos de software se refiere. Con ella el grupo de trabajo de calidad de cada proyecto tendrá un procedimiento a seguir, y una forma más organizada de realizar el trabajo.

PALABRAS CLAVE: Aseguramiento de la calidad, calidad del software, proceso de producción, procedimiento.

TABLA DE CONTENIDOS

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 4

Introducción 4
1.1. Definición de conceptos fundamentales 4
1.2. Calidad del software 6
1.3. Factores que determinan la calidad del SW 7
1.4. Métricas de software 9
1.5. Gestión de la calidad del software 12
1.5.1. Planificación de la calidad del software 13
1.5.2. Control de la calidad del software 14
1.5.3. Aseguramiento de la calidad del software 16
1.5.4. Mejora de la calidad del software 18
1.6. Modelos y estándares existentes para la administración de la calidad 18
1.7. Calidad del software en Cuba. 24
1.7.1. Calidad del software en La Universidad de las Ciencias informáticas 26
Lineamientos mínimos de calidad definidos por la UCI 27
Conclusiones del capítulo 1 30
CAPÍTULO 2: CARACTERIZACIÓN Y DIAGNÓSTICO A LA PRODUCCIÓN DE SOFTWARE DE LA FACULTAD 3. 31

Introducción 31
2.1. Grupo de control y aseguramiento de la calidad en la facultad 3. 31
2.2. Aplicación de técnicas para el diagnóstico 32
2.2.1. Muestreo probabilístico 34
2.3. Diagnóstico aplicado a los proyectos productivos de la Facultad 3. 37
Conclusiones del capítulo 2 39
CAPITULO 3: PROPUESTA DE UN PROCEDIMIENTO PARA EL ASEGURAMIENTO DE LA CALIDAD EN LOS PROYECTOS PRODUCTIVOS DE LA FACULTAD 3. 40

3.1. Propuesta de un procedimiento de SQA 40
3.2. Etapas propuestas para el aseguramiento de la calidad. 44
3.2.1. Capacitación del grupo SQA del proyecto. 44
3.2.2. Planificación de la calidad 45
3.2.3. Control de la calidad 55
3.2.4. Evaluación de la calidad del producto 57
3.3. Cumplimiento de los lineamientos mínimos de calidad definidos por la universidad. 58
Conclusiones del capítulo 3 59
CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA 60

Introducción 60
4.1. Método de evaluación de expertos 60
4.2. Selección de los expertos 61
4.2.1. Determinación de la cantidad de expertos 61
4.3. Elaboración del cuestionario 61
4.4. Resultados de la evaluación 61

Conclusiones del capítulo 4 64
CONCLUSIONES GENERALES 65

RECOMENDACIONES 66

REFERENCIA BIBLIOGRAFICA 67

BIBLIOGRAFÍA 68

ANEXOS 70

INTRODUCCIÓN

Hoy en día el papel de la calidad se ha elevado considerablemente si se tiene en cuenta la tendencia cada vez más creciente de la globalización de la economía y el comercio internacional, donde el consumidor pasa a ser el principal sujeto de todo cuanto se hace. Además se debe suministrar productos y servicios a un costo que el cliente esté dispuesto a pagar y para lograr este objetivo se tiene que reducir al mínimo el número de errores y correcciones que se pueda tener en los procesos tanto productivos como administrativos. La palabra calidad siempre está presente en la vida diaria ya que todo lo que se quiere tener se exige con buena calidad. Si se quiere comprar un artículo para los hogares, consumir algún alimento o vestirse bien solamente se tiene que buscar la calidad. Se debe asegurar que en el mundo competitivo de las empresas de hoy el proceso esté dirigido a mejorar el nivel de satisfacción del cliente como resultado de las actividades. Sin dudas con la globalización de los mercados uno de los factores trascendentales para que una empresa tenga éxito es la calidad de sus productos ó servicios. Últimamente existe una tendencia mundial por parte de los clientes hacia requisitos más exigentes respecto a la calidad donde se está produciendo una creciente toma de conciencia de que es necesario mejorarla de forma sistemática para obtener un buen rendimiento económico.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En los pasados 50 años ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por sí misma [1]. Sin embargo nada de esto sería posible sin un factor muy importante dentro del proceso de desarrollo, llamado calidad del software.

El objetivo del aseguramiento de la calidad del software es brindar una protección a cada paso del proceso de desarrollo del software de forma tal que lo que se obtiene es la madurez de la ingeniería del software [1]. La ingeniería del software hoy en día es relativamente reciente y muchos de sus conceptos importantes están aún inmaduros.

En Cuba el desarrollo del software es aún incipiente, es por ello que ha ido creciendo el interés por parte del gobierno cubano de realizar importantes tareas como el desarrollo de la industria del software con el fin de elevar la producción y calidad del software cubano. Por eso surge la Universidad de las Ciencias Informáticas (UCI) la cual es la primera universidad surgida al calor de la Batalla de Ideas, sobre la base del nuevo concepto de universidad productiva, logrando una fuerte vinculación Universidad-Empresa donde los estudiantes y profesores deben estar vinculados a un proyecto productivo. La UCI pretende convertir la industria del software en un renglón fundamental de la

economía e insertarse en el mercado internacional así como llevar la informatización a todos los sectores de la sociedad tales como: Turismo, Educación, Salud, Deporte, etc.

Actualmente en la Facultad 3 de la Universidad de las Ciencias Informáticas se llevan a cabo una serie de proyectos donde se viene evidenciado problemas en cuanto a la calidad del software que se produce. Esto es debido a que existe dificultades tales como la mala captura de requisitos lo que trae consigo que no se cumpla con las especificaciones que se requiere desarrollar. No se hace una buena definición de los roles por lo que no existe un compromiso en tiempo y esfuerzo de todos los miembros del equipo. En ocasiones los errores relacionados con el software no se detectan a tiempo, esto puede traer inconformidades por parte de los clientes después de entregarse el software. Otro de los problemas es que no existe una planificación bien definida haciendo que los tiempos de desarrollo sean mayores que los planeados. Unido a esto está la poca experiencia de la mayoría de los profesores y estudiantes que integran los proyectos productivos de la facultad lo cual atenta contra la calidad de la producción de software, y del producto en si mismo.

Esta situación pone de manifiesto el siguiente **problema** que se pretende resolver: ¿Cómo asegurar la calidad del software en los proyectos productivos de la Facultad 3? De aquí que el **objeto de estudio** lo constituya el proceso de desarrollo del software y el **Campo de acción** el proceso de aseguramiento de la calidad en los proyectos productivos de la Facultad 3. Todo ello encaminado a cumplir el **objetivo general** de elaborar una propuesta de un procedimiento para asegurar la calidad del software en los proyectos productivos de la Facultad 3. Basado en el problema y como vía para alcanzar el objetivo propuesto, se formuló la siguiente **hipótesis:** la creación de un procedimiento para el aseguramiento de la calidad permitirá mejorar considerablemente la calidad en la producción de software de los proyectos de la Facultad 3, lo que a su vez contribuirá a elevar la calidad del producto final.

Para darle cumplimiento al objetivo general se trazan las siguientes **tareas de investigación**:

- Definición de la investigación.
- Elaboración del marco teórico de la investigación.
    - Búsqueda de bibliografía acerca de los principales temas y conceptos relacionados.
- Realización de un diagnóstico para determinar la situación actual.
    - Realización de encuestas y/o entrevistas a las personas relacionadas con el tema.
    - Realización de un estudio acerca de la producción de software en la facultad 3.
    - Análisis de los resultados obtenidos.
- Elaboración de la propuesta de solución.
- Validación de la propuesta.

- Redacción del informe final.

Para llevar a cabo la realización de este trabajo se utilizaron diferentes **métodos de trabajo científico**. Como método teórico se empleó el **Analítico-Sintético** analizando las teorías, documentos, etc., para extraer los elementos más importantes que se relacionan con el objeto de estudio. Además se utilizó el **Histórico-Lógico** ya que ayuda a comprobar teóricamente datos históricos que han ido evolucionando en un período de tiempo. A partir de la hipótesis se llega a nuevos conocimientos y predicciones a partir del conocimiento que se posee y para ello fue de gran importancia la utilización del método **Hipotético-deductivo**. De igual manera fue de gran importancia la utilización de métodos empíricos tales como la **entrevista** y la **encuesta** donde se debe realizar una preparación previa, sobre el tema a tratar, y elaborar una guía para su desarrollo partiendo de la necesidad de obtener información, ideas, experiencias que permitan entender las características propias del proceso y aporten conocimientos específicos del tema.

El resultado de toda investigación es el objetivo final de la misma. De los **aportes prácticos** de este trabajo se espera obtener un procedimiento para asegurar la calidad del software en los proyectos productivos de la Facultad 3 que permita mejorar el nivel de satisfacción del cliente.

El trabajo está compuesto por cuatro capítulos.

Capítulo 1: en este capítulo se definen algunos conceptos que son necesarios para un mejor entendimiento y desarrollo del trabajo, así como la importancia de la calidad del software, modelos y estándares de calidad en un proyecto de software y métricas de calidad del software.

Capítulo 2: se hace una caracterización sobre la producción de software en la facultad 3 para después realizar un diagnóstico a los proyectos para determinar la situación actual en cuanto al aseguramiento de la calidad y para ello se emplearon diferentes técnicas como las entrevistas y encuestas.

Capítulo 3: se elabora la propuesta de un procedimiento, dicho procedimiento tiene como objetivo asegurar la calidad en la producción de software de la Facultad 3.

Capítulo 4: Validación de la propuesta mediante el método Delphi utilizando expertos en el tema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En la actualidad el software juega un papel muy importante para el desarrollo de las organizaciones, lo que resulta difícil imaginar alguna situación en la que el software no estuviera presente ya que es uno de los componentes básicos de la tecnología que se involucra en las empresas no sólo como soporte a los procesos de negocio sino como parte de las estrategias para la generación de ventajas competitivas. Cualquier organización que se dedica a la investigación, producción y comercialización de software debe considerar la calidad, hoy con más razón, donde existe un mercado en el cual el cliente es cada vez más exigente, no sólo en lo que se refiere al precio, sino sobre todo, en cuanto a los servicios y a la confiabilidad que brindan los productos de software.

Este capítulo abordará diferentes epígrafes que serán de gran ayuda para comprender mejor el objetivo de este trabajo el cual comienza con una introducción a la calidad del software donde se definen aquellos conceptos que conforman la misma. Se expone la importancia que tiene el aseguramiento de la calidad para las empresas que se dedican al desarrollo del software ya que es necesario mejorar la calidad para tener éxito en el mercado y mejorar el rendimiento económico. A medida que se va desarrollando este capítulo también se realizará un estudio profundo acerca de los modelos y estándares existentes para la administración de la calidad los cuales guían de manera eficiente la producción de software y permiten mejorar la calidad de los productos obtenidos. Esto se alcanza haciendo un estudio de los modelos más usados tanto a nivel internacional como nacional, especificando la evolución que estos han tenido en el transcurso del tiempo. También mientras avanza este capítulo se abordará acerca de las diferentes métricas de calidad y como se comporta la evolución y desarrollo de la calidad del software en Cuba y en la UCI.

1.1. Definición de conceptos fundamentales

Calidad

El término calidad es un concepto fácil de visualizar y sin embargo difícil de medir, algunos autores la definen como un término subjetivo donde cada persona tiene su propia definición la cual puede ser a la vez absoluta y relativa. El concepto de calidad ha evolucionado con el tiempo y en dependencia de la profesión base de la persona que la estudie y la utilice como herramienta en la gerencia de las industrias.

Crosby, médico psicólogo, vicepresidente de la compañía telefónica ITT (International Telephone and Telegraph Corp.) durante 14 años para el control de la calidad define calidad en 1979 como: “calidad

no es más que conformidad con los requerimientos indicando que el 100% de la conformancia es igual a cero defectos" [2]. Tiempo después, en 1994, Crosby amplía el concepto de calidad dado que la definición anterior estaba restringida, solo hacía alusión a los requerimientos. "calidad es entregar a los clientes y a nuestros compañeros de trabajo productos y servicios sin defectos y hacerlo a tiempo" [3].
Feigenbaum, presidente de la Academia Internacional de Calidad en 1971 define la calidad como: "la resultante de una combinación de características de ingeniería y de fabricación determinantes del grado de satisfacción que el producto proporcione al consumidor durante su uso". Más tarde, en 1996, Feigenbaum plantea que "calidad es un sistema eficaz para integrar los esfuerzos de mejora de la gestión de los distintos grupos de la organización para proporcionar productos y servicios a niveles que permitan la satisfacción del cliente".
Amador Durán Toro profesor titular de universidad en el área de conocimiento de lenguajes y sistemas informáticos plantea en el 2000 que: "calidad es el conjunto de características de un producto que satisfacen las necesidades de los clientes y en consecuencia hacen satisfactorio el producto" [4].
Edward Deming consultor y difusor del concepto de calidad total y a quien se le acredita parte del éxito de las industrias japonesas, en 1986 define la calidad como: "predecible grado de uniformidad, a bajo costo y útil para el mercado".

Norma de calidad

"Es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad". Los principales organizaciones internacionales, emisoras de normas de calidad son: ISO (Organización Internacional de Estándares) y IEC (Comisión Electrotécnica Internacional).

Proceso: Es un conjunto de prácticas que se ejecutan con un propósito determinado, las cuales transforman entradas en salidas, que son de valor para el cliente. El proceso puede incluir herramientas, métodos, materiales y/o personas [5].

Proceso de producción

Según ISO 9000:2000 los procesos productivos son un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales es preciso realizar para alcanzar la calidad de los productos, eficaz y eficientemente [6].

Harrington otro de los autores respetados define un proceso de producción como: cualquier proceso que entre en contacto físico con el hardware o software que se entregará a un cliente externo hasta aquel punto en el cual el producto se empaca. Esto no incluye los procesos de embarque y distribución.

Procedimiento: Forma especificada para llevar a cabo una actividad o un proceso [7].

Producto: Cualquier elemento que se genera en un proceso [7].

1.2. Calidad del software

En general todos en el mundo de la informática, tienen o pueden crear una idea de lo que significa la definición calidad del software, ya que con el progreso acelerado de las TIC, y la incorporación de estas en todos los entornos productivos y de servicios, el mercado del software ha tenido un gran auge a nivel global y por tanto existen muchos productores de software.

Enfocándose en el cliente, calidad del software podría ser el grado en que un cliente y/o usuario percibe que el producto software satisface sus necesidades y expectativas [8]; pero en la condición industrial del producto, calidad del software es la habilidad de un producto de software de satisfacer su especificación de requerimientos.

El profesor Roger S. Pressman, uno de los grandes investigadores de la ingeniería de software define la calidad del software como: “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”. La calidad del software (CS) es una disciplina más dentro de la ingeniería del software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que sean cumplidos [9].

La calidad está de moda, en todos los aspectos, pero especialmente en el desarrollo de software. El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar productos poco fiables o que realmente no dan respuesta a sus necesidades. La calidad del software (CS) es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia, la cual plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.

Los requisitos del software son la base de las medidas de calidad y la falta de concordancia con los requisitos es una falta de calidad. Si no se sigue ninguna metodología siempre habrá falta de calidad. Todas las metodologías y herramientas tienen un único fin, producir software de alta calidad.
La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software [10]. La calidad del software debe implementarse en todo el ciclo de vida del mismo y se logra a través de la gestión de la calidad, la cual, según ISO 9000:2000, consiste en la realización de actividades coordinadas que permiten dirigir y controlar una organización en lo relativo a la calidad[6].

1.3. Factores que determinan la calidad del SW

La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según los clientes que las necesiten. Los factores que afectan a la calidad del software se pueden categorizar en dos amplios grupos: (1) factores que se pueden medir directamente (por ejemplo, defectos) y (2) factores que se pueden medir solo indirectamente (por ejemplo, facilidad de uso o de mantenimiento) [1] .

<u>Clasificación de factores de calidad</u>:

- Modelo de McCall

El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto (1) operación del producto, (2) revisión del producto y (3) transición del producto.

Operaciones del producto: características operativas.
– Corrección. (¿Hace lo que se le pide?).
• 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. (¿Qué recursos hardware y software necesito?).

• La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.
– Integridad. (¿Puedo controlar su uso?).
• El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
– Facilidad de uso. (¿Es fácil y cómodo de manejar?).
• El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

Revisión del producto: capacidad para soportar cambios.
– Facilidad de mantenimiento. (¿Puedo localizar los fallos?).
• El esfuerzo requerido para localizar y reparar errores.
– Flexibilidad. (¿Puedo añadir nuevas opciones?).
• El esfuerzo requerido para modificar una aplicación en funcionamiento.
– Facilidad de prueba. (¿Puedo probar todas las opciones?).
• 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. (¿Podré usarlo en otra máquina?).
• El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
– Reusabilidad. (¿Podré utilizar alguna parte del software en otra aplicación?).
• Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
– Interoperabilidad. (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?).
• El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

- Modelo FURPS

El modelo FURPS cuenta con 5 características de calidad del software: (1) funcionalidad, (2) facilidad de uso, (3) confiabilidad, (4) performance y (5) facilidad de soporte. Además plantea 2 categorías de requerimientos, las cuales son:
1- requerimientos funcionales (F): especifican funciones que el sistema debe ser capaz de realizar, sin tomar restricciones físicas a consideración, y se definen a través de las entradas y salidas esperadas.

2- requerimientos no funcionales (URPS): usability (facilidad de uso), reliability (confiabilidad), performance y supportability (facilidad de soporte). Describen atributos del sistema o atributos del ambiente del sistema.

Funcionalidad - Los requisitos de funcionalidad deben incluir: (1) conjunto de características, (2) capacidades y (3) seguridad.
Facilidad de uso - Deben incluir subcategorías tales como: (1) factores humanos, (2) estéticos, (3) consistencia en la interfaz de usuario, (4) ayuda en línea, (5) asistentes, (6) documentación del usuario y (7) material de capacitación.
Confiabilidad - Se considera requisitos de confiabilidad: (1) frecuencia y severidad de fallas, (2) recuperación a fallos y (3) tiempo entre fallos.
Performance (rendimiento) - Un requisito de rendimiento impone condiciones a los requisitos funcionales. Por ejemplo a una acción dada, se pueden especificar los siguientes parámetros de rendimiento: (1) velocidad, (2) eficiencia, (3) disponibilidad, (4) tiempo de respuesta, (5) tiempo de recuperación y (6) utilización de recursos.
Soporte - Los requisitos de soporte pueden incluir: (1) requisitos de instalación, (2) requisitos de configuración, (3) requisitos de adaptabilidad y (4) requisitos de compatibilidad.

1.4. Métricas de software

Una métrica es una asignación de un valor a un atributo (tiempo, complejidad, etc.) de una entidad software, ya sea un producto (código) o un proceso (pruebas). Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable. Cuando se habla de software se refiere a la disciplina de recopilar y analizar datos basándose en mediciones reales de software, así como a las escalas de medición. 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.
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. Los principios básicos de la medición son: (1) los objetivos de la medición deberían establecerse antes de empezar la recopilación de datos, (2) todas las técnicas sobre métricas deberían definirse sin ambigüedades, (3) las métricas deberían obtenerse basándose en una teoría válida para

el dominio de aplicación, (4) hay que hacer las métricas a medida para acomodar mejor los productos y procesos específicos, (5) siempre que sea posible, la recopilación de datos y el análisis debería automatizarse, y (6) se deberían aplicar técnicas estadísticas válidas para establecer las relaciones entre los atributos internos del producto y las características externas de la calidad.

Las actividades del proceso de medición son: (1) formulación: obtención de medidas y métricas apropiadas para la presentación del software, (2) colección: mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas, (3) análisis: cálculo de las métricas y aplicación de herramientas matemáticas, (4) interpretación: la evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación, (5) retroalimentación: recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

Se han propuesto cientos de métricas para el software, pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas y otras violan la nociones básicas intuitivas de lo que realmente es el software de alta calidad [1].

Las métricas pueden agruparse según varias clasificaciones (métricas orientadas al software, proyecto, proceso y personas de forma genérica):

Las métricas del proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software. El uso de métricas para el proyecto tiene 2 aspectos fundamentales: (1) minimizar la planificación del desarrollo haciendo los ajustes necesarios que eviten retrasos y reducir problemas/riesgos potenciales; y (2) evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

Las métricas del proceso se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que lleven a mejorar los procesos de software a largo plazo. Estos indicadores de proceso permiten que una organización de ISW pueda tener una visión más profunda de la eficacia de un proceso ya existente y permiten que los gestores evalúen lo que funciona y lo que no. Las métricas del proceso se caracterizan por: (1) el control y ejecución del proyecto, (2) medición de tiempos del análisis, diseño, implementación, implantación y post implantación, (3) medición de las pruebas (errores, cubrimiento, resultado en número de defectos y número de éxito) y (4) medición de la transformación o evolución del producto.

Las métricas de producto son privadas para un individuo y a menudo se combinan para desarrollar métricas del proyecto que sean públicas para un equipo de software. Están enfocadas a predecir y

controlar: (1) el tamaño (líneas de código, bytes de código, operadores y operandos), (2) la estructura (control de flujo, relación entre componentes, cohesión y acoplamiento), (3) la complejidad (combinación de tamaño y estructura), (4) los índices para controlar la documentación, (5) la calidad (independencia, completo, entendible, aumentado) y (6) la estabilidad (los cambios aumentan el número de fallas, los cambios se pueden dar por definición de requerimientos o por cambios del entorno).

A continuación se enuncian algunas de las clasificaciones para las métricas.

Métricas técnicas: se centran en las características del software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho (no en cómo se obtiene).
Métricas de calidad: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.
Métricas de productividad: se centran en el rendimiento del proceso de la ingeniería del software. Es decir qué tan productivo va a ser el software que voy a diseñar.
Métricas orientadas a la persona: proporcionan medidas e información sobre la forma en que las personas desarrollan el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que se harán del personal que va a realizar el sistema.
Métricas orientadas al tamaño: para saber en qué tiempo se va a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño.
Métricas orientadas a la función: son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcularlas, las líneas de código (LDC), las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Ejiogu define un conjunto de atributos que deberían acompañar a las métricas efectivas del software [11]. Las métricas obtenidas y las medidas que conducen a ello deberían ser:
Simples y fáciles de calcular: debería ser relativamente fácil aprender a obtener la métrica y su cálculo no debería demandar un esfuerzo o cantidad de tiempo inusuales.

Empírica e intuitivamente persuasivas: las métricas deberían satisfacer las nociones intuitivas del ingeniero sobre el atributo del producto.
Consistentes y objetivas: las métricas deberían siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información de software.
Consistentes en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería emplear medidas que no conduzcan a extrañas combinaciones de unidades.
Independiente del lenguaje de programación: las métricas deberían basarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa.
Un eficaz mecanismo para la realimentación de calidad: la métrica debería proporcionar, al desarrollador de software, información que le lleve a un producto final de mayor calidad.

De forma general las métricas permiten que los desarrolladores y líderes de proyectos mejoren el proceso de software, planificación y evalúen la calidad del producto. Proporcionan además una visión estratégica de la efectividad de un proceso de software.

1.5. Gestión de la calidad del software

La gestión de la calidad de software es una actividad esencial en cualquier empresa de software para asegurar la calidad de sus productos, y la competitividad frente a la oferta del mercado. Es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. Se basa en la determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales). La gestión o administración de la calidad se aplica normalmente a nivel empresa. También puede haber una gestión de la calidad dentro de la gestión de cada proyecto.
El propósito de la administración de la CS es, en primer lugar, entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas. Dado que la calidad está definida por el cliente, podría parecer que es completamente subjetiva. De cualquier forma, hay muchas cosas acerca de la calidad que pueden hacerse objetivamente. Esto requiere examinar cada una de las características individuales del software y determinar una o más métricas que pueden recolectarse para reflejar dichas características. Por ejemplo, una característica de calidad puede ser que la solución tenga la menor cantidad de errores. Esta característica puede medirse contando los errores y defectos de la solución.

Actividades del proceso de gestión de calidad:

- Establecer políticas, estándares y procedimientos de gestión de la calidad.
- Establecer objetivos de gestión de la calidad de la organización basados en la estrategia empresarial para la satisfacción del cliente.
- Definir las responsabilidades y autoridades para implementar la gestión de la calidad.
- Evaluar e informar sobre la satisfacción del cliente.
- Llevar a cabo revisiones periódicas de planes de calidad de proyectos.
- Monitorizar el estado de las mejoras de calidad de los productos y servicios.

Desde el punto de vista de la calidad, la gestión de la calidad del software está formada por 4 partes, las cuales son: (1) planificación de la CS, (2) control de la CS, (3) aseguramiento de la CS y (4) mejora de la CS.

1.5.1. Planificación de la calidad del software

Según la norma ISO 9000:2000, la planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad. El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias. Los aspectos a considerar en la planificación de la CS son: modelos/estándares de CS (calidad de software) a utilizar, costos de la CS, recursos humanos y materiales necesarios, etc. Los factores que determinan el modelo o estándar de CS a elegir son: (1) la complejidad del proceso de diseño, (2) la madurez del diseño, (3) la complejidad del proceso de producción, (4) las características del producto o servicio, (5) la seguridad del producto o servicio, y (6) económico.

En la planificación de la calidad del software se debe determinar: (1) rol de la planificación, (2) requerimientos de la CS, (3) preparación de un plan de CS, (4) implementación de un plan de CS y (5) preparar un manual de calidad.

Según Humphrey (2002) 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 respecto de 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 usados 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 puede ser presentada en diferentes documentos.

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.

1.5.2. Control de la calidad del software

El control de la calidad del software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centrados en dos objetivos fundamentales:
– Mantener bajo control un proceso.
– Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.
Las distintas actividades para la implantación del control de calidad en el desarrollo de software son: (1) aplicación de metodología y técnicas de desarrollo, (2) reutilización de procesos de revisión formales, (3) prueba del software, (4) ajustes a los estándares de desarrollo, (5) control de cambios, mediciones y recopilación de información; y (6) gestión de informes sobre el control de calidad. En general son las actividades para evaluar la calidad de los productos desarrollados.
El aspecto a considerar en el control de la calidad del software es la “prueba del software”. La prueba es el proceso de ejecutar un programa con intención de encontrar defectos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades. Demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la confiabilidad del software e indican la calidad del software como un todo. Pero, la prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Algunos principios de las pruebas recogen lo siguiente:

- Las pruebas deberían planificarse mucho antes de que comiencen.

- No son posibles las pruebas exhaustivas: El número de permutaciones de camino para incluso programas pequeños es excepcionalmente grande. Por ese motivo es imposible ejecutar todas las combinaciones de caminos durante las pruebas.
- Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente: El ingeniero de software que creó el sistema no es el más adecuado para realizar las pruebas del software, ya que consciente o inconscientemente tiende a probar lo que sabe que funciona.

Existen varios tipos de pruebas que pueden realizarse durante el proceso de desarrollo de software como son:
Unitarias: pretenden probar cada función en un archivo de programa simple (una clase en terminología de objetos).
Integración: pretenden comprobar la integración de los componentes, es decir, la comunicación a través de interfaces, acceso incoherente a estructuras de datos globales. Las pruebas de integración pueden realizarse de forma ascendente o descendente.
Validación: pretende comprobar que se satisfacen los requisitos.
Sistema: se centran en comprobar la recuperación, seguridad, resistencia, rendimiento.

Asociado a los tipos de pruebas existen también técnicas de pruebas que ayudan a definir conjuntos de casos de pruebas aplicando ciertos criterios, como son:
Pruebas de caja blanca: se centra en comprobar la interacción interna de los componentes del sistema.
Pruebas de caja negra: “se centran en los requisitos funcionales del software. O sea, la prueba de de caja negra permite al ingeniero del software obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos”.

La prueba del software es un concepto más amplio que, a menudo, es conocido como verificación y validación. La verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad respecto de los requisitos del cliente. Un plan de prueba traza las clases de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición. 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. 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 [1].
Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad [10].

Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, que requiere los siguientes pasos:

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados, etc.

1.5.3. Aseguramiento de la calidad del software

El aseguramiento de la calidad del software(SQA) es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad [9].

Objetivos del proceso de aseguramiento de la calidad:
-Identificar y planificar las actividades de aseguramiento de la calidad para el proceso o producto.
-Identificar estándares, metodologías, procedimientos y herramientas para llevar a cabo las actividades de aseguramiento de la calidad y su adaptación al proyecto
-Identificar los recursos y las responsabilidades para la realización de las actividades de aseguramiento de la calidad.

-Establecer y garantizar la independencia de los responsables de llevar a cabo las actividades de aseguramiento de la calidad.
-Realizar las actividades identificadas de aseguramiento de la calidad en consonancia con los planes y procedimientos relevantes.
-Aplicar los sistemas de gestión de calidad organizacionales al proyecto.

Todo el que esté involucrado en el proceso de desarrollo del software es responsable de la calidad: desarrolladores, analistas, arquitectos, jefes de proyectos, clientes y aquellas personas que en los proyectos se llaman grupo de aseguramiento de la calidad. El grupo de aseguramiento de la calidad sirve como representación del cliente en casa, la gente que lleva a cabo el aseguramiento de la calidad debe mirar el software desde el punto de vista del cliente. Para esto es necesario formularse preguntas como. ¿Se ha desarrollado el software de acuerdo con estándares preestablecidos? ¿Han desempeñado correctamente sus papeles las disciplinas técnicas como parte del aseguramiento de la calidad? Un grupo de aseguramiento debe responder a este tipo de pregunta para asegurar que mantiene la calidad del software.
El Instituto de Ingeniería del Software (SEI) recomienda un conjunto de actividades como: (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) auditoria 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. 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.
Las revisiones del software son un "filtro" para el proceso de ISW. Es decir, las revisiones se aplican a varios momentos del desarrollo del software y sirven para detectar errores y defectos que pueden ser eliminados. Las mismas sirven para "purificar" las actividades de ISW que suceden como resultado del análisis, diseño y codificación. La revisión técnica formal (RTF), a veces llamada inspección, es el filtro más efectivo desde el punto de vista del aseguramiento de la calidad y es un medio efectivo para mejorar la calidad del software. Dentro de los objetivos de la RTF se puede encontrar: (1) descubrir errores en la función, la lógica o la implementación de cualquier representación del software, (2) verificar que el software bajo revisión alcanza sus requisitos, (3) garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos, (4) conseguir un software desarrollado

en forma uniforme y (5) hacer que los proyectos sean más manejables. El beneficio de estas RTF es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software.

1.5.4. Mejora de la calidad del 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 [6]. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad. También 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. Las razones para realizar una auditoría son: (1) establecer el estado del proyecto, (2) verificar la capacidad de realizar o continuar un trabajo específico, (3) verificar qué elementos aplicables del programa o Plan de aseguramiento de la calidad han sido desarrollados y documentados y (4) verificar la adherencia de esos elementos con el programa o Plan de aseguramiento de la calidad. El propósito y la actividad de la auditoría es recoger, examinar y analizar la información necesaria para tomar las decisiones de aprobación. La misma es realizada de acuerdo con los planes y procedimientos documentados.

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.

1.6. Modelos y estándares existentes para la administración de la calidad

Los modelos y/o estándares permiten que las empresas de software realicen sus tareas y funciones teniendo en cuenta la calidad donde la misma desempeña un rol determinante para la competitividad de la empresa. Cuando una empresa está funcionando y decide implantar un modelo / estándar de calidad, es señal que la empresa tiene el propósito de permanecer y crecer en el mercado, ser competitiva, proteger los intereses de los accionistas, cuidar la fuente de trabajo y mejorar la calidad de vida de su personal.

No existe un único enfoque para solucionar el mal del software. Sin embargo, los resultados de aplicar una mejor tecnología no han resuelto el problema y es en este contexto que diversos organismos internacionales han puesto especial énfasis en la búsqueda de normas para el desarrollo de productos de software, tendiente a garantizar la calidad y la satisfacción total del usuario. Para lograr una mejor gestión de la calidad del software se utilizan modelos y estándares de calidad, los cuales consisten en reunir todas las actividades y funciones de forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático. La correcta determinación e implementación del modelo o estándar de calidad ayuda a que la empresa pueda disminuir sus costos de desarrollo, aumentar las ganancias y administrar mejor sus recursos.

Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos claves y permiten medir los avances en calidad. Los estándares de calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la ISW. Suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad.

Una desventaja de los modelos o estándares es que aún no ha sido demostrada la validez absoluta de ninguno de estos. Las conexiones que se establecen entre características, atributos y métricas se derivan de la experiencia. Esto originó que existan múltiples modelos y estándares de calidad. La terminología para las características de la calidad del software difiere de un modelo de calidad a otro. Cada modelo tiene un número diferente de niveles jerárquicos, así como también un número totalmente diferente de características.

Las ventajas de implantar modelos o estándares de calidad son:

-Tener una oportunidad para corregir los procesos de software que se hayan desajustado con el tiempo.

- Desarrollar y mejorar el nivel y la calidad de vida del personal.
- Generar una cultura organizacional enfocada a cumplir con los requisitos de los clientes.

-Realizar una mejora continua en la calidad de los procesos de software utilizados, servicios y productos de software.

- Lograr que la empresa de software sea más competitiva.
- Reducir los costos en todos los procesos.
- Aumentar la productividad, efectividad y utilidad de la empresa.
- Asegurar la satisfacción de los clientes internos y externos.
- Tener aceptación total de los clientes.

- Tener permanentemente mejores procesos, productos de software y servicios.
- Tener criterios de medición e indicadores congruentes que se utilizan en la empresa para comparar respecto de las mejores prácticas, para conocer fortalezas y debilidades de la empresa; y establecer las estrategias necesarias para realizar mejoras.

Tanto a nivel internacional como nacional se usan diversos modelos que describen características de procesos efectivos y de calidad que indican "qué hacer". Entre estos se pueden mencionar: ISO, CMMI, MoProSoft, SPICE, IEEE, y otros, destacándose ISO como uno de los modelos más usado en nuestro país.

- **Modelo de Capacidad y Madurez (CMM)**

El desarrollo del modelo CMM fue encargado por el Departamento de Defensa de los Estados Unidos, como una ramificación de los problemas experimentados durante la obtención del software, ya que ellos perseguían un medio para la evaluación de los contratistas potenciales. El firme propósito de realizar progresos en los resultados de la calidad del software, se ve envuelto el mejoramiento del proceso de desarrollo, lo cual llevó al Instituto de Ingeniería de Software (SEI) ha desarrollar el Modelo de Capacidad y Madurez, el SEI es el centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la universidad Carnegie-Mellon.

En noviembre de 1986 el SEI, a requerimiento del Gobierno Federal de los Estados Unidos de América, desarrolló una primera definición de un modelo de madurez de procesos en el desarrollo de software, pero no se publicó hasta septiembre de 1987. Este trabajo evolucionó al modelo CMM o SW-CMM (CMM para el Software), se publicó en febrero de 1993. Este modelo establece un conjunto de prácticas o procesos clave agrupados en áreas clave de proceso, conocidas a través de la documentación del CMM por su sigla inglesa: KPA, para cada una de ellas se define un conjunto de buenas prácticas que deberán ser:

- Definidas en un procedimiento documentado.
- Provistas (la organización) de los medios y formación necesarios.
- Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas).
- Medidas.
- Verificadas.

Modelo: CMMI v1.2.

En agosto de 2006 aparece CMMI v1.2, incluye mejoras significativas dando respuesta a asuntos que fueron surgiendo en la práctica con la versión anterior (CMMI v1.1). Los cambios se producen para mejorar la calidad de los productos CMMI y la consistencia con que son aplicados.

El modelo CMMI constituye un marco de referencia de la capacidad de las organizaciones de desarrollo de software en el desempeño de sus diferentes procesos, proporcionando una base para la evaluación de la madurez de las mismas y una guía para implementar una estrategia para la mejora continua de los mismos. El propósito del modelo de Madurez de Capacidad Integrado (CMMI), es guiar a las organizaciones en la tarea de mejorar sus procesos y su habilidad para organizar, desarrollar, adquirir y mantener productos y servicios informáticos.

Este modelo de calidad cuenta con 22 áreas de procesos las cuales se agrupan en 4 categorías:

Administración de procesos.

Innovación y despliegue de organización.

Definición de proceso de organización +IPPD.

Foco de proceso de organización.

Funcionamiento de proceso de organización.

Entrenamiento de la organización.

Administración de proyecto.

Gestión del proyecto integrada + IPPD.

Monitorización y control del proyecto.

Planificación del proyecto.

Gestión cuantitativa del proyecto.

Gestión de Riesgo.

Gestión de acuerdos con los proveedores.

Ingeniería.

Integración del producto.

Desarrollo de los requisitos.

Administración de requisitos.

Solución técnica.

Validación.

Verificación.

Soporte.

Análisis causal y resolución.

Administración de configuración.

Análisis y resolución de la decisión.

Medida y análisis.

- **SPICE**

SPICE también es un modelo de calidad de suma importancia. El objetivo de este modelo es proporcionar un marco de referencia para la valoración de los procesos de software. Fomentar la calidad de los productos de software y generar un proceso de valoración repetible, comparable y verificable. SPICE fomenta productos de calidad, promueve la optimización de procesos y facilita la evaluación del producto a través de los procesos de desarrollo. Tiene diversos alcances, se aplica tanto a nivel directivo como a nivel de usuarios para asegurar que el proceso se encuentra alineado con las necesidades del negocio, se apoya en que los proveedores de software tengan que someterse a una sola evaluación para aspirar a nuevos negocios y busca que las organizaciones de software dispongan de una herramienta universalmente reconocida para dar soporte a su programa de mejoramiento continuo.

SPICE tiene tres características principales: (1) el marco de valor que contempla una dimensión funcional de los procesos, (2) la evidencia para la evaluación y; (3) la recurrencia dada por la selección de instancias de proyectos o productos. Este estándar proporciona un enfoque estructurado para la evaluación de procesos de software, es decir, organizaciones con el objetivo de: (1) comprender el estado de sus propios procesos para la mejora de los mismos, (2) determinar la idoneidad de sus propios procesos para un requerimiento particular o clases de requerimientos y (3) determinar la idoneidad de procesos de otras organizaciones para un contrato particular o clase de contratos.

- **IEEE Standards Software Engineering**

A la hora de hablar de los estándares para la calidad de software es prácticamente imposible dejar de mencionar el estándar IEEE para el Plan de Aseguramiento de la Calidad del Software.

El Instituto de Ingeniería en Eléctrica y Electrónica (IEEE - Institute of Electrical and Electronics Engineers) es una organización de ámbito mundial, fundada en 1884, que agrupa a profesionales de los campos de la ingeniería y la informática.

Los objetivos de IEEE son:
(1) Promover el avance de las teorías y las prácticas de electro tecnología.
(2) Fomentar el progreso y el desarrollo profesional de su membrecía.
(3) Mejorar la calidad de vida a través de la aplicación de electro tecnología.
(4) Promover el entendimiento de electro tecnología ante el público.
Los estándares de IEEE están orientados: (1) al cliente y a la terminología estándar. (2) al proceso, (3) al producto y (4) al recurso y a las técnicas.

El estándar IEEE 730 establece el puente entre la gestión de la calidad y la ingeniería del software, el cual recomienda unos requerimientos para llevar a cabo un Plan de Garantía de Calidad asociado a un proyecto de software. Cabe señalar, que mientras que ISO 9000-3 está pensado para ser aplicado en toda una organización, el IEEE 730 es aplicado a un único proyecto dentro de esa organización. A su vez el IEEE 730 tiene relación con otros estándares como el IEEE 828 o el 1219, que son de utilidad para aplicar las necesidades expuestas en el 730.
Siguiendo el estándar IEEE 730 sobre planes de aseguramiento de la calidad del SW, el plan debe incluir:
–Objetivos de calidad del proyecto y enfoque adoptado para alcanzarlos.
–Documentación referenciada en el plan (manuales, procedimientos, etc.)
–Gestión del aseguramiento de la calidad (organización, actividades y responsabilidades).
–Documentación mínima exigida a los desarrolladores tanto del desarrollo del SW (especificaciones, diseño, manuales de usuario, etc.) como de control (planes de Validación y Verificación).
–Estándares que se deben aplicar obligatoriamente.
–Actividades de revisión y auditoria.
–Gestión de la configuración del SW (mediante un plan de gestión específico o describiendo sus actividades).
–Informes de problemas, especificando la forma de tratar y corregir los problemas y los responsables que han de hacerlo.
–Herramientas para apoyar el aseguramiento de la calidad (revisiones, inspecciones, analizadores de código, generadores de entornos de prueba, etc.), especificando sus objetivos y la manera de utilizarlas.
–Control del código (almacenamiento y versiones), control de acceso a equipos y prevenciones de seguridad y control de las características del SW a los suministradores.

–Recogida, almacenamiento y mantenimiento de datos sobre el aseguramiento de la calidad.
–Métricas del SW para controlar el proyecto.
–Verificación y validación del SW (incluyendo pruebas y revisiones) en todas las fases del ciclo de vida.

- **ISO**

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios de aplicaciones. ISO 9000 es el nombre genérico con el que coloquialmente se designa a una familia de normas de aseguramiento de la calidad. La serie ISO 9000 está formada por 5 documentos, 3 de ellos son modelos de aseguramiento de la calidad, específicamente el 9001, 9002 y 9003. Los otros dos son simples lineamientos que sirven de apoyo.
ISO 9000, principios y conceptos. Lineamientos para su selección y utilización.
ISO 9001, modelo de aseguramiento de la calidad, aplicable al diseño, desarrollo, fabricación, instalación y servicio.
ISO 9002, modelo de aseguramiento de la calidad, aplicable a la fabricación y a la instalación.
ISO 9003, modelo de aseguramiento de la calidad, aplicable a la inspección y ensayos finales.
ISO 9004, principios y conceptos. Lineamientos para la gestión de calidad y elementos del sistema de calidad.

1.7. Calidad del software en Cuba.

Las tecnologías de información (TI) se han convertido en los últimos años en un elemento clave para el aumento del comportamiento de la economía en varios países. El término Informatización de la sociedad, es común ya en casi todos los programas de desarrollado de todos los países. Para Cuba, como país en vía de desarrollo, el desarrollo de las TI y la industria del software en particular, es un medio importante para incrementar su desempeño económico, atendiendo a que la industria del software requiere menos inversión que las restantes, tomando en consideración que la inversión principal ya se ha realizado a partir de la política educacional que ha seguido la revolución desde 1959.

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. La promoción de la ICSW en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la

educación y el deporte. El continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación. A pesar de ello, los resultados alcanzados no cubren las expectativas, ya que la productividad es baja, la cantidad real de recursos a consumir en tiempo principalmente es casi impredecible y el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida, los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas no se detectan en este medio indisciplinado y caótico de desarrollo [12]

En Cuba, el Ministerio de la Informática y las Comunicaciones (MIC), es el organismo encargado de regular, dirigir, supervisar y controlar la política del Estado y del Gobierno en cuanto a las actividades de tecnologías informáticas, telecomunicaciones, redes de infocomunicaciones, entre otras. El cual, entre sus principales funciones tiene el establecimiento, regulación y control de la política para el desarrollo, producción y comercialización de la Industria Nacional de Software.

La producción de software se convierte en una actividad cada vez más demandada, lo que provoca que las empresas dedicadas a la producción de bienes y servicios informáticos tiendan a buscar las vías idóneas para el perfeccionamiento de la gestión empresarial, para lograrlo también utilizan los modelos de calidad, en este contexto la gestión comercial es determinante para el mejoramiento de los resultados productivos de estas empresas.

Una de las empresas punteras en este sector es DESOFT, su interés por lograr la calidad crece de forma continua a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades, se rigen por las normas cubanas ISO 9000:2005, 90003:2004 y 12207:1995, pudiendo así cumplir con las expectativas de sus clientes.

Otra empresa de gran prestigio lo es sin duda el Centro de calidad de software (CALISOFT), surgida en el año 2004 con el objetivo de diseñar sistemas informáticos de alto nivel, sus principales objetivos son: entregar productos de gran calidad con estricta puntualidad, para lograrlos cuentan con un grupo de profesionales altamente capacitados y con amplia experiencia garantizando así la exitosa entrega del proyecto. Contribuye en gran medida al desarrollo de la industria de software en el país incrementando su competitividad en el mercado internacional.

La Casa de software Sigta de ETECSA es una empresa cubana que llevando a cabo buenas prácticas cumplió con las exigencias que propone CMM, alcanzando el nivel 2 y logrando de esta manera resultados evidentes con la aplicación de este modelo de procesos. [13]
Entre los modelos adoptados por Cuba usados en la industria del software vigentes actualmente, se pueden mencionar el modelo NC ISO 9001:2001, NC-ISO-IEC 12119: 2005 y NC- ISO/IEC 9126-1:2005. [14]

1.7.1. Calidad del software en La Universidad de las Ciencias informáticas

La UCI pretende convertir la industria del software en un renglón fundamental de la economía e insertarnos en el mercado internacional, se coordinan y desarrollan los programas de informatización de sectores fundamentales del país, y proyectos estratégicos para la exportación. El área de comercialización fue creada en el 2005 que integra las soluciones de todos y se denomina Albet [15]. La UCI es un programa de la revolución que garantiza la formación de personal en las ciencias informáticas que acaba con la crisis de personal. Esta es una industria de jóvenes, quienes tienen todos los recursos y le dedican todas sus energías al desarrollo de los proyectos, la universidad convoca a expertos de otras instituciones para integrarse a ella y lograr sus objetivos con éxito.

Actualmente en la UCI no se usa un modelo de procesos en específico, ya que ninguno de los modelos de procesos se adapta al 100% debido a la compleja estructura productiva que posee la institución, por lo que la tendencia es crear un modelo a partir de varios, usando plantillas predeterminadas de modelos como CMMI, IEEE, MoProSoft, ISO [16].

La UCI se ha concebido como agente de cambio en la ICSW dada a la concentración de miles de especialistas y recursos para la producción del software. La producción se orienta a las necesidades del mercado. El proyecto más importante es el de informatización del sistema nacional de salud, este es chequeado semanalmente por viceministros del MINSAP, MIC y vicerrectora de producción. En estos momentos 48 bancos de sangres ya se encuentran informatizados y existe un Registro Nacional de Donantes. La infraestructura de producción de la universidad está organizada por: Calidad: Lab de certificación, grupo de auditoría y revisiones, aseguramiento de la calidad y métricas. CALISOFT centro de calidad del software. Tecnología: tecnologías y arquitectura, arquitectura de información, soporte al desarrollo. Comunicación visual: grupo creación, grupo producción, grupo realización.

La universidad básicamente tiene 3 líneas de producción: produce para la informatización del país, para los proyectos que indica la dirección del país y los grandes proyectos de exportación que el país tiene. Los proyectos son organizados, gestionados y ejecutados desde la facultad y en cada uno existe una estructura capaz de cumplir todas las tareas que el proyecto genera.

En la actualidad la UCI trabaja con más de 20 organismos centrales del estado, en cuanto a la exportación se trabaja con los proyectos del ALBA aunque se han recibido propuestas de otras empresas extranjeras. El modelo es la integración de UCI, empresa y expertos, los cuales conforman el equipo "Cuba", como se le conoce en la UCI. Hay que adecuar los modelos de producción y los de calidad internacional a las características de la universidad.

Lineamientos mínimos de calidad definidos por la UCI

A continuación se muestran los lineamientos mínimos de calidad definidos por la UCI [17].

Proyecto

1. Definir en cada proyecto una estrategia de trabajo educativo en concordancia con los valores que se propone formar la universidad en sus graduados [Propuesta: Planilla "Plan Estratégico para el trabajo político Ideológico"].
2. Las primeras actividades planificadas deben estar encaminadas a la motivación con el proyecto.
3. Se debe llevar un registro de las actividades conjuntas y visitas de los diferentes factores al proyecto.
4. El líder de proyecto debe contar con conocimientos básicos del proceso de desarrollo de software.
5. Los miembros del equipo deben recibir superación antes o durante del proceso que les permita enfrentarse al problema en cuestión con una visión amplia y con un número mayor de herramientas.
6. Debe realizarse una asignación de responsabilidad material y entrega de medios estrictos, tanto para medios gastables o no gastables.
7. Debe estar claramente definida la Visión del Proyecto, en la que se integre de manera armónica la visión de todos los factores que intervienen en el negocio. [Propuesta: Documentos para la Gestión del Proyecto, iniciamos por el documento de Visión del Proyecto"].
8. Deben estar identificados los riesgos que amenazan al proyecto, donde se debe hacer hincapié en el tema de la seguridad de la información y extenderlo a los recursos materiales del proyecto [Propuesta: Planilla "Lista de riesgos"].

Trabajo en equipo

1. Debe quedar definida y documentada una estructura organizacional que permita identificar y agrupar las tareas requeridas, seleccionar y establecer las estructuras organizativas, definir roles y relaciones de autoridad en el proyecto.
2. Debe realizarse un estudio de los roles que serán desempeñados en el proyecto y de las personas a las cuales serán asignados dichos roles, teniendo en cuenta responsabilidades y objetivos que debe cumplir este rol así como la formación y los conocimientos que posee la persona.
3. En el proyecto no deben olvidarse definir roles comúnmente obviados como es el caso de: Planificador, Integrador, Ingeniero en Componentes, Diseñador de pruebas y Probador.
4. Las reuniones de equipo deben realizarse de forma sistemática y quedar reflejadas en el cronograma de trabajo como parte del proceso normal de desarrollo.
5. La realización de las tareas en todas sus fases debe ser registrada de forma medible en función del tiempo y del volumen de trabajo (el estimado contra el real) y definir métricas sobre el tiempo que dedica un desarrollador a cada tarea específica [Propuesta: Planilla “Cuaderno de trabajos”].
6. En cada fase del proceso de desarrollo deben ser registrados los defectos detectados, obtener datos sobre estos, analizarlos y determinar métricas que permitan detectar la mejor forma de reducirlos. [Propuesta: Planilla “Registro de No Conformidades”].

Planificación

1. Deben ser planificadas las fases y las iteraciones del ciclo de vida del proyecto y quedar registrada dicha planificación en el cronograma, definiéndose los entregables que evidencien que se ha vencido una iteración o una fase.
2. Deben establecerse puntos de chequeo que permitan verificar el estado del proceso.
3. Las tareas deben ser atómicas y bien delimitado su tamaño para poder precisar su avance.
4. Definir y documentar en cada etapa los artefactos que deben producirse y una vez obtenidos deben ser evaluados.

Gestión de configuración

1. Debe definirse cuáles son los tipos de elementos de configuración que se obtendrán en el sistema.

2. Realizar control de versiones sobre los elementos de configuración y definir una herramienta para este fin.
3. Deben documentarse todos los cambios realizados y existir una persona que apruebe dicho cambio.
4. Deben documentarse todos los elementos que fueron afectados con el cambio.
5. Una vez implementado un cambio debe probarse nuevamente.
6. Deben estar definidas las líneas base del proyecto.
7. Deben controlarse los releases (Liberaciones parciales o totales del producto).

Trabajo personal

1. Deben registrarse el tiempo personal de trabajo de los miembros de equipo.
2. Deben registrarse de forma individual los defectos encontrados.
3. Debe llevarse un control personal de los compromisos.

Modelos de calidad

1. Debe existir conocimiento por parte de los miembros del equipo de los modelos de calidad existentes y regirse por alguno de ellos que estará bien definido y documentado.
2. Si no se utiliza ningún modelo de calidad debe existir, al menos, definido en el proyecto un plan de calidad.

Conclusiones del capítulo 1

- El software juega un papel muy importante para el desarrollo de las organizaciones, ya que sirve de soporte a los procesos de negocio, productivos y administrativos; y como parte integral de las estrategias corporativas para la generación de ventajas competitivas.
- Para el logro de esta calidad será necesario efectuar una gestión de la calidad, la cual consiste en un conjunto de actividades que permiten dirigir y controlar la organización en lo relativo a la calidad del software.
- Para lograr una mejor gestión de la calidad del software se utilizan modelos y estándares de calidad ya que está demostrado lo necesario que es la aplicación de modelos para asegurar la calidad de un producto porque se aprecia que en el mundo entero la demanda de los productos está dada por la certificación en algún modelo que mida la calidad y la capacidad de los productos y los procesos de las empresas desarrolladoras de software.
- El aseguramiento de la calidad del software es una actividad de protección que debe estar presente en todas las etapas de desarrollo de cualquier producto y debe de estar presente en toda organización que esté encaminada a obtener un software con la calidad requerida.

CAPÍTULO 2: CARACTERIZACIÓN Y DIAGNÓSTICO DE LA PRODUCCIÓN DE SOFTWARE DE LA FACULTAD 3.

Introducción

El presente capítulo contiene un estudio a cerca de la producción de software de la Facultad 3 para determinar su situación actual. Para este diagnóstico se aplicaron técnicas para el diagnóstico tales como las encuestas y entrevistas.

2.1. Grupo de control y aseguramiento de la calidad en la facultad 3.

En la facultad existe un grupo de calidad el cual tiene la misión de orientar y guiar a todos los proyectos en el proceso de control y aseguramiento de la calidad en cada una de sus etapas de desarrollo [18].

Misión

Ayudar a los proyectos productivos a mejorar su gestión y en consecuencia, sus resultados y la satisfacción de sus clientes y otras partes interesadas a través de la optimización de sus procesos y recursos para que sean más eficaces y eficientes.

Visión

El grupo de calidad nació con el propósito de dignificar la producción de software en la Facultad 3, con base en los valores de honestidad, esfuerzo compartido, trabajo en equipo, unidad, y credibilidad.

Servicios

Consultorías:

–Guiar y orientar el proceso de aseguramiento de la calidad de cada uno de los proyectos de la facultad.

– Proporcionar los lineamientos a seguir para cada una de las etapas de desarrollo del SW.

– Brindar capacitación a los equipos de SQA de cada proyecto.

Auditorías:

– Revisar toda la documentación del proyecto, que se encuentre completa y actualizada; específicamente el expediente del proyecto.

Revisiones:

– Detectar la mayor cantidad de errores posibles durante cada una de las etapas de desarrollo, evitando que los mismos se conviertan en defectos que afecten el producto final, y por ende, la imagen del equipo de desarrollo ante los clientes.

Las revisiones se dividen en dos tipos fundamentales:

a) Revisiones técnicas formales:

– Cada RTF se centra en una parte específica y pequeña del SW total, puede tener como objetivos: descubrir errores en la función, la lógica o la implementación de cualquier representación de SW; verificar que el SW bajo revisión alcanza sus requisitos; garantizar que el SW haya sido representado de acuerdo a ciertos estándares predefinidos; conseguir un SW desarrollado de forma uniforme y hacer que los proyectos sean más manejables.

b) Pruebas de SW

– Las pruebas de software se realizan al terminar cada una de las etapas del proceso de desarrollo, dentro de estas están las pruebas de sistema, de integración y a los módulos.

2.2. Aplicación de técnicas para el diagnóstico

Las principales técnicas que se emplearon para fundamentar la investigación de este trabajo fueron las siguientes:

Entrevista

Dentro de una organización, la entrevista es la técnica más significativa y productiva de que dispone el analista para recabar datos. En otras palabras, la entrevista es un intercambio de información que se efectúa cara a cara. Es un canal de comunicación entre el analista y la organización; sirve para obtener información acerca de las necesidades y la manera de satisfacerlas, así como consejo y comprensión por parte del usuario para toda idea o métodos nuevos. Por otra parte, la entrevista ofrece al analista

una excelente oportunidad para establecer una corriente de simpatía con el personal usuario, lo cual es fundamental en el transcurso del estudio. Es una forma de conversación, no de interrogación [19].

Para detectar problemas existentes en los proyectos productivos de la facultad en cuanto al aseguramiento de la calidad fue necesario realizar la entrevista, aplicada a los integrantes del grupo de calidad de los diferentes proyectos productivos (ver anexo 1).

A partir de los resultados de estas entrevistas realizadas a los responsables de calidad en cada proyecto se demostró que existen problemas en cuanto la forma de asegurar la calidad. De las entrevistas que se realizaron el 77% afirmó asegurar la calidad mediante el plan de aseguramiento; mientras que el 23% restante la asegura de diferentes formas, como la selección de los roles de los integrantes del proyecto, y que las personas sean buenas en lo que hacen. El 66% de los entrevistados no sabe claramente cuáles son los pasos para asegurar la calidad, pero plantean que es un proceso que depende de cómo se van haciendo las cosas que hay que hacer, el resto dice que se debe de seguir el plan de aseguramiento con todo lo establecido en él, se deben de hacer RTF y las pruebas correspondientes en cada etapa por las que transite el proyecto. El 88% de los entrevistados responden que para asegurar que se ha hecho correctamente se debe probar el sistema y si es una investigación pues asegurar que la información sea la más óptima, indagando y observando la realidad; mientras que el 12 % dice que al final del desarrollo el cliente está totalmente satisfecho con el producto, aunque esto no implica que realmente se halla hecho de la forma más correcta. El 100% de los entrevistados plantean que los principales problemas que existen en cuanto al aseguramiento de la calidad es que el conocimiento que tienen sobre el tema lo aprendieron sobre la marcha, no recibieron preparación sobre el tema para realizar las actividades de SQA, que existen problemas, desde que se seleccionan los roles, cuando se hacen las cosas apresuradamente, y cuando no se asegura que lo que se hace es importante, porque la gente trabaja solo para cumplir y no porque se siente parte de el proyecto.

Encuesta

Se realiza cuando la información puede ser obtenida a partir de la respuesta que una persona o varias puedan dar a un cuestionario pre-elaborado, y las mismas están dispuestas a colaborar con la investigación.

La encuesta es semejante a la entrevista pero escrita, donde a través de un conjunto de preguntas se pretende obtener una información sobre el mundo interior del encuestado o su percepción del fenómeno que se investiga, por lo que no puede ser obtenida por observación [19].

2.2.1. Muestreo probabilístico

El muestreo es una herramienta de la investigación científica. Su función básica es determinar que parte de una realidad en estudio (población o universo) debe examinarse con la finalidad de hacer inferencias sobre dicha población. El error que se comete debido a hecho de que se obtienen conclusiones sobre cierta realidad a partir de la observación de solo una parte de ella, se denomina error de muestreo. Obtener una muestra adecuada significa lograr una versión simplificada de la población, que reproduzca de algún modo sus rasgos básicos.

Los métodos de muestreo probabilísticos son aquellos que se basan en el principio de equiprobabilidad. Es decir, aquellos en los que todos los individuos tienen la misma probabilidad de ser elegidos para formar parte de una muestra y, consiguientemente, todas las posibles muestras de tamaño n tienen la misma probabilidad de ser elegidas. Sólo estos métodos de muestreo probabilísticos nos aseguran la representatividad de la muestra extraída y son, por tanto, los más recomendables. Dentro de los métodos de muestreo probabilísticos encontramos los siguientes tipos:

Muestreo aleatorio simple.

Muestreo sistemático.

Muestreo aleatorio estratificado.

Muestreo por conglomerados (clusters).

Muestreo Aleatorio estratificado

En este trabajo se utilizó este método para determinar la cantidad de población que será encuestada, y el tamaño de la muestra, es decir, la cantidad de encuestas que se deben realizar.

Una muestra es estratificada cuando los elementos de la muestra son proporcionales a su presencia en la población. La presencia de un elemento en un estrato excluye su presencia en otro. Para este tipo de muestreo, se divide a la población en varios grupos o estratos con el fin de dar representatividad a los distintos factores que integran el universo de estudio. Para la selección de los elementos o unidades representantes, se utiliza el método de muestreo aleatorio.

A las subpoblaciones se les llamará estratos, se tratará de conformar estos de modo que los elementos dentro de ellas sean homogéneos. El tamaño de la muestra se distribuirá entre los estratos, en función de distintos criterios, pero lo que caracteriza al MAE (Muestreo Aleatorio Estratificado) es

que la selección de la muestra de cada estrato se hará bajo el procedimiento de muestreo irrestricto aleatorio y se realizará independientemente de los diferentes estratos.

Es recomendable estratificar en función del tamaño de las unidades y distribuir la muestra proporcionalmente al número de unidades de los estratos.

Notación

Para una descripción general del muestreo aleatorio estratificado y los métodos de inferencia asociados con este procedimiento, suponemos que la población está dividida en h subpoblaciones o estratos de tamaños conocidos N_1, N_2,..., N_h tal que las unidades en cada estrato sean homogéneas respecto a la característica en cuestión.

$N = N_1 + N_2 + N_3 + \ldots + N_h$

Total de la población:

$$x_h = \sum_{1}^{N_h} x_{hi}$$

Si el tamaño de la muestra de los estratos se distribuye proporcionalmente al número de unidades en el estrato, se cumple que:

$$n_h = n\frac{N_h}{N}$$

Para todo h. En este caso se dice que la distribución de la muestra se ha hecho con asignación proporcional.

Para la determinación del tamaño de muestra de una población con varianza (S^2) desconocida se puede determinar mediante la expresión:

$$n = \frac{\left[\frac{Z_{1-\frac{\alpha}{2}}}{d}\right]^2 P(1-P)}{1 + \frac{1}{N}\left[\frac{Z_{1-\frac{\alpha}{2}}}{d}\right]^2 P(1-P) - \frac{1}{N}}$$

Donde:

1-α es el nivel de confianza;

d es el error absoluto;

$Z_{1-\alpha/2}$ es el percentil de la distribución normal;

P proporción de la población;

N tamaño de la población.

Realizando el cálculo del tamaño de muestra para un nivel de confianza 1-α=90%, con un error absoluto de d=0.10, se obtiene el percentil de la distribución normal Z_1=1.64 y se asume como proporción de la población *P*=0.5, *N*=256 que es la cantidad total de todos los integrantes de los proyectos de la facultad; se obtiene que:

$$n = \frac{\left[\frac{1,64}{0,10}\right]^2 0,5(1-0,5)}{1 + \frac{1}{256}\left[\frac{1,64}{0,10}\right]^2 0,5(1-0,5) - \frac{1}{256}} = \frac{67,24}{1,06} \approx 63$$

Se deben aplicar un total de 63 encuestas.

Conociendo que *n* =63, N_h es la cantidad de integrantes en cada proyecto, *N*=256 que es total de integrantes de los diferentes proyectos.

Calculando el tamaño de muestra por estrato (se considera estrato a cada proyecto) aplicando

$$n_h = n\frac{N_h}{N}$$

Resultan los tamaños de muestra por estrato que se muestran en la siguiente tabla.

Estratos	Cantidad	Tamaño de la muestra
CCV	44	11
Informatización de la IP	10	2
Menpet	15	4
Arquitectura	6	1
ONE	30	7
CE	9	2
ERP	55	14
RN	65	16
FGR	22	5

Tabla 2.1 Tamaños de muestra por estratos de la población.

2.3. Diagnóstico aplicado a los proyectos productivos de la Facultad 3.

Resultados de la encuesta aplicada a los integrantes de los proyectos productivos de la facultad 3.

Para detectar problemas existentes en los proyectos productivos de la facultad en cuanto al aseguramiento de la calidad se realizó una encuesta, aplicada a los integrantes de los diferentes proyectos productivos, la cual fue aprobada por Ramsés Delgado Martínez (ver anexo 2).

La muestra hallada por regiones representa el 25; 20; 26,66; 16,66; 23,33; 22,22; 25,45; 24,61; 22,72 %, respectivamente, de los integrantes (estudiantes y profesores) de los diferentes proyectos de la facultad. A estos grupos se les aplicó la encuesta relacionada con el aseguramiento de la calidad en la que se recoge cómo se manifiesta el aseguramiento de la calidad en el software que se desarrolla, con vista a detectar los principales problemas que afectan la calidad del producto y de los procesos en los proyectos de la facultad.

Los resultados obtenidos revelan en sentido general que no todos los integrantes del proyecto tienen conocimientos acerca de los principales estándares y modelos para la gestión de la calidad y no todos tienen bien definidos el concepto de aseguramiento de la calidad del software, además de que una gran parte no cree que es importante tener un sistema de aseguramiento de la calidad.

El 7,5% de los encuestados plantea que la calidad del producto que se obtiene es excelente, el 68,25% buena y el 23,81% dice que es regular, por lo que se puede decir que el producto que se desarrolla en la Facultad 3 tiene buena calidad.

En cuanto a la etapa en la que se desarrolla el aseguramiento de la calidad, el 96,83% de los encuestados asegura que es en todo el proceso del software y solo un 3,17% dice que es al final.

Según los resultados obtenidos en la mayoría de los proyectos, las tareas se realizan a través de roles.

En cuanto a la utilización de algún estándar en el proyecto el 19,05% plantea que lo hacen a través de modelos o estándares conocidos, el 19,05% dice que no lo utilizan y un 61,90% desconoce sobre la utilización de algún estándar en su proyecto.

A continuación aparecen los resultados de la encuesta a través de las siguientes tablas:

Concepto de SQA	Total	Porciento (%)
Si	45	71,43
No	18	28,57

Tabla 2.2. Resultados sobre el conocimiento del concepto de SQA en los proyectos.

Etapa en que se desarrollan	Total	Porciento (%)
No se realiza	0	0
Al inicio del proceso del software	0	0
Todo el proceso de software	61	96,83
Al final del proceso del software	2	3,17

Tabla 2.3. Resultados en los proyectos sobre las etapas en la que se desarrolla el aseguramiento de la calidad

Calidad del producto	Total	Porciento (%)
Excelente	5	7,94
Buena	43	68,25
Regular	15	23,81
Mala	0	0

Tabla 2.4. Resultados sobre la calidad del producto en los proyectos.

Realizan las tareas a través de los roles	Total	Porciento (%)
Si	61	96,83
No	1	1,59
No sé	1	1,59

Tabla 2.5. Resultados sobre la realización de las tareas a través de roles

Conoce algún modelo o estándar	Total	Porciento (%)
Si	42	66,67
No	9	14,29
No sé	11	17,46

Tabla 2.6. Resultados sobre el conocimiento de los modelos o estándares existentes para la gestión de la calidad

Utilizan algún estándar en el proyecto	Total	Porciento (%)
Si	12	19,05
No	12	19,05
No sé	39	61,90

Tabla 2.7. Resultados sobre la utilización de algún estándar en el proyecto.

Es importante tener un SQA	Total	Porciento (%)
Si	50	79,37
No	13	20,63

Tabla2.8. Resultados sobre la importancia de tener un SQA en lo proyectos.

Conclusiones del capítulo 2

- Existe una mala asignación de roles debido a que los integrantes del proyecto tienen poca experiencia o ninguna en cuanto a su rol.
- Los proyectos productivos de la facultad 3 no tienen definido un procedimiento que los guíe para asegurar la calidad.
- Los integrantes del proyecto no tienen un conocimiento adecuado sobre el tema, lo que dificulta realizar correctamente las actividades de aseguramiento de la calidad.

CAPÍTULO 3: PROPUESTA DE UN PROCEDIMIENTO PARA EL ASEGURAMIENTO DE LA CALIDAD EN LOS PROYECTOS PRODUCTIVOS DE LA FACULTAD 3.

Introducción

En este capítulo se explica la propuesta de solución, que puede ser aplicada a los proyectos productivos que se desarrollan en la Facultad 3. Para la realización de la propuesta de este procedimiento se ha definido un conjunto de etapas donde se definen las actividades a desarrollar, quienes son los responsables y los artefactos que se generan. Estas etapas son: capacitación del personal; planificación de la calidad; control de la calidad y por último la evaluación de la calidad del producto final que ha sido desarrollado. El procedimiento está basado en la metodología RUP, que es la más utilizada en los proyectos productivos de la facultad.

Las actividades de aseguramiento de la calidad están dirigidas a garantizar que se realicen los diferentes procesos en cada una de la fases de desarrollo del software y se obtenga un producto al culminar cada fase; de esta forma en la última fase, el software obtenido tendrá una alta calidad y el cliente se sentirá satisfecho con el mismo. Para ello es necesario realizar las actividades de SQA en cada una de las fases de desarrollo (ver anexos 3, 4, 5, 6). Para el buen funcionamiento de este procedimiento, las actividades de SQA deben realizarse en el orden que se desarrollan las etapas propuestas, recordando siempre que las actividades de la etapa de capacitación del personal se realizan fuera de las fases de desarrollo que propone RUP.

3.1. Propuesta de un procedimiento de SQA

El procedimiento está conformado por la siguiente estructura:

1 Propósito

El propósito que se pretende con la propuesta de este procedimiento es definir una serie de pasos a seguir que guíen la producción en la facultad para lograr alcanzar en el software desarrollado el nivel de calidad que se requiere.

2 Objetivos

Asegurar la calidad del software en los proyectos productivos de la Facultad 3.

3 Definiciones y términos

Artefacto: productos tangibles del proyecto, que son producidos, modificados y usados por las actividades.

Auditoría: dirigida por una persona autorizada con el propósito de proporcionar una evaluación independiente de los productos y procesos de software con el fin de evaluar la conformidad con los requisitos.

Fases: representa un ciclo de desarrollo en la vida de un producto de software.

Iteraciones: secuencia de actividades.

Metodología: es un proceso que define quién debe hacer qué, cuándo y cómo debe hacerlo.

Proceso: es una definición del conjunto completo de actividades para transformar los requisitos del usuario en un producto.

Producto de software: artefacto que se crean durante la vida del proyecto.

Proyecto: elemento organizativo a través del cual se gestiona el desarrollo del software, el resultado de un proyecto es una versión del producto

Riesgo: peligro, probabilidad de un daño.

Rol: responsabilidades de cada uno de los miembros del proyecto.

4 Responsabilidad del equipo de SQA

En cada proyecto de la facultad debe crearse un grupo SQA, el cual debe participar en el establecimiento de los planes, procesos, estándares y procedimientos definidos en el Plan del Proyecto. Este grupo debe funcionar como el cliente en casa y mirar el software desde el punto de vista del mismo, así como asegurar la calidad del proceso y el producto. Para llevar a cabo un proceso con calidad se deben poner en práctica un grupo de actividades a lo largo de todo el proceso de desarrollo, teniendo en cuenta que las mismas se desarrollan paralelamente, sin darle a unas mayor o menor importancia.

Las actividades propuestas son las siguientes:

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- Auditorías de los procesos 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 de software se documentan y se manejan de acuerdo con un procedimiento establecido.
6- Medir los procesos.
7- Revisión de las actividades del grupo SQA del proyecto.
8- Entrenamiento del grupo SQA del proyecto.
9-Evaluar los procesos del grupo SQA del proyecto.
10- Gestión de Configuración de los artefactos generados por el grupo SQA del proyecto.

5 Definición de roles para cumplir con las actividades de aseguramiento de la calidad del software.

Se propone que el equipo de aseguramiento de la calidad debe ser diferente al equipo de desarrollo del software, así se detecta con mayor facilidad los posibles errores que puedan afectar la calidad del software. Para el equipo de aseguramiento de la calidad se proponen los siguientes roles y se describe de una forma detallada las actividades que debe cumplir cada persona según los roles definidos:

El jefe de calidad es el responsable de:

1. Hacer entrevistas y visitas a los clientes durante todo el ciclo de desarrollo del software; al culminar esta actividad debe realizar el informe con las necesidades del cliente.
2. Identificar las propiedades de calidad que debe cumplir el software, para eso hacer un listado de las propiedades según las necesidades del cliente y los requisitos funcionales y no funcionales que debe cumplir cada software; e insertarlo en el plan de SQA.
3. Definir las métricas que se utilizarán para medir las propiedades de calidad que se identifiquen; las mismas se insertan en el plan de SQA.
4. Revisar las entregas semanales que hace el equipo de desarrollo del software, debe asegurar que los productos se realicen en la fecha señalada en el plan de proyecto y que además se realicen todos los artefactos vinculados de forma directa con la calidad del producto; esta actividad culmina con un informe de revisión.
5. Revisar la calidad del producto y hacer un resumen de la situación actual del producto que se está revisando; este resumen se introduce en el informe de revisión.
6. En la fase inicial del desarrollo del proyecto el jefe de calidad elabora el plan de SQA y le da un seguimiento sistemático en las restantes fases, sin embargo debe elaborar un plan de prueba para cada fase de desarrollo.

El analista de pruebas es el responsable de:

1- Hacer entrevistas y visitas a los clientes durante todo el ciclo de desarrollo del software.

2- Identificar los métodos de pruebas y los procedimientos de prueba que se realizarán en las diferentes fases de desarrollo; es importante aclarar que este plan de prueba es totalmente diferente al plan de prueba que debe realizar el equipo de desarrollo de software en el flujo de prueba que propone RUP.

El diseñador de prueba es el responsable:

1- Diseñar los casos de pruebas que se realizarán en todo el ciclo de vida del software; para ello construye el documento diseño de casos de pruebas.

2- Elaborar las listas de chequeo que se realizarán en las diferentes fases.

El probador es el responsable de:

1- Ejecutar los casos de pruebas y anotar las no conformidades encontradas durante la realización de las pruebas.

2- Realizar las listas de chequeos definidas por el diseñador de pruebas.

6 Actividades de SQA por etapas

Para obtener y seguir las actividades de ingeniería de software, los proyectos se ven en la necesidad de realizar el Plan de Aseguramiento de la Calidad del Software para controlar dichas actividades. Para llevar a cabo este plan se efectúan diferentes actividades que apoyan el desarrollo del mismo. Estas actividades abarcan todos los puntos y fronteras del plan, donde se le da cumplimiento y seguimiento al proceso de desarrollo de los proyectos, de tal forma que el producto final cumpla con la calidad requerida. Cada una de estas actividades es un paso importante en el plan de aseguramiento de la calidad del software propuesto por RUP.

A continuación se muestra el orden que debe seguir cada etapa.

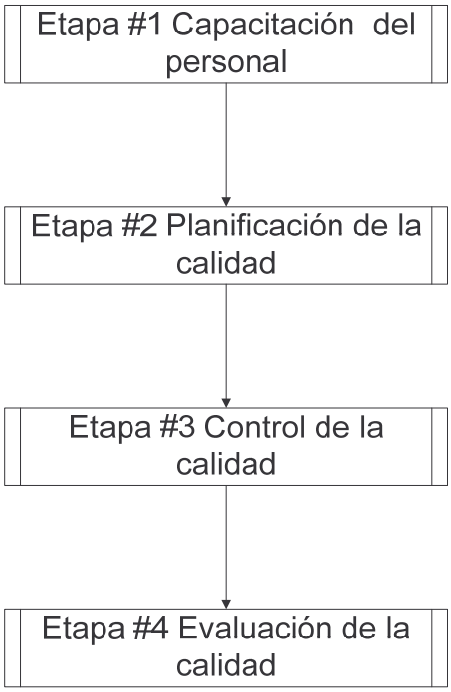


Figura1: Etapas propuestas para el aseguramiento de la calidad.

3.2. Etapas propuestas para el aseguramiento de la calidad.

3.2.1. Capacitación del grupo SQA del proyecto.

Descripción de la etapa

Los integrantes de los proyectos de la facultad varían de un año a otro, por lo que es necesario entrenar a los que ingresan al proyecto y que pertenezcan al grupo SQA. Para llevar a cabo esta actividad se debe definir un Plan de Entrenamiento para el proyecto y se deben crear cursos que sean obligatorios donde se abarquen los temas de calidad. El personal encargado de las actividades de aseguramiento de la calidad debe tener amplio conocimiento en diferentes materias, por lo que se proponen algunos temas en los que pueden ser entrenados:

- Las responsabilidades que tiene una persona a la hora de ocupar un rol determinado, como parte del grupo SQA del proyecto.
- Cómo desarrollar las pruebas al software.
- Cómo medir los procesos a través de métricas.
- Cómo llevar a cabo los procesos de revisiones, revisiones técnicas formales y auditorías.

- Cómo identificar las desviaciones resultantes de los procesos de revisiones, revisiones técnicas formales y auditorías.
- Cómo evaluar los procesos a través de métodos de evaluación.

Sus miembros deben tener como características:

- Experiencia en el desarrollo de software.
- Conocimiento de las metodologías de desarrollo y de los métodos y técnicas de control de calidad.
- Capacidad de comunicación oral y escrita.
- Capacidad de interrelación personal.
- Capacidad de hacer frente a problemas.
- Capacidad de diálogo.

Artefacto de salida

Plan de Capacitación.

Plan de capacitación: se utiliza para controlar todas las actividades que se orienten con un objetivo de capacitación, y poder tener la relación de cual es el estado de los integrantes del proyecto con respecto a estas actividades.

3.2.2. Planificación de la calidad

Descripción de la etapa

En esta etapa el jefe de calidad y el analista de pruebas deben entrevistarse con los clientes para identificar las necesidades de los mismos, esta actividad culmina con un listado de necesidades del cliente. De esta forma se puede seleccionar y clasificar las propiedades de calidad, los requisitos funcionales y no funcionales que debe cumplir el software e insertarlas en el Plan de SQA. Identificar los mecanismos de control de calidad e insertarlos en el plan de pruebas. Se debe tener claro que este plan de pruebas es realizado por el equipo de SQA y es totalmente diferente al plan de prueba que defina el equipo de desarrollo del software.

Como esta etapa sirve para evitar la introducción de errores, en este procedimiento se propone como una de las principales actividades de tipo organizativas:

- Asegurar que se lleven a cabo todos los procesos vinculados al desarrollo de software de una forma eficiente, para ello se define como procesos en cada fase de desarrollo los flujos de trabajo que propone RUP para realizar en cada una de estas fases, y que los procesos de Gestión de configuración, Pruebas y Gestión de riesgos se deben llevar a cabo en las 4 fases de desarrollo.
- Asegurar que se utilicen estándares de desarrollo de software para cada una de las fases, para la fase de Inicio: estándares de documentación; fase de elaboración: estándares de diseño; fase de construcción: estándares de implementación y por último en la fase de transición: estándares de documentación técnica para el usuario. Además el equipo de SQA debe guiar y asesorar al equipo de desarrollo de SW para la selección y la correcta aplicación de técnicas y herramientas de ISW.
- Realizar Informe de actividades. Este informe tiene como objetivo identificar las diferentes actividades que se realicen a lo largo de todo el ciclo de desarrollo del SW para asegurar la calidad del producto, estas actividades son responsabilidad del equipo de aseguramiento de la calidad, para ello se identifica la actividad, el tipo de actividad, el responsable de realizarla, el alcance de la actividad, es decir, a quién o para quién va dirigida, los resultados esperados de dicha actividad y por último, una vez concluida la actividad se elabora un resumen de los resultados alcanzados después de desarrollar la actividad.

Artefactos de salida

Plan de Aseguramiento de la Calidad.
Plan de Pruebas.
Informe de Actividades.

- **Elaborar el Plan de SQA**

Establecimiento del Plan de Aseguramiento de la Calidad del Software

Para adaptar las directrices marcadas por los sistemas de calidad a cada proyecto se debe generar un plan específico a cada uno de ellos. El proceso de aseguramiento de la calidad del software comienza con la confección del Plan de Aseguramiento de la Calidad. Este plan es generado por un grupo de aseguramiento de un proyecto y sirve como guía para el desarrollo e institución de las actividades de aseguramiento de la calidad. Este plan se desarrolla durante la planificación del proyecto, en la fase de Inicio y es revisado por todas las partes interesadas del mismo, llevándose conjuntamente con el Plan

de Desarrollo del Software, enriqueciéndose si es necesario durante el ciclo de vida del proyecto, en dependencia de la complejidad y características del mismo.
Un plan de aseguramiento debe contener por lo general:

- Objetivos de la calidad del proyecto y su enfoque para su consecución.
- Documentación referenciada en el plan.
- Gestión del aseguramiento de la calidad.
- Documentación de desarrollo y de control o gestión.
- Estándares, normas o prácticas que hay que cumplir.
- Actividades de revisión y auditorías.
- Gestión de la configuración del software.
- Informes de problemas
- Pruebas
- Herramientas, técnicas y métodos de apoyo.
- Recogida, mantenimiento y almacenamiento de datos sobre la documentación de actividades de aseguramiento de la calidad realizadas en el proyecto.

El Plan de Aseguramiento de la Calidad del Software (PSQA), es un artefacto que provee una visualización clara de cómo debe ser seguro un producto, cuáles los artefactos, y obtiene como resultado la calidad del proceso. Contiene el Plan de Revisiones y Auditorías. Es mantenido durante todo el proceso de desarrollo del software [20].
El objetivo que se persigue con la realización del Plan de Aseguramiento de la Calidad del Software (PSQA), no es más que presentar un modelo para sistematizar y planificar todas las acciones necesarias para proveer la confianza adecuada según los requerimientos técnicos establecidos por el proyecto. Con este plan no se pretende otra cosa que el aseguramiento del producto de software que se está implementando.

Para el formato del plan de SQA se utilizó la plantilla del estándar IEEE y el mismo tiene la siguiente estructura:

1- Propósito: redactar el propósito que tiene el Plan de Aseguramiento de la Calidad.

2- Acrónimos y Abreviaturas: definir las diferentes abreviaturas o acrónimos que se utilizarán en el plan para una mejor comprensión de todas las personas que lo lean.

3- Referencias: referenciar la bibliografía que se utilice para realizar el plan.

4- Gestión: especificar los elementos de las organizaciones que tienen influencia sobre la calidad del software, la lista de tareas cubiertas por este plan (incluye la planificación de la calidad, e identificar las propiedades de calidad que se deben medir) y las responsabilidades por cada tarea.

Organización: el jefe de calidad es el responsable de realizar la gestión que asegura que el proceso establecido sea realmente implantado y que los productos de ese proceso cumplan con los criterios de calidad establecidos en este plan. La gestión de calidad es una disciplina de gestión, junto con Gestión de proyecto y Gestión de la configuración del software. Las disciplina de gestión brinda soporte a las disciplinas ingenieriles (Modelación del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y se realizan en forma paralela a ellas.

Tareas y responsabilidades

Tareas: en esta sección se describen las tareas de aseguramiento de la calidad a realizar, indicando para cada una el entregable asociado.
Responsabilidades: el equipo de SQA es el responsable de realizar las actividades y entregables. Como parte de las actividades se revisarán los productos que se consideren relevantes para la calidad del producto y del proceso.

5. Documentación: especificar los documentos que dirigen el desarrollo del proyecto y que deberán ser revisados como parte de las actividades de aseguramiento de la calidad.

6. Estándares y Métricas: identificar los estándares utilizados en el proyecto (documentación, codificación, prueba, diseño, etc.). Para cada entregable que se revise se debe indicar el estándar o plantilla que se utiliza para su creación. Detallar las métricas de calidad que se aplicarán al producto y proceso.
Estándares: para cada entregable que se entregue en las distintas etapas del software se debe identificar el estándar que se utiliza.

Estándar	Ubicación	Comentarios
IEEE 828-1998		Gestión de la configuración
IEEE 730-1998		Para el aseguramiento de la calidad de software
SO/IEC 9126		Para definir los objetivos de calidad y métricas
ISO 9003		Dirigidas al control final y las pruebas.
ISO 1901-2002		Revisiones y auditorías
RUP		Para el proceso de desarrollo de software
LMC (emitidos por la Dirección Central SW)		Se aplica durante el proceso de desarrollo de software

Métricas: las métricas que se utilizarán para medir las propiedades de calidad.

Las métricas del software se refieren a un amplio elenco de mediciones para el software de computadora. La medición se puede aplicar al proceso de software, proyecto, calidad del producto entre otras con el intento de mejorarlo sobre la base continua. En esta sección se describirán un conjunto de métricas que se usarán para el monitoreo del producto.

Eficacia en la eliminación de defectos (EED)*:* Medida de la actividad de filtrar de las actividades de aseguramiento de la calidad y de control. Se define de la forma siguiente:

EED = E / (E + D) donde E es el número de defectos encontrados antes de entregar al usuario final y D es el número de defectos encontrados después de la entrega. El valor ideal de EED es 1. Esto significa que no se han encontrado defectos en el software. EED será utilizada dentro del proyecto para evaluar la habilidad del equipo para encontrar errores antes de que pasen a la siguiente actividad. EED se vuelve a definir como:

EED (i) = E (i) / (E (i) + E (i + 1)) donde E (i) es el número de errores encontrados en la actividad i y E (i + 1) los errores encontrados en durante la actividad i + 1. Para el caso de este proyecto la actividad i estaría representada por las RTF y las actividades i + 1 por las pruebas tanto internas como de liberación por calidad central.

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.

Impacto de cambio: Cual es la frecuencia de impactos adversos después de una modificación. El método de aplicación es contar el número de impactos adversos después de la modificación y comparar esto con el número de modificaciones realizadas. Se define de la siguiente forma:

X = 1 – A / B donde A es número de impactos adversos detectados después de la modificación y B el número de aplicaciones realizadas.

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.

Tiempo: Tiempo real dedicado a cada una de las tareas.

7. Revisiones: en esta sección se definen los tres tipos de revisiones (revisión de calidad del producto, revisión de entregas semanales y revisión técnica formal), sus objetivos y mecanismos.

Revisión de calidad del producto: su objetivo es revisar los productos que se definieron como claves para asegurar la calidad. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.
Se revisan los productos para verificar que cumplan con los estándares y con los objetivos de calidad.
Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentran alguna no resuelta, debe ser incluida en la siguiente revisión. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.
Como salida se obtiene el informe de revisión, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables del producto y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

Revisión de entregas semanales: Su objetivo es revisar los artefactos que se generan semanalmente. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.
Se revisan los productos para verificar que cumplan con los estándares definidos anteriormente y con los objetivos de calidad. Como salida se obtiene el informe de revisión.

Revisión técnica formal (RTF): el equipo de SQA participa en las RTF como una forma de controlar la calidad del producto. Es un proceso de revisión riguroso. Su objetivo es descubrir errores en la función, lógica ó la implementación de cualquier producto del software; llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo; verificar que el software bajo revisión alcanza sus requisitos, garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos., señalando las posibles desviaciones detectadas, conseguir un software desarrollado de forma uniforme, hacer que los proyectos sean más manejables. Por estas características se adopta esta práctica para productos que son de especial importancia.
En la reunión participan el jefe de calidad e integrantes del equipo de desarrollo. Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado. Como salida se obtiene el informe de RTF.

8. Verificación: identificar las propiedades de calidad que no se hallan verificado, y aquellas que no estén implícitas en la sección de planificación de la calidad. Para cada una de esas propiedades definir y detallar las pruebas a realizar para asegurar su cumplimiento.

9. Reporte de problemas y acciones correctivas: indicar de que manera y a quienes se reportarán los problemas encontrados en los productos y en el proceso. Indicar como se realizará el seguimiento de los problemas. Indicar de quien es la responsabilidad de tomar acciones correctivas para cada problema. Es decir, se debe detallar cuál es el procedimiento para información y seguimiento de errores entre el equipo de SQA y el equipo de desarrollo.

***Procedimiento para información y seguimiento de errores**:* detallar el funcionamiento de dicho procedimiento; una vez que se realicen las diferentes revisiones con sus respectivas pruebas y se detecten los errores, estos son anotados en la Plantilla de las No Conformidades, dichas no conformidades se le entregan al equipo de desarrollo del software, los mismos corrigen los errores y se vuelve a entregar el producto al equipo de aseguramiento de la calidad para una nueva revisión, y se sigue los mismos pasos hasta que el producto revisado no tenga errores. Para informar al equipo de desarrollo de software de las posibles no conformidades detectadas se utiliza una reunión de revisión.

10. Herramientas, técnicas y metodologías: especificar las herramientas que se han utilizado para la realización de este plan.

11. Gestión de configuración: El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de GCS y que se realizan tal como están establecidas en el proceso. Incluye identificar los métodos para la gestión de la configuración. Estos métodos se utilizarán para mantener, almacenar, asegurar y documentar las versiones controladas identificadas en las fases de desarrollo, también se utilizarán para proteger el almacenamiento adecuado de los programas, documentación, etc., así como también la prevención de acceso sin autorización, daño, etc., lo cual será definido en conjunto con el responsable de gestión de la configuración del software.

Actividades para asegurar la calidad: Se deberán explicitar en esta sección las actividades que llevará a cabo el jefe de calidad para asegurar que la gestión de configuración se realiza como lo establece el proceso definido para el proyecto.

Se proponen algunas actividades mínimas que se deberían realizar.

- Asegurar que se generó la línea base del proyecto en el momento establecido en el modelo de proceso.
- Asegurar que la línea base del proyecto generado es correcta.
- Se verifica periódicamente que el responsable de gestión de la configuración del software mantiene apropiadamente el control de la línea base, así como el registro completo de cambios para requerimientos, diseño, código, verificación y documentación.
- Se monitorean los procedimientos del Comité de control de cambios para verificar que son efectivamente realizados como se especificaron en el Plan de configuración.

12. Gestión de riesgos: el objetivo del SQA en esta área es asegurar que la gestión de riesgos se realiza y se hace de forma correcta. Se indican los métodos y procedimientos que serán utilizados para identificar, monitorear y controlar los riesgos identificados en el proyecto.

***Actividades para asegurar la calidad**:* Para evaluar que se cumple el objetivo planteado se deberán realizar actividades de SQA, se proponen las siguientes actividades:

- Asegurar que se realice una evaluación de riesgos en el proyecto.
- Que se evalúen riesgos en todas las áreas del proyecto y por personas diferentes.
- Que esa evaluación de riesgos se haga en forma individual y luego se reúnan los riesgos detectados para determinar el impacto de cada riesgo, con todos los integrantes del equipo de trabajo.
- Que se prioricen los riesgos.
- Que se planifique su eliminación, mitigación y/o contingencia.
- Que se incorporen los riesgos a los planes del proyecto para su seguimiento.
- Asegurar que se realice el seguimiento de los riesgos.

➢ Elaborar el Plan de Pruebas.

El analista de prueba es el responsable de analizar el Plan de Pruebas. Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Se debe especificar el alcance de la prueba para cada fase.

También identifica las pruebas a realizar en cada fase así como los mecanismos de pruebas y el tipo de prueba. El jefe de calidad elabora el plan de prueba. A continuación se propone el comportamiento de las pruebas en cada fase:

Inicio: el desarrollo del prototipo exploratorio de demostración; se debe realizar pruebas a la documentación y no está de más ejecutar algunas pruebas exploratorias al primer prototipo de interfaz de usuario que se obtenga.

Elaboración: probar los componentes ejecutables que se han implementado y que deben corresponderse con la arquitectura básica de la aplicación.

Construcción: desarrollar los casos de prueba y procedimientos de prueba para hacerlos.

Transición: el producto en su entorno de operación por lo que es probado por usuarios reales.

Niveles de prueba: la prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo, definir en qué nivel se realizará las pruebas teniendo en cuenta las diferentes fases de desarrollo y los artefactos que se probarán. Se proponen tener en cuenta los siguientes niveles de pruebas:

Prueba independiente: es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva diferente y en un ambiente más rico que los desarrolladores.

Prueba de integración: es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación.

Prueba de sistema: son las pruebas que se hacen cuando el software está funcionando como un todo.

Prueba de aceptación: prueba de aceptación del usuario, es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Métodos de prueba: definir los métodos de prueba que se utilizarán; se propone utilizar los métodos de caja blanca y de caja negra.

-La prueba de la caja blanca del software comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar

el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

-La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software; o sea, 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.

3.2.3. Control de la calidad

Descripción de la etapa

Para llegar a establecer los requisitos establecidos se necesita controlar y corregir el trabajo que se está realizando, pero se propone no esperar a la última fase de desarrollo para corregir los errores sino realizar auditorías durante todas las fases de desarrollo, así como revisiones de entregas semanales para controlar que los diferentes artefactos se están realizando en la fecha trazada y cumplen con los principios definidos, también se propone hacer una revisión de calidad del producto. Al culminar cada fase de desarrollo, se propone al equipo de desarrollo del SW realizar una revisión técnica formal y esta revisión se realiza a los artefactos que influyen notablemente en la calidad del producto. Si el artefacto no es probado no se puede pasar a la siguiente fase y se vuelve a hacer la revisión hasta que el producto presente la calidad requerida. Todas estas revisiones culminan con un informe de revisión.

Es importante aclarar que la RTF la realiza el equipo de desarrollo del SW, el equipo de SQA solo participa en este tipo de revisión para controlar la calidad del producto y debe anotar las desviaciones que se detecten e informarles.

Diseño de casos de pruebas (caja negra): es responsabilidad del diseñador de pruebas elaborar los casos de pruebas pero el probador ejecuta todos aquellos casos diseñados y llena la plantilla de diseño de casos de pruebas, en el caso de que se estén realizando pruebas de caja negra por cada caso de uso que exista debe existir un caso de prueba conformados por las posibles clases válidas e inválidas que proponga el diseñador de pruebas.

Diseño de casos de pruebas (caja blanca): es responsabilidad del diseñador de pruebas elaborar los casos de pruebas pero el probador ejecuta todos aquellos casos de pruebas. Cuando se esté realizando las pruebas de caja blanca se propone seguir la técnica del camino básico, la misma permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño y

usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Primeramente se debe calcular la complejidad ciclomática que es la que brinda la cantidad de casos de pruebas que se deben realizar, luego se sacan los caminos básicos que ofrece los principales caminos a seguir y los parámetros que se deben probar, y por último se ejecutan los casos de prueba.

Artefactos de entrada y salida

Entrada
Expediente del Proyecto.

Salida
Informe de Revisión.
Listas de Chequeo.
Plantilla Diseño de Casos de Pruebas.
Plantilla de No Conformidades.

Expediente del Proyecto: se recoge toda la documentación perteneciente al proyecto, tanto técnica, como de prueba, contratos con el cliente y actas de reuniones.

Realizar Informe de Revisión: el informe de revisión identifica el artefacto que se está revisando, el responsable de la elaboración de dicho artefacto y la fecha de entrega, además se llena la columna de observaciones donde el jefe de calidad inserta todas las dificultades que detecta en dicho artefacto para luego cuando se realice la RTF con el equipo de desarrollo del SW se analice si el artefacto es aprobado o no y el mismo puede determinar si se comienza con la siguiente fase o si se debe seguir en la fase actual hasta que dicho artefacto esté aprobado.
Es el documento encargado de describir de forma específica como el equipo de calidad realizará las revisiones. Su planificación, flujos de trabajo, herramientas y funcionamiento de los roles, deberá quedar muy bien definido.

Realizar Listas de Chequeo (ver anexo 7): es responsabilidad del diseñador de prueba elaborar las listas de chequeo pero el probador es quien la realiza, la misma tiene como objetivo chequear la calidad de un determinado producto, y comprobar si los objetivos definidos para cada fase se están

cumpliendo satisfactoriamente; se debe utilizar para chequear los productos que determinen de una forma importante la calidad del software.

Plantilla de Diseño de Casos de Prueba (ver anexo 8): se describen los elementos que serán cubiertos por los casos de pruebas, como se medirá en cada uno de ellos y de qué manera. Se especifican cuáles son los objetivos del caso de prueba así como la relación que se establece entre estos. Se listan los elementos que se someterán a prueba, los recursos que serán necesarios para la correcta aplicación de las pruebas así como las personas responsables del diseño y aplicación de estas pruebas.

3.2.4. Evaluación de la calidad del producto

Descripción de la etapa

La evaluación de la calidad del producto se realizará una vez culminada la fase de transición. Si el producto es aceptado por el usuario final, el equipo de SQA se reúne para dar una evaluación final de la calidad del software desarrollado culminando con esta actividad el procedimiento propuesto.
El equipo de SQA para dar la evaluación final debe tener en cuenta los resultados de las diferentes auditorías realizadas en cada fase de desarrollo, si el software desarrollado satisface las necesidades del cliente, y si además cumple todas sus expectativas, seguidamente se analiza si el producto tiene concordancia con los requerimientos explícitamente establecidos y con los estándares de desarrollo definidos, así como el ajuste del producto con los procesos involucrados en el desarrollo del software.
Seguidamente se propone utilizar Listas de Chequeos para evaluar el cumplimiento del software con la sub-características de las características de la calidad especificadas en el plan de SQA. Finalmente el equipo de SQA se reúne y teniendo en cuenta los aspectos mencionados anteriormente, se evalúa la calidad del producto de baja, media, o alta, ésta actividad concluye con un informe de evaluación.

Artefactos de entrada y salida

Entradas
Informe de Revisión.
Listas de Chequeo.
Plantilla Diseño de Casos de Pruebas.

Plantilla de No Conformidades (ver anexo 9).

Salidas
Informe de Evaluación.

Realizar Informe de Evaluación: en este informe se hace un resumen de forma general señalando los aspectos que se analizaron para evaluar la calidad del producto, así como los resultados satisfactorios que haya logrado el equipo del proyecto. También se le hace recomendaciones al equipo de desarrollo.

3.3. Cumplimiento de los lineamientos mínimos de calidad definidos por la universidad.

En las etapas propuestas se evidencia el cumplimiento de algunos lineamientos mínimos de calidad definidos por la UCI. Dentro de estos lineamientos se define que los miembros del equipo deben recibir superación antes o durante del proceso que les permita enfrentarse al problema en cuestión con una visión amplia y con un número mayor de herramientas. Este aspecto se cumple en la etapa de capacitación del grupo de SQA, donde los integrantes del mismo deben tener conocimientos básicos relacionados con el tema de calidad. El lineamiento que plantea que se debe llevar un registro de las actividades conjuntas y visitas de los diferentes factores al proyecto se cumple en todas las etapas del procedimiento propuesto, ya que cada actividad que se realiza debe estar registrada, así como el responsable de realizarla. Para la realización de estas etapas se definen un conjunto de roles que serán los encargados de llevar a cabo las actividades de SQA; y la UCI dentro de los lineamientos que propone, dice que debe quedar definida y documentada una estructura organizacional que permita identificar y agrupar las tareas requeridas, seleccionar y establecer las estructuras organizativas, definir roles y relaciones de autoridad en el proyecto, así como que debe realizarse un estudio de los roles que serán desempeñados en el proyecto y de las personas a las cuales serán asignados dichos roles, teniendo en cuenta responsabilidades y objetivos que debe cumplir este rol, así como la formación y los conocimientos que posee la persona.
Otros de los lineamientos plantea que en cada fase del proceso de desarrollo deben ser registrados los defectos detectados, obtener datos sobre estos, analizarlos y determinar métricas que permitan detectar la mejor forma de reducirlos; el mismo es evidenciado en la etapa de control donde se le realizará auditorias y revisiones a los procesos involucrados en el desarrollo del software, donde se debe chequear cada actividad que se propone y deben registrarse los defectos encontrados. Por otra

parte, el lineamiento que plantea definir y documentar en cada etapa los artefactos que deben producirse y una vez obtenidos deben ser evaluados; se cumple en las etapas propuestas, ya que todos los artefactos que se generen deben ser documentados y a través de las RTF se define si el artefacto es aprobado o no. Las actividades de Gestión de configuración cumplen los lineamientos definidos por la UCI ya que aquí se realizan actividades tales como: documentarse todos los cambios realizados y existir una persona que apruebe dicho cambio; estar definidas las líneas base del proyecto, etc. Otros de los lineamientos es que debe existir conocimiento por parte de los miembros del equipo de los modelos de calidad existentes y regirse por algunos de ellos que estará bien definido y documentado y si no se utiliza ningún modelo de calidad debe existir, al menos, definido en el proyecto un plan de calidad; dentro de la propuesta de solución de este trabajo se cumple con estos lineamientos ya que se define el plan de aseguramiento de la calidad el cual se lleva a cabo durante todo el desarrollo del software y las actividades propuestas abarcan las fronteras de este plan. Por último este procedimiento fue basado en la metodología RUP la cual es la más usada en los proyectos de la facultad y cumple con el lineamiento de que el proceso de desarrollo de software debe organizarse basándose en los principios del proceso de desarrollo RUP.

Conclusiones del capítulo 3

- El equipo de desarrollo del software y el equipo de aseguramiento de la calidad deben estar en constante intercambio.
- Para realizar el procedimiento de aseguramiento de la calidad, se debe cumplir con todas las actividades descritas en el plan de aseguramiento de la calidad.
- La preparación del personal de SQA posibilitará una mejora en las actividades de aseguramiento de la calidad, y una mayor calidad del proceso de producción de software.

CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA

Introducción

Para la validación y aceptación de este procedimiento que se presenta en el capítulo 3, se utilizó el criterio de un panel de expertos y el empleo de técnicas que propone el método Delphi. Este panel se conformó con especialistas en los temas de calidad y gestión de software de la UCI

4.1. Método de evaluación de expertos

El método Delphi, considerado como una de las técnicas subjetivas de pronosticación más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de expertos en el tema tratado. El mismo permite rebasar el marco de las condicionantes actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos, fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva.

Es conocido también como Delfos, nombre cuyo origen proviene del oráculo de la antigua Grecia, el método Delphi fue creado alrededor de los años 1963-1964 por la Rand Corporation, y específicamente por Olaf Helmer, Dalkey y Gordon, con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política. Así pues el método Delphi es la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas [21].

La esencia de este método consiste en la organización de un diálogo anónimo entre los expertos consultados individualmente, mediante cuestionarios, con vistas a obtener un consenso general o, al menos, los motivos de la discrepancia, la confrontación de las opiniones se lleva a cabo mediante una serie de interrogantes sucesivas, entre cada una de las cuales la información obtenida sufre un procesamiento estadístico - matemático.

En el presente capítulo se hará la descripción de las técnicas que fueron utilizadas del método y los resultados obtenidos.

4.2. Selección de los expertos

Se entiende por experto, tanto al individuo en sí, como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

4.2.1. Determinación de la cantidad de expertos

Se seleccionaron 7 expertos (ver anexo 10), tomando como criterio de selección la efectividad de la actividad profesional que realizan, la experiencia que poseen de calidad, los años vinculados a la UCI y por su participación en diversos eventos científicos.

Elegir los expertos atendiendo a las características mencionadas propicia obtener resultados con calidad, junto a otras cualidades propias de éstos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

4.3. Elaboración del cuestionario

El cuestionario elaborado consta de 4 preguntas (ver anexo 11). El mismo fue conformado con preguntas abiertas de enfoque investigativo. Se prefirieron estas preguntas, porque tienen la ventaja de proporcionar una mayor riqueza en las respuestas que son brindadas por los especialistas. También se les dio la posibilidad de presentar su opinión general del procedimiento, para que tuvieran la libertad de expresar todo lo que se pudo obviar en el cuestionario.

4.4. Resultados de la evaluación

A partir de los resultados obtenidos del cuestionario se pudo determinar que los expertos estuvieron de acuerdo con el procedimiento propuesto.

En la pregunta relacionada con la aplicación del procedimiento a los proyectos productivos de la Facultad 3 el 100% consideraron que sí se debía aplicar ya que poniendo en práctica este procedimiento se puede asegurar la calidad en cada fase de desarrollo, de forma tal que se puedan corregir errores desde el inicio.

En relación a la pregunta de que si se aplicara el procedimiento propuesto mejoraría la calidad del software que se desarrolla en los proyectos de la Facultad 3, el 100% de los expertos afirmaron que sí, ya que los productos estarían siendo desarrollados de una forma organizada y estandarizada, permitiendo que los errores sean corregidos en edades tempranas del desarrollo; además se tendría

una forma de comprobar que los productos realmente tienen la calidad necesaria para entrar a una fase superior de pruebas.

En el cuestionario se indagó sobre cuáles eran las actividades que deben llevarse a cabo para asegurar la calidad del software. En las respuestas hubo variabilidad, pero se pudieron definir, de todas las propuestas, las actividades que más prevalecieron y que son mostradas en la Tabla 4.1 En ella se pueden apreciar las coincidencias que tuvieron los expertos en sus criterios, marcados con el número 1. Si ellos no involucraron cierta actividad en sus respuestas; o la que citaron no tenía ninguna otra coincidencia, entonces se puso en 0 la casilla correspondiente.

En el gráfico 1, se muestra la totalización de las coincidencias que tuvieron los expertos y a la vez se puede definir el nivel de importancia que estas actividades tienen en la evaluación del procedimiento.

Actividades propuestas	**Exp 1**	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**	**Exp 6**	**Exp 7**	**Total**
Establecimiento de un Plan de SQA para el proyecto	1	1	1	1	1	1	1	7
Participación en el desarrollo de la descripción del proceso de software del proyecto.	1	1	1	0	1	0	0	4
Revisión de las actividades de ingeniería de software para verificar su ajuste al proceso de software definido.	1	1	1	1	0	1	1	6
Auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de software	1	1	1	1	1	1	1	7
Asegurar que las desviaciones del trabajo y los productos de software se documentan y se manejan de acuerdo con un procedimiento establecido	0	0	1	1	0	1	0	3
Medir los procesos	0	1	0	1	0	0	0	2
Revisión de las actividades del grupo SQA del proyecto	1	1	1	1	1	1	1	7
Entrenamiento del grupo SQA del proyecto	1	1	0	0	1	1	1	5
Evaluar los procesos del grupo SQA del proyecto	1	1	1	1	1	1	1	7
Gestión de Configuración de los artefactos generados por el grupo SQA del proyecto	1	0	1	0	1	1	1	5

Tabla 4.1. Actividades propuestas para el aseguramiento de la calidad.

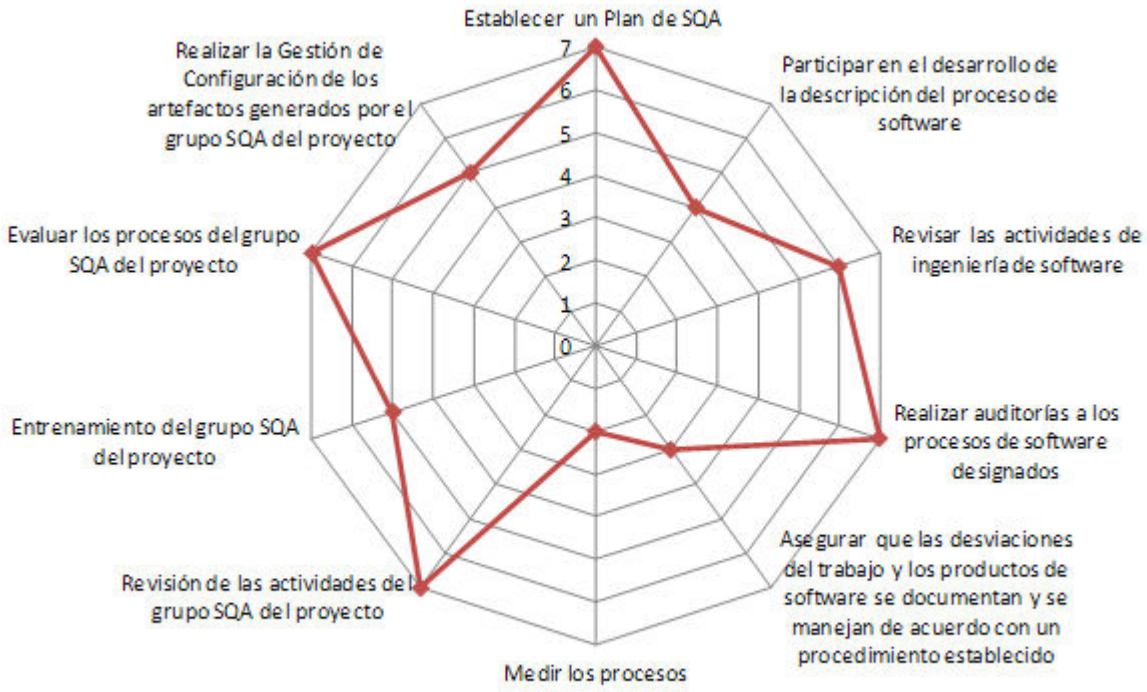


Gráfico 1. Distribución radial de la importancia dada a las actividades de aseguramiento de la calidad.

Como se puede apreciar en el gráfico la mayoría de los expertos coincidieron en las actividades propuestas para el aseguramiento de la calidad. Identificaron como muy positiva las actividades de establecimiento de un plan de aseguramiento de la calidad, ya que de esta forma esta actividad estaría organizada y estandarizada para todos los proyectos de la Facultad 3. También señalaron de forma positiva las actividades de auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de software, las de revisión de las actividades del grupo SQA del proyecto y también las actividades de evaluación de los procesos del grupo SQA del proyecto, donde destacaron que todas estas actividades contribuyen en la detección de posibles errores y fallos que pueda tener el software en su desarrollo.

El panel de expertos le dio importancia además al resto de las actividades, ya que de una forma u otra todas contribuyen a que la calidad del software desarrollado sea satisfactoria y se obtenga de forma organizada.

De forma general los expertos expresaron que la propuesta aportaría numerosos beneficios a los proyectos que integran a la Facultad 3 ya que se desarrollaría software más profesionalmente, se identificaría de una forma más temprana los posibles fallos que tiene el proceso de desarrollo que se lleva a cabo en el proyecto, por lo tanto mejoraría la calidad del software. Dentro de las dificultades que se puede presentar con el procedimiento propuesto, según los expertos, es que solo puede aplicarse en los proyectos que utilicen metodología RUP, porque si surge algún proyecto en la facultad que utilice otra metodología no se podrá aplicar.

Conclusiones del capítulo 4

- La utilización de un método de expertos permite la validación de la propuesta para su aplicación en los proyectos.
- La propuesta fue considera por los expertos como una herramienta de gran utilidad para la realización del proceso de aseguramiento de la calidad de software de los proyectos.

CONCLUSIONES GENERALES

- El proceso de aseguramiento de la calidad del software representa un factor importante dentro del proceso de desarrollo de un proyecto, pues lleva a que se cumplan los requisitos del cliente con eficiencia y eficacia.
- El procedimiento elaborado, basado en la metodología RUP, permitirá desarrollar las actividades de aseguramiento de la calidad de los proyectos productivos organizadamente.
- Los roles propuestos dentro del grupo de aseguramiento de la calidad permitirán un mayor control de las actividades propuestas.
- El Plan de Revisiones y Auditorías es uno de los elementos más importantes del Plan de Aseguramiento de la Calidad, ya que sin él no se lograrían los objetivos del proceso de aseguramiento de la calidad del software.

RECOMENDACIONES

- Realizar un estudio más profundo del aseguramiento de la calidad del proceso de producción de software en la facultad.
- Aplicar y dar seguimiento al procedimiento propuesto en los proyectos productivos de la Facultad 3.
- Realizar una capacitación previa a las personas que van a desempeñar los roles propuestos.
- Adaptar este procedimiento para otras metodologías de desarrollo del software.

REFERENCIA BIBLIOGRAFICA

1. Pressman, R.S., *"Ingeniería del Software, Un enfoque práctico"*. 2005.
2. Crosby, *Quality is free.* 1979.
3. Crosby, *Quality is free.* 1994.
4. Durán Toro, A., *"Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información"*. Departamento de Lenguajes y Sistemas Informáticos 2000.
5. Huacoto (2002-2007) *PROPUESTA PARA IMPLANTAR CMMI EN UNA EMPRESA CON MÚLTIPLES UNIDADES DESARROLLADORAS DE SOFTWARE*.
6. ISO 9000:2000, *Sistema de Gestión de la Calidad - Principios y Vocabulario.* 2000.
7. PÉREZ GARCÍA, R., *Propuesta de modelo de procesos para la producción de software en la UCI.* 2006.
8. LÓPEZ and LTDA, *CALIDAD CON PARTICIPACIÓN - LOS CÍRCULOS DE CALIDAD* 2000-2006a: p. Disponible en :http://www.gestiopolis.com/canales/gerencial/articulos/no%203/qc.htm.
9. Pressman, R.S., *"Ingeniería del Software, Un enfoque práctico"*. 2002.
10. Ferré, X.G., *Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software.* 2005.
11. EJIOGU, L.O., *Software Engineering with Formal Metrics.* 1991.
12. Febles, A.E., *"Calidad de Software y la empresa, enseñanza de un tema imprescindible para el Ingeniero Informático"* 2000.
13. Delgado Dapena, *Informática Evento virtual 2007, 2007* 2007: p. Disponible en : http://www.informaticahabana.com/evento_virtual/?q=node/583.
14. Normalización, *NORMAS CUBANAS VIGENTES AL CIERRE DE ABRIL, 2007.*
15. Febles, A.E., *III Taller de Calidad en las Tecnologías de la Información y las Comunicaciones.* 2007.
16. Delgado Martínez, R., *Calidad del Software.* 2002.
17. *Lineamientos mínimos de calidad.* 2007: p. disponible en http://calidadsoft.prod.uci.cu.
18. Piñero, P., *Sitio de Producción e Investigación de la Facultad 3.* 2005: p. Disponible en: http://facultad3.uci.cu/ProdF3v2.
19. Hernández León, R.A., *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA.* 2002.
20. RUP, *Rational Unified Process®. Versión* 2003.
21. GORDON, H., DALKEY, *Método Delphi.* 1963-1964: p. Disponible en: http://www.12manage.com/methods_helmer_delphi_method_es.html.

BIBLIOGRAFÍA

Crosby, P., Quality is free, McGraw-Hill, 1979

[Crosby, P., Quality is free, McGraw-Hill, 1994

Toro Durán Amador: "*Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", Tesis Doctoral, septiembre 2000. Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla.*

Delgado Martínez, Ramsés. Calidad del Software. Universidad de las Ciencias Informáticas, Cuba. 2002.

EJIOGU, L. O. Software Engineering with Formal Metrics. 1991. p.

Febles Estrada Ailyn,"Calidad de Software y la empresa, enseñanza de un tema imprescindible para el Ingeniero Informático", Instituto Superior Politécnico "José Antonio Echeverría, 2000.

Febles Estrada Ailyn y otros, "III Taller de Calidad en las Tecnologías de la Información y las Comunicaciones", Universidad de las Ciencias Informáticas, 2007.

Ferré Grau, Xavier; *Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software.* Tesis Doctoral. 2005.

Fernández, L. and M. I. Alarcón (1999). "Necesidades de medición en la gestión y aseguramiento de calidad del software." http://www.sc.ehu.es/jiwdocoj/remis/docs/aseguracal.htm

GORDON, H, DALKEY. Método Delphi, 1963-1964. [Disponible en: http://www.12manage.com/methods_helmer_delphi_method_es.html.

Humphrey Watts S, "Introducción al Proceso de Software Personal (PSP)". Addison-Wesley. 2002.

HUACOTO, N. E. C. PROPUESTA PARA IMPLANTAR CMMI EN UNA EMPRESA CON MÚLTIPLES UNIDADES DESARROLLADORAS DE SOFTWARE, 2002-2007.

Pressman, R.S: Ingeniería del Software. Un enfoque práctico. Mc Graw Hill, 2002.

Pressman Roger S. "Ingeniería del Software, Un enfoque práctico", Félix
Varela, La habana, 2005.

Piattini , García, "Calidad en el desarrollo y mantenimiento del software", RA-MA Editorial, Madrid, 2003.

PALACIO, J. Sinopsis de los modelos SW-CMM y CMMI, 2006.

RENIER PÉREZ GARCÍA, R. D., AILYN FEBLES, YADENIS PIÑERO, LISSETTE SOTO, YAMILIS FERNÁNDEZ, AILÍA PARRA. Propuesta de modelo de procesos para la producción de software en la UCI, 2006.

PIÑERO, P. Sitio de Producción e Investigación de la Facultad 3, 2005. [Disponible en:

http://facultad3.uci.cu/ProdF3v2

Rational Unified Process ®. Versión 2003.06.00.

EMA, E. La evolución desde la gestión convencional a la gestión moderna del software (CMM y CMMI), 2005.

MARTHA DELGADO DAPENA, Y. F. P. Informática Evento virtual 2007, 2007. [Disponible en: http://www.informaticahabana.com/evento_virtual/?q=node/583

NORMALIZACIÓN. NORMAS CUBANAS VIGENTES AL CIERRE DE ABRIL 2007, 2007.

LÓPEZ, C. and W. LTDA. CALIDAD CON PARTICIPACIÓN - LOS CÍRCULOS DE CALIDAD, 2000-2006. [2006]. Disponible en: http://www.gestiopolis.com/canales/gerencial/articulos/no%203/qc.htm

WIKIPEDIA, C. D. Norma de calidad, 2007. [Disponible en: http://es.wikipedia.org/w/index.php?title=Norma_de_calidad&oldid=8009615

GUERRERO, L. A. Calidad, 2006.

MANUEL DE LA VILLA, M. R. E. I. R. *Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo*, 2004.

ANEXOS

Anexo1: Entrevista realizada a los responsables de calidad en los proyectos de la facultad 3

1- ¿Existe alguna forma de asegurar la calidad en el proyecto? ¿Cuál?

2- ¿Cuáles son los pasos para asegurar la calidad?

3- ¿Cómo puedes asegurar que lo has hecho correctamente?

4- ¿Qué problemas existen en cuanto al aseguramiento de la calidad?

Anexo 2: Encuesta aplicada a los proyectos productivos de la facultad 3.

Facultad 3 Grupo _____

Nombre del proyecto _____ Rol _____

1. **¿Conoce el concepto de aseguramiento de la calidad? :** Si__ No___

a) De los siguientes conceptos cual escogería:

___ Conjunto de características de un producto que satisfacen las necesidades de los clientes y en consecuencia hacen satisfactorio el producto.

___ Es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad.

___ Es un conjunto de prácticas que se ejecutan con un propósito determinado, las cuales transforman entradas en salidas, que son de valor para el cliente. El proceso puede incluir herramientas, métodos, materiales y/o personas.

___ Es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad.

2. **¿Durante qué etapas se desarrollan las actividades de aseguramiento de la calidad en su proyecto?**

__ No se realiza

__ Al inicio del proceso del software

__ Todo el proceso de software

__ Al final del proceso del software

3. ¿Cree que es importante tener un sistema de aseguramiento de la calidad en el proyecto? Argumente su respuesta.

4. **La calidad del producto que se fabrica es:** E__ B__ R__ M__

5. ¿En el proyecto los integrantes realizan las tareas a través de los roles?

Si__ No___ No sé___

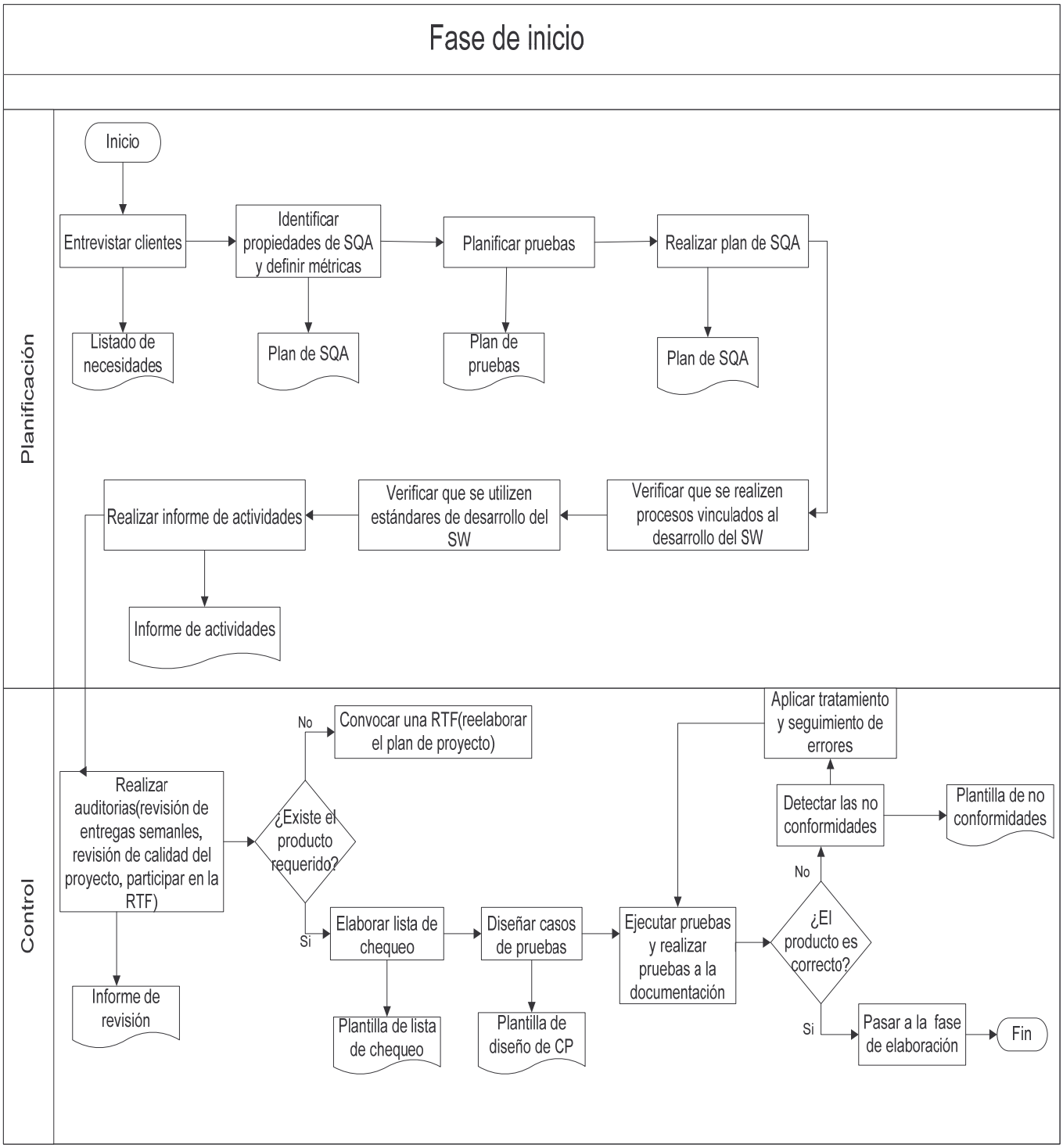
6. **¿Conoce algún modelo o estándar que se utilice a nivel mundial para la gestión de la calidad?** Si__ No__ No sé __

a)Si su respuesta es Sí, menciónelos ¿Utiliza algunos de estos en el proyecto?

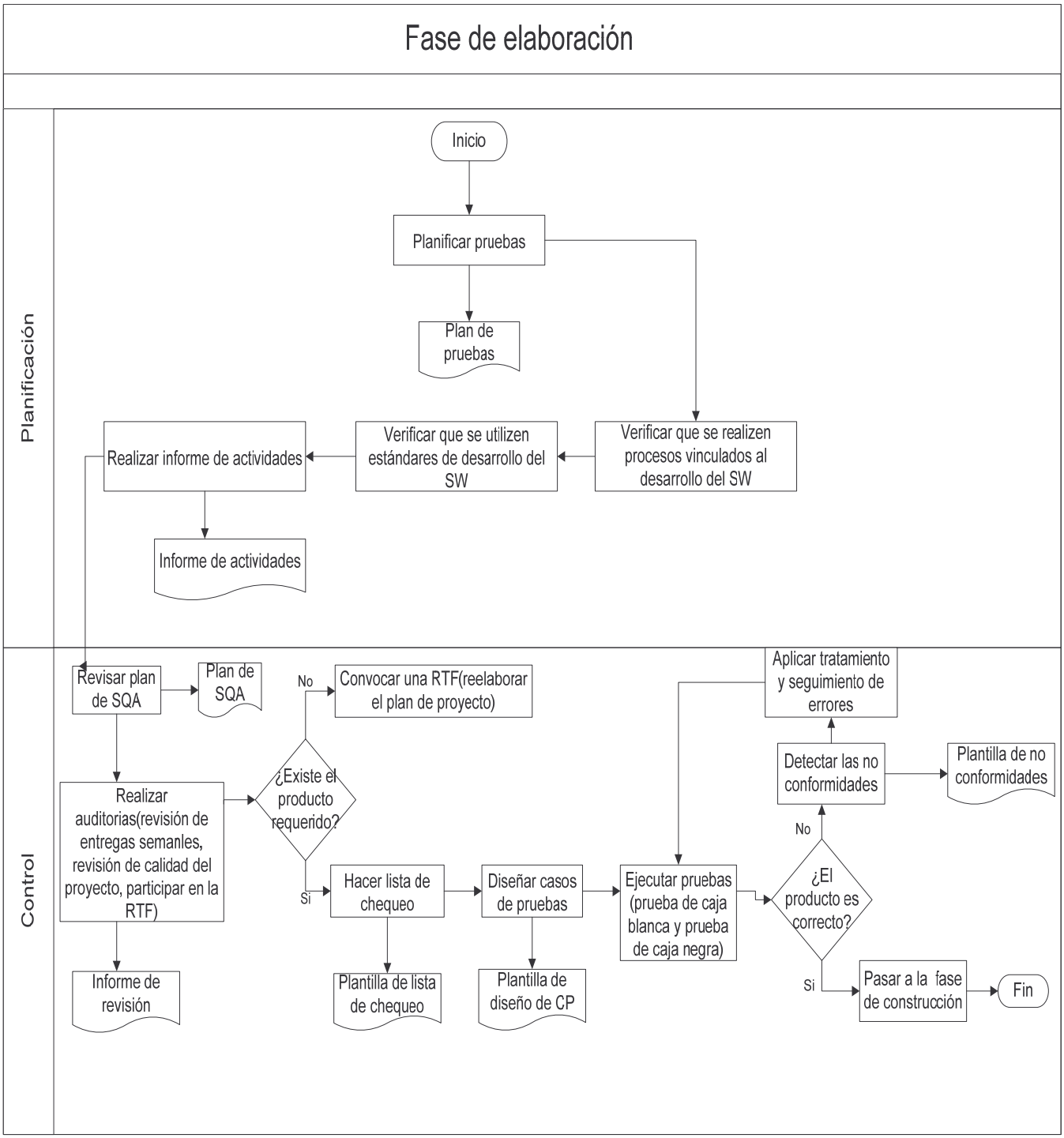
Si__ No__ No sé__

b) Si su respuesta es No, mencione cómo garantizan la calidad del software.

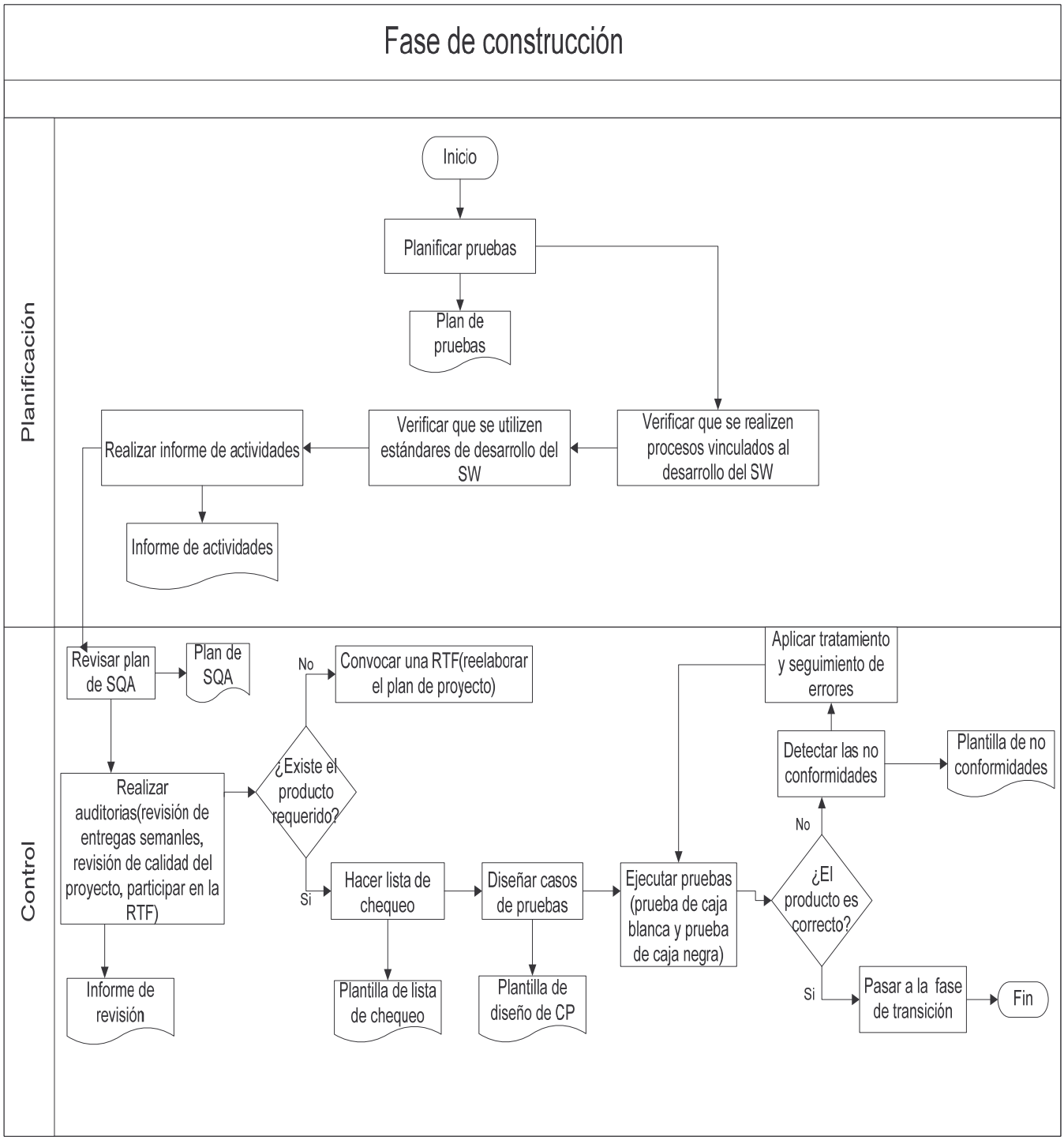
Anexo 3: Principales actividades de SQA en la fase de inicio.



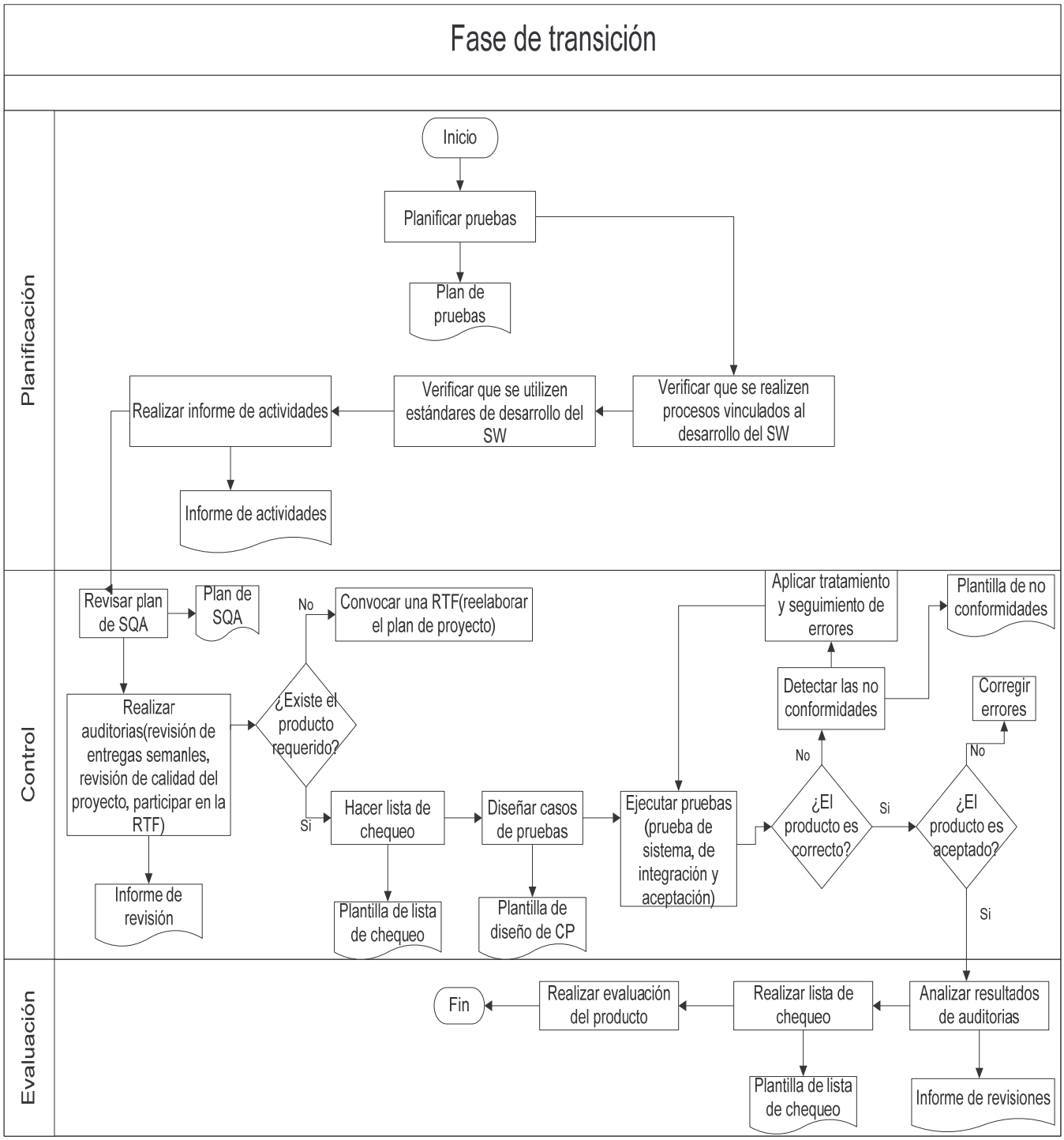
Anexo 4: Principales actividades de SQA en la fase de elaboración.



Anexo 5: Principales actividades de SQA en la fase de construcción.



Anexo 6: Principales actividades de SQA en la fase de transición.



Anexo 7: Plantilla de lista de chequeo

Lista de chequeos: Insertar la funcionalidad de la lista de chequeos **Fecha:** **Forma:_____ Descripción:_____** **Analista:_____ Probador:_____**				
Revisión: Insertar lo que se esta revisando				
Nivel	**Actividad**	**E**	**NP**	**Comentario**
Importancia del aspecto a evaluar	Lo que se desea chequear, se realiza en forma de pregunta	Evaluación	No de procedencia	Es obligatorio en las respuestas negativas

Las evaluaciones serán:

-- Malo- propiedad no disponible (0)
- Satisfactorio- propiedad parcialmente disponible (2)
+ Bien- propiedad disponible (4)
++ Excelente- propiedad muy bien implementada (5)

El peso de la pregunta será:
!! Muy importante (5)
! Menos importante (3)

Anexo 8: Plantilla del diseño para casos de pruebas

Caso de prueba				
Fecha				
Resultado:				
Observaciones				
Responsables de las pruebas				
Clases válidas	**Clases inválidas**	**Resultado esperado**	**Resultado de la prueba**	**Observaciones**
[se describe detalladamente la entrada y la acción que evidenciará el comportamiento correcto del sistema]	[se describe detalladamente la entrada y la acción que evidenciará el comportamiento incorrecto del sistema]	[se describe detalladamente como se espera que funcione el sistema antes esta entrada específica, hasta esta columna se hace la etapa de diseño de la prueba]	[durante la aplicación de la prueba se describe detalladamente que ha ocurrido realmente, como ha reaccionado el sistema]	[se recogen los detalles que se escapen o ajusten en las columnas anteriores]

Anexo 9: Plantilla de no conformidades

Elemento de configuración	No de la no Conformidad	Descripción de la no Conformidad	Aspecto correspondiente a su localización	Importancia	Recomendaciones
[elemento de configuración del que se describe las no conformidades, pueden existir varias sobre el mismo elemento]		[se especifica de manera clara y precisa en que consiste la no conformidad, utilizando un lenguaje técnico y claro, no se debe hablar de manera general,]	[se especifica de manera clara y precisa en que lugar fue detectada la no conformidad]	[se señala con una x si la no conformidad es importante]	[se señala con una x si la no conformidad es una recomendación]

Anexo 10: Caracterización de los expertos.

Expertos	Graduado de	Años vinculados a la UCI	Cargo que ocupa	Experiencia
Experto 1	Ingeniería en Ciencias Informáticas	1	Especialista Calidad de Software.	3 años
Experto 2	Ingeniería en Ciencias Informáticas	1	Especialista de la Dirección Calidad y Asesor del DDC-IGSW.	4 años
Experto 3	Ingeniería en Ciencias Informáticas	1	Asesor de Calidad.	4 años
Experto 4	Ingeniería Informática	3	Jefa de Departamento de IGSW y PP.	2 años
Experto 5	Ingeniería Informática	2	Especialista de la Dirección de Regulaciones y Normas.	2 años
Experto 6	Ingeniería en Ciencias Informáticas	1	Especialista de Calidad.	3 años
Experto 7	Ingeniería Informática	2	Especialista General de Calidad.	2 años

Anexo 11: Cuestionario presentado a los expertos.

Usted ha sido seleccionado por su conocimiento en calidad de software como experto para evaluar los resultados teóricos de esta investigación.

1- En su opinión, el procedimiento propuesto se debe aplicar a los proyectos productivos de la facultad 3.
SI__ NO__ ¿Por qué?

_____

_____

_____

2- ¿Considera usted que si se aplicará el procedimiento propuesto mejoraría la calidad del software que son desarrollados en los proyectos de la facultad 3?
Si __No__ ¿Por qué?

_____

_____

_____

3- Mencione qué actividades deben llevarse a cabo para asegurar la calidad del software.
1-_____
2- _____
3- _____
4- _____
5- _____
6- _____
7- _____
8- _____
9- _____
10-_____

4- Haga un comentario sobre los beneficios o dificultades que se pueden presentar al aplicarse el procedimiento propuesto en los proyectos productivos de la facultad 3 (el comentario debe aportar elementos que contribuyan a la mejora del procedimiento).

Para finalizar, queremos expresarle que sus criterios y opiniones se manejarán de forma anónima, además le agradecemos por anticipado su valiosa colaboración y estamos seguros que sus sugerencias contribuirán a perfeccionar el procedimiento propuesto.

Muchas gracias por su cooperación y le pedimos disculpas por las molestias ocasionadas

GLOSARIO DE TERMINOS Y SIGLAS

Aseguramiento de la calidad del software (SQA, Software Quality Assurance): es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad.

Actividad: conjunto de tareas interrelacionadas que garantizan el resultado esperado.

Complejidad ciclomática: es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa, da el límite superior del número de pruebas que se deben realizar.

CMM: Modelo de Madurez de Capacidad**.** Dirige su enfoque a la mejora de procesos en una organización, estudia los procesos de desarrollo y produce una evaluación de la madurez (indicador para medir la capacidad para construir un software de calidad) de la organización según una escala de cinco niveles (inicial, repetible, definido, dirigido y optimizado).

CMMI: Modelo de Madurez de Capacidad Integrado. Es un conjunto de modelos elaborados por el Instituto de Ingenieros del Software (Software Engineering Institute, SEI) que permiten obtener un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización, y describen las tareas que se tienen que llevar a cabo para mejorar esos procesos

Calidad del software (CS)**:** es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente

Eficiencia: relación entre los resultados logrados y los recursos utilizados.

Flujo de trabajo: secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Ingeniería de Software: es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos, el proceso y las herramientas.

ISO: conjunto de estándares internacionales para sistemas de calidad. Diseñado para la gestión y aseguramiento de la calidad, especifica los requisitos básicos para el desarrollo producción, instalación y servicio a nivel de sistema y a nivel de producto.

Lineamientos: conjunto de acciones específicas que determinan la forma, lugar y modo para llevar a cabo una política en materia de obra y servicios relacionados con la misma.

MoProSoft: es un modelo de procesos para la industria de software mexicano.

Método Delphi: Es una técnica prospectiva para obtener información esencialmente cualitativa, pero relativamente precisa, acerca del futuro. Consiste básicamente en solicitar de forma sistemática las opiniones de un grupo de expertos, pero prescindiendo de la discusión abierta, lo que permite evitar los inconvenientes de ésta.

Norma de calidad: es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad

Procedimiento: serie de pasos, claramente definidos, que permiten trabajar correctamente, disminuyendo la probabilidad de accidentes. Modo de ejecutar determinadas operaciones. Suelen realizarse de la misma manera.

Proceso: es un conjunto de prácticas que se ejecutan con un propósito determinado, las cuales transforman entradas en salidas, que son de valor para el cliente. El proceso puede incluir herramientas, métodos, materiales y/o personas

RUP: Rational Unified Process. Es un proceso de ingeniería de software.

Revisión técnica formal (RTF): es una actividad para asegurar la calidad del software.

SEI: Instituto de Ingeniería de Software (Software Engineering Institute) de la Universidad Carnegie Mellon (EEUU). Autor de los modelos CMMI y del método SCAMPI.

SPICE: Software Process Improvement and Capability Determination. Es un emergente estándar internacional de evaluación y determinación de la capacidad y mejora continua de procesos de ingeniería del software.

Software (SW): es un término genérico que designa al conjunto de programas de distinto tipo (sistema operativo y aplicaciones diversas que hacen posible operar con la computadora).

Tarea: conjunto de acciones simples interrelacionadas para ejecutar una actividad.

(UCI) Universidad de las Ciencias Informáticas.

(IEC) Comisión Electrotécnica Internacional.

(TI) Las tecnologías de información.

(ICSW) Industria Cubana del Software.

(ISW) Ingeniería de Software.

(GCS) Gestión de configuración del software

(LMC) Lineamientos mínimos de calidad.