Universidad de las Ciencias Informáticas
Facultad #3

Título: Diseño del proceso de Aseguramiento de la Calidad del Software para SIGIA

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

Autora: Yudaika Ray Jiménez

Tutora: Ing. Yenisleidy Rendon Vigil

Ciudad de la Habana

Junio, 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad #3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los __18__ días del mes de ___junio_____ del año ___2007_____.

Yudaika Ray Jiménez	Yenisleidy Rendon Vigil
_____	_____
(Autora)	(Tutora)

AGRADECIMIENTOS

En primer lugar a la Revolución y a Fidel por permitir que yo fuera partícipe de este gran proyecto.

A mi Madre por estar a mi lado siempre y darme el apoyo necesario para seguir adelante aun en los momentos más difíciles

A mi hermana Yudisleika y mi sobrinita Charlene por ser la luz de mis ojos.

A mis amigas Leidy B., Geidy, Yunet (la maga), a mis dos Yane y Lisandra (LHM) por soportarme todo este tiempo y por ser las mejores amigas del mundo.

Y un agradecimiento especial a kelsey Martínez Mesa la persona más linda del mundo y a la cual amo con todas las fuerzas de mi corazón, por alumbrar mi camino cuando todo estaba oscuro, y por darme siempre el impulso suficiente para seguir adelante.

DEDICATORIA

Muchos dedican a sus padres, incluso a su memoria… yo dedico a mis esfuerzos.

A Armando Ray Pérez (papi) se que estarías orgulloso de mi, donde quiera que estés… gracias, a tu lado fueron los mejores años de mi vida.

RESUMEN

Con el avance gradual de las tecnologías de la informática y las comunicaciones existe un aumento sustancial de la utilización de las mismas han provocado que se perfeccionen cada día más las técnicas y herramientas de los procesos de desarrollo del software.

Muchos de los proceso de desarrollo del software no son culminados en tiempo, además de no cumplir con las exigencias expuestas por los más beneficiados (clientes). Una de las causas que provocan este suceso es la falta de un proceso de aseguramiento de la calidad del software bien definido, planificado y ejecutado por parte de los ingenieros de software.

El proceso de aseguramiento de la calidad del software es el destinado a cumplir con las especificaciones tanto del cliente como de calidad, el cumplimiento con los estándares preestablecidos proporcionando que el producto sea entregado en el tiempo establecido.
Este proceso es de suma importancia dentro del desarrollo del software ya que encontrar y corregir defectos es muy costoso, es muy conveniente que los ingenieros encuentren y corrijan inmediatamente los defectos introducidos.
La presente investigación pretende diseñar un proceso de Aseguramiento de la calidad del Software en función de orientar un proceso de desarrollo de software que viabilice la ejecución de los procesos de la GIA en el país.

PALABRAS CLAVE

Aseguramiento de la Calidad del Software, Sistema de Gestión de Inventarios y Almacenes

TABLA DE CONTENIDOS

INTRODUCCIÓN

Hoy en día la demanda que existe de producción de software asciende a un ritmo acelerado, desarrollar productos que cumplan con los requisitos y que respondan a la planificación y el presupuesto establecido son temas que provocan un proceso de desarrollo organizado.

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En los pasados 50 años el software ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por sí misma [Pressman, 2005]. Sin embargo nada de esto sería posible sin un factor muy importante dentro del proceso de desarrollo, llamado calidad del software.

La Calidad del Software viene desarrollándose desde el siglo XIII cuando la producción era artesanal y estaba definida por la comunicación directa que existía entre el productor y el consumidor, en aquel entonces el producto se ajustaba exactamente a las necesidades del cliente, en este caso el artesano controlaba la calidad del producto.

A principios del siglo pasado calidad se consideraba sinónimo de inspecciones donde todos los productos terminados se inspeccionan y todos los defectos encontrados se corrigen.

En la segunda mitad de la década del veinte comenzó el desarrollo de los métodos estadísticos para el control de la calidad, sus métodos son herramientas y técnicas estadísticas. Este método utilizado para el control de la calidad únicamente tiene relación con el producto final.

Del control de la calidad se pasa al aseguramiento de la misma, su objetivo principal es incorporar la calidad desde los inicios del producto hasta su fase final de una forma planificada.

En los años 80, un grupo de empresas se unieron y dieron lugar a la gestión de la calidad total (TQM), esta por el contrario no se regía por normas, sino por modelos, ejemplo el modelo Malcom Baldrige, el modelo Deming, y modelo EFQM. Esta fase de la calidad tiene como objetivo dirigir la organización, con la asistencia del personal de servicio de la misma, hacia la mejora de los productos.

El software cada vez adquiere mayor relevancia en las actividades que se realizan y en los dispositivos que se utilizan para ello, estando así presente en muchos aspectos de la vida diaria: en el gobierno, bancos, educación, transporte, entretenimientos, medicina, etc. Las áreas de

aplicación del software han crecido de manera impresionante, tanto así que en Cuba se creó una Universidad donde los estudiantes están vinculados a proyectos productivos dedicados al software, con el objetivo de brindar software de calidad a los diferentes clientes.

Uno de esos proyectos productivos es el Sistema de Gestión de Inventarios y Almacenes el cual tiene como objetivo principal ser un patrón para la gestión de inventarios en nuestro país y todo lo relativo al control y manejo de la existencias de determinados bienes, en el cual se aplica métodos y estrategias que pueden hacer rentable y productivo la tendencia de estos bienes y a la vez sirve para evaluar los procedimientos de entradas y salidas de dicho producto. El objetivo principal del proyecto es obtener un producto de calidad para la industria del software, para esto se necesita la aplicación de procedimientos y estrategias que lleva consigo el aseguramiento de la 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 [Pressman, 2005].

La ingeniería del software hoy en día es relativamente reciente y muchos de sus conceptos importantes están aun inmaduros. El proyecto productivo Sistema de Gestión de Inventarios y Almacenes ha estado presentando dificultades debido a que no existe un previo conocimiento de los verdaderos objetivos para llevar a cabo un proceso de calidad y de los mecanismos que esta proporciona para su entendimiento y desarrollo. Dada esta situación se está en presencia de la **situación problema** de este trabajo de diploma: El proyecto productivo Sistema de Gestión de Inventarios y Almacenes (SIGIA) carece de los mecanismos necesarios para el aseguramiento de la calidad interna del software durante su desarrollo y por consiguiente de la calidad en la entrega del producto final. Con el objetivo de dar solución a la problemática anterior se plantea la siguiente **pregunta científica**: ¿Qué hacer para que el producto final (SIGIA) sea entregado con la calidad requerida?

Con el fin de darle solución a la interrogante anterior se define la **hipótesis.** Suponiendo que se asegure la calidad del software en el proyecto productivo **(SIGIA),** entonces el producto final alcanzará la calidad requerida dándole cumplimiento a los requisitos propuestos. Por esta razón

el presente Trabajo de Diploma tiene como **objetivo fundamental**, aplicar mecanismos para asegurar la calidad del software en el sistema Gestión de Inventarios y Almacenes de forma tal que este cumpla con los requisitos del sistema.

Tomando en cuenta el problema planteado y el objetivo fundamental se tiene como **objeto de estudio** el proceso de Aseguramiento de la calidad del software.

Y el **campo de acción** al cual está dedicado este Trabajo de Diploma es: La planificación y control de la calidad del software en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

Con el fin de darle cumplimiento al objetivo fundamental del presente trabajo se plantean los diferentes **objetivos específicos:**

1. Realizar un análisis a la bibliografía con el fin de apoyar el Trabajo de Diploma.
2. Realizar un plan de calidad que se ajuste a las normas y estándares expuestos tanto por calidad como por el propio proyecto Sistema de Gestión de Inventarios y Almacenes.
3. Efectuar diferentes revisiones en el proyecto Sistema de Gestión de Inventarios y Almacenes, con el propósito de lograr que este se ajuste a los requisitos expuestos por el Ministerios de Precios y Finanzas, además de asegurar la documentación completa de todas las iteraciones ejecutadas en el proyecto.
4. Coordinar el control y la gestión de cambios para que todo el proyecto fluya uniformemente.
5. Definir métricas del software.

Para la realización del actual Trabajo de Diploma se tuvo en cuenta diferentes métodos Científicos.

En el comienzo de la investigación hubo que hacer un estudio de los mecanismos y estándares de calidad y relación entre ellos para ello se utilizó el **método Analítico-Sintético.** Más adelante para llevar a cabo el proceso de calidad y darle cumplimiento y seguimiento al mismo se utilizó el método **Histórico-Lógico**.

En el momento de determinación de la hipótesis permitiendo inferir conclusiones a partir de los conocimientos previos, dándole solución a los objetivos específicos mediante métodos científicos bien fundamentados fue de gran importancia la utilización del método **Hipotético-deductivo**.

Igualmente fue de gran importancia el uso de métodos empíricos los cuales permiten extraer informaciones necesarias para evaluar y comparar resultados obtenidos, para de esta forma saber si se está dando seguimiento y cumplimiento a los objetivos planteados por lo cual se utilizó el **Experimento.**

Los **aportes prácticos esperados del trabajo:**

Se espera obtener un proceso de Aseguramiento de la Calidad del Software en el proyecto Sistema de Gestión de Inventarios y Almacenes que permita al producto final SIGIA satisfacer los requisitos propuestos por los clientes.

El **trabajo está estructurado en 2 capítulos:**

Capítulo 1: Fundamentación Teórica:

Definición del marco teórico de la investigación y estudio del arte de la calidad del software.

Capítulo 2: Diseño del proceso de Aseguramiento de la Calidad del software:

En este capítulo se realiza la descripción de la solución propuesta en el diseño teórico de la investigación.

La solución propuesta está dividida en 2 epígrafes principales:

- ✓ Confección del plan de Aseguramiento de la Calidad del Software
- ✓ Confección del plan de Revisiones y Auditorías

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

La situación actual y las perspectivas de la industria de software cubana están caracterizadas por el trabajo que se viene realizando en materia de capacitación y en la inserción en el mercado nacional como internacional.

A esto se le adiciona la puesta en marcha de un nuevo paradigma productivo: Fábrica de Software; cuyo objetivo es responder productivamente a demandas crecientes del mercado Europeo y de Latinoamérica; organizados productivamente según los estándares internacionales [Febles, 2002].

El presente capítulo tratará diferentes epígrafes que servirán como base de conocimientos para un mejor entendimiento del verdadero objetivo de este Trabajo de Diploma.

Inicialmente se brinda una serie de conceptos bien fundamentados a lo que calidad respecta que expresan diferentes autores indagadores del tema. Estos conceptos muestran una visión profunda de la calidad del software, así como también sus verdaderos objetivos y metas. También se tratarán los conceptos de calidad del software, garantía y aseguramiento de la calidad del software, control de calidad del software, costos de la calidad del software así como también las tendencias de la calidad del software.

Se expone la importancia que tiene el aseguramiento de la calidad del software para cualquier institución desarrolladora de programas informáticos. Se trata un tema fundamental en el desarrollo de este trabajo de diploma y es lo referentes a las revisiones que se van a llevar a cabo para el control y aseguramiento de la calidad en el proyecto.

Más adelante se realiza un estudio profundo de los modelos y estándares existentes para la administración de la calidad en un proyecto de software. En este epígrafe se da a conocer los diferentes modelos y estándares que se utilizan de apoyo al proceso de calidad, así como también acotaciones expuestas por diferentes autores en referencia al tema expuesto.

Teniendo en cuenta que calidad sin mediciones estaría incompleta; se brinda el adecuado tratamiento al tema dedicado a las métricas de calidad del software, y los diferentes factores que

dan lugar a la obtención de un software que cumpla con los objetivos específicos expuestos por los clientes.

Por último se realiza un estudio de la calidad del software en Cuba, su evolución y desarrollo, presentando de esta manera el estado actual en que se encuentra el tema en la industria cubana del software. Posteriormente se realiza el mismo estudio pero esta vez en la Universidad de las Ciencias Informáticas (UCI).

1.1 Conceptos fundamentales

Proceso

Un proceso transforma elementos de entrada en elementos de salida utilizando para ello mecanismos (recursos), regulados mediante controles. Un proceso no es más que la sucesión de pasos y decisiones que se siguen para realizar una determinada actividad o tarea.

También se define proceso como cualquier actividad o grupo de actividades relacionadas, mediante las cuales se agrega valor a unas entradas (materiales o inmateriales) y, de esta forma se suministran productos, servicios e información a un cliente externo o interno a la empresa [Cockburn, 2000].

Se puede definir proceso como la secuencia ordenada y lógica de actividades repetitivas que se realizan en la organización por una persona, grupo o departamento, con la capacidad de transformar las entradas (inputs), en salidas o resultados programados (output) para un destinatario (dentro o fuera de la empresa que lo ha solicitado y que son los clientes de cada proceso) con un valor agregado. Los procesos, generalmente, cruzan repetidamente las fronteras funcionales, fuerzan a la cooperación y crean una cultura de empresas distintas (más abierta, menos jerárquica, más orientada a obtener resultados que a mantener privilegios) [Nogueira, 2004].

Dada estas definiciones de proceso se llega a la conclusión de que un proceso es un conjunto de actividades relacionadas entre si, las cuales convierten los elementos de entrada en salidas o resultados.

Metodología de desarrollo

Una metodología es el estudio de los métodos; de manera que la metodología de las matemáticas es el estudio de los métodos que se utilizan para encontrar y comprobar verdades matemáticas. También define al método como un medio sistemático para trabajar mediante el cual se obtiene un resultado deseado [Wieringa, 1996].

Tomando como referencia esta definición, se concluye que una metodología de desarrollo de software, no es más que el estudio de los métodos más apropiados que se emplean para

desarrollar software de manera eficiente; o como precisan otros autores (Jacobson, et al., 2001), define Quien debe hacer Que, Cuando, y Como debe hacerlo.

Sistema

Conjunto de elementos mutuamente relacionados o que interactúan, y los elementos pueden considerarse actividades.

Inventarios

El inventario es una reserva de materiales que incluye materia prima, trabajo o producto en proceso y producto terminado.

Sistemas de inventarios

Es un conjunto de políticas y controles que supervisa los niveles de inventario y determina cuáles son los niveles que deben mantenerse, cuándo hay que reabastecer el inventario y de qué tamaño deben ser los pedidos.

En sentido amplio, el inventario incluye insumos de tipo humano, financiero, energético, de equipo y materias primas, y salida como piezas, componentes y bienes terminados, así como las etapas intermedias del proceso, como los productos en curso o semiterminado. En sentido estricto, el inventario de manufactura está formado por los entes materiales que forman parte de los productos de la empresa, y se divide, tradicionalmente en materias primas, productos terminados, piezas o componentes, productos en proceso y suministros. En las actividades de servicio, el inventario se refiere generalmente a los bienes tangibles que pueden venderse, y a los suministros necesarios para administrar el servicio.

Gestión de inventarios

Se entiende por Gestión de Inventarios, todo lo relativo al control y manejo de las existencias de determinados bienes, en la cual se aplican métodos y estrategias que pueden hacer rentable y productivo la tenencia de estos bienes y a la vez sirve para evaluar los procedimientos de entradas y salidas de dichos productos [Bauer, 1969].

La gestión de inventarios se define en la literatura con diferentes acepciones, tales como: El conjunto de acciones destinadas a minimizar los gastos e incrementar los beneficios originados en el almacenamiento de existencias.

Calidad

El concepto de calidad se ha desarrollado mucho a través de la historia, muchas personas reconocidas en todo el mundo por sus grandes aportes a este aspecto de la ingeniería del software dan constancia de ello.

Joseph M. Juran se le reconoce por sus grandes aportes en el proceso de calidad, logró desarrollar la técnica de los Costos de Calidad elaborando un manual de calidad en donde existe un fuerte contenido administrativo dedicado a la planeación, organización y responsabilidad.

Calidad es el conjunto de características de un producto que satisfacen las necesidades de los clientes y en consecuencia hacen satisfactorio el producto [Durán, 2000].

Calidad no es más que conformidad con los requerimientos indicando que el 100% de la conformancia es igual a cero defectos [Crosby, 1979].

Tiempo después se amplia 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 [Crosby, 1994].

Totalidad de las características de una entidad que le confieren la aptitud para satisfacer necesidades establecidas o implícitas [ISO 8402, 1994]

Grado en el que un conjunto de características inherentes cumple con los requisitos. Y añade dos notas al respecto:

Nota 1. El término calidad puede utilizarse acompañado de adjetivos tales como pobre, buena o excelente.

Nota 2. “Inherente” en contraposición a “asignado” significa que existe en algo, especialmente como una característica permanente [ISO 9000, 2000].

Planificación de 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 la calidad. El establecimiento de los planes de la calidad puede ser parte de la planificación de la calidad.

Mejoramiento de la calidad

El mejoramiento de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad.

Costo de calidad

El costo de calidad incluye todos los costes acarreados en la búsqueda de calidad o en las actividades relacionadas en la obtención de la calidad [Pressman, 2005].

1.2 Calidad del software

En la gran mayoría del software nacional e internacional se habla de llevar a cabo un proceso de aseguramiento de calidad del software que permita el mejoramiento del producto en su gran totalidad, darle al cliente lo que realmente quiere y que estos se vayan con la satisfacción en sus manos.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación. La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas deriva dos 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 [Ferré, 2005].

Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente [Pressman, 2005].
Teniendo en cuenta las diferentes definiciones expuestas anteriormente, las cuales dan una visión profunda y clara de la calidad del software, el presente trabajo se adscribe a la definición que da Pressman, puesto que se puede vincular fácilmente al proyecto productivo Sistema de Gestión de Inventarios y Almacenes que es el tema principal de este Trabajo.

1.2.1 Control de la calidad del software

El control de cambios puede equiparse al control de calidad. Pero, ¿Cómo se logra el control de calidad? 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. El control de la calidad 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 [Pressman, 2005].
El objetivo principal de control de calidad consiste en detectar y corregir los errores que surgen durante el proceso de desarrollo de un producto de software, así como garantizar que el producto final responda a las expectativas del cliente.
Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición. Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Por ejemplo, John Wiley define métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. 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 [Ferré, 2005].

1.2.2 Cómo obtener un software de calidad

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico. El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software. El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software. El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado. La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación [Ferré, 2005].

1.2.3 Importancia de la calidad del software

Los ingenieros de software desarrollan partes de grandes productos o sistemas, no importa como sean de pequeños con respecto al tamaño total del producto o la escasa importancia que tengan los mismos, cualquier defecto en ellos podría potencialmente dañar todo el sistema. La calidad de un sistema de software no solamente está gobernada por la calidad de sus partes, sino que cualquier error trivial en los programas de soporte puede tener efectos devastadores.

Los sistemas informáticos modernos ejecutan millones de instrucciones por segundo, así que un defecto poco probable que pueda suceder de una vez en un billón puede ocurrir varias veces en un día. Con el software, las condiciones inusuales se presentan varias veces, condiciones que parecen imposibles, ocurren en poco tiempo. Los defectos en los elementos más pequeños de un gran sistema pueden causar serios problemas de forma impredecible y ocasional. Si se comete un error trivial que deja un defecto en el producto, el resultado puede causar un gran inconveniente o incluso un daño físico al usuario.

La calidad del software consiste en satisfacer las necesidades de los clientes haciendo el trabajo de los mismos de una forma fiable y consistente. Encontrar y corregir defectos es muy costoso, es por esto que es muy conveniente que los ingenieros encuentren y corrijan inmediatamente los defectos introducidos.

Mantener la calidad del software es algo costosa pero comparado con lo que puede costarte aplica el proceso de calidad luego de haber salido el producto final es incomparable. La Calidad del software es un proceso de suma importancia dentro del desarrollo de software, permitiendo que el producto final cumpla con las exigencias de los clientes, además ahorra tiempo y dinero.

Se cree que no es posible establecer la calidad de los sistemas que se están construyendo hasta se tenga el producto terminado. Nada más errado. Si se definen entregables durante todo el proceso de desarrollo del software como diagramas de clases, diseño de casos de pruebas es posible establecer si el producto que se está construyendo es de alta calidad aún en sus etapas iniciales. Las revisiones técnicas formales es una técnica muy efectiva; si se ha realizado una cuidadosa planificación del trabajo y se han asignado responsabilidades y establecido resultados de cada actividad es posible realizar revisiones por personas diferentes al desarrollador del producto.

Esencialmente, la Calidad del Software consiste en la revisión de los productos y su documentación relacionada, para verificar su cobertura, corrección, confiabilidad y facilidad de mantenimiento. Y, por supuesto, incluye la garantía de que un sistema cumpla las especificaciones y los requerimientos para el uso y desempeño deseados. En la figura 1.1, se muestra un gráfico presentado a partir de un estudio realizado en empresas productoras de software [Reynolds, 1995], que muestra una comparación entre sistemas que han sido construidos con alta calidad y sistemas construidos con escasa calidad, lo que evidencia que en los sistemas desarrollados con alta calidad se disminuyen considerablemente los costos en el mantenimiento, pues estos sistemas han sido sometidos a revisiones que detentan y eliminan defectos en etapas tempranas de su construcción.

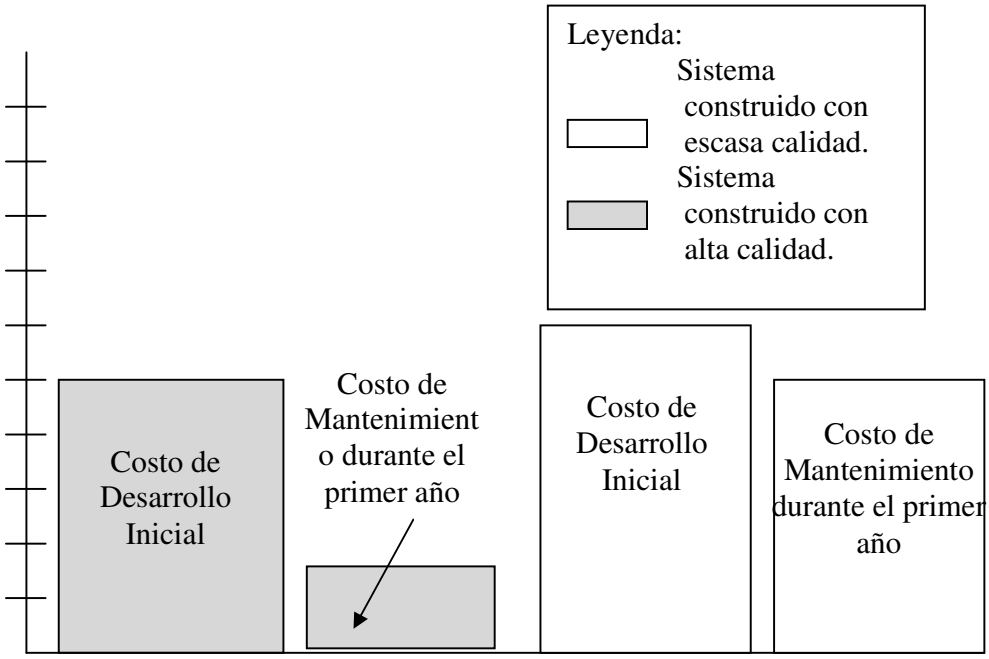


Fig. 1.1 Costo de Construcción y Mantenimiento en un sistema de acuerdo a la calidad de éste

1.3 Revisiones del software

Las revisiones del software son un filtro para el proceso de ingeniería del software. Las revisiones se aplican en varios momentos del desarrollo del software y sirven para detectar errores y defectos que puedan así ser eliminados. Las revisiones del software sirven para purificar las actividades de ingeniería del software que suceden como resultado del análisis, el diseño y la codificación. Existen muchos tipos de revisiones que se pueden llevar adelante como parte de la ingeniería del software, en el caso del proyecto productivo Sistema de Gestión de Inventarios y Almacenes se utilizará la revisión técnica formal (RTF) a veces denominada inspección.

1.4 Garantía de calidad del software (Software Quality Assurance/Gestión de Calidad del Software)

La gestión de cualquier proceso no es más que las actividades coordinadas para dirigir y controlar una organización

La SQA es una actividad de protección que se aplica a cada paso del proceso del software. Comprende procedimientos para la aplicación efectiva de métodos y herramientas,

revisiones técnicas formales, técnicas y estrategias de pruebas, procedimientos de garantía de ajuste a los estándares y mecanismos de medida e información. El aseguramiento de la calidad del software o como otros libros lo llaman, garantía de la calidad del software (SQA/GCS) no es más que un patrón de acciones planificado y sistemático que se requiere para asegurar la calidad del software [Pressman, 2005].

La SQA es la guía de los preceptos, de gestión y de las disciplinas de diseño de la SQA, para el espacio tecnológico y la aplicación de la ingeniería del software [IEEE, 2004]

La garantía de la calidad del software es un conjunto de actividades de planificación, estimación y supervisión de las actividades de desarrollo, que se realizan de forma independiente al equipo de desarrollo, de tal forma que los productos software resultantes cumplen los requisitos establecidos [Reuse, 2006].

La garantía de calidad del software es un conjunto de procedimientos, técnicas y herramientas, aplicados por profesionales, durante el ciclo de desarrollo de un producto, para asegurar que le producto satisface o excede los estándares o niveles de calidad preestablecidos [IEEE, 1990].

El objetivo de SQA es proporcionar personas y gestión con el objetivo de que los procesos y los elementos de trabajo cumplan los procesos. Esto se consigue mediante diferentes actividades. (1) Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos. (2) Identificar y documentar los elementos no conformes. (3) Proporcionar información a las personas que están usando los procesos y a los gestores, de los resultados de las actividades del aseguramiento de la calidad. (4) Asegurar de que los elementos no conformes son arreglados [Pressman, 2005].

1.5 Tendencias de la calidad del software

Hoy en día los responsables expertos de compañías de todo el mundo industrializado reconocen que la alta calidad del producto se traduce en ahorro de coste y en una mejora general. Sin embargo esto no era simplemente el caso. La tendencia de la calidad comenzó en los años cuarenta con el influyente trabajo de W. Edwards Deming, y se hizo la primera

verificación en Japón [Pressman, 2005]. Mediante las ideas de Deming como piedra angular, los japonenses han desarrollado un enfoque sistemático para la eliminación de las causas raíz de defectos en productos. A lo largo de los años setenta y ochenta, su trabajo emigró al mundo occidental y a veces se llama gestión total de la calidad, aunque la terminología difiere según los diferentes países y autores, normalmente se encuentra una progresión básica de cuatro pasos que constituye el fundamento de cualquier programa de GTC [Pressman, 2005]. Estos pasos son:

- *Kaisen* (sistema de mejora continua): Objetivo de este paso es desarrollar un proceso (en este caso, proceso del software) que sea visible, repetible y mensurable.

- *Atarimae hinshitsu*: Examina lo intangible que afecta el proceso y trabaja para optimizar su impacto en el proceso. Por ejemplo, el proceso de software se puede ver afectado por la alta rotación del personal, este paso lleva a la gestión a sugerir cambios en la forma en que ocurren la reorganización.

- *Kansei*: Mientras que los dos primeros pasos se centran en el proceso este está centrado en el usuario del producto. En esencia examinando la forma en que el usuario aplica el producto, este paso conduce a la mejora en el producto mismo, y potencialmente al proceso que lo creó.

- *Kuteki hinshitsu*: Este paso es un intento de detectar productos nuevos y beneficiosos, o aplicaciones que sean una extensión de un sistema ya existente basado en computadoras.

Los sistemas de software son cada vez más importantes en la sociedad moderna y están creciendo rápidamente en tamaño y complejidad. Forzado por la competencia y la tendencia a ciclos de vidas más cortos, la calidad de los productos debe ser cada vez mayor. Aparte de las técnicas modernas de especificación, diseño e implementación de software la introducción de un proceso de testeo es vital para asegurar un nivel de calidad apropiado [Pressman, 2005].

1.6 Modelos y estándares existentes para la administración de la calidad en un proyecto de software

En la actualidad, se enfrenta una situación de dos caras. Por una parte las organizaciones quieren ser capaces de desarrollar y entregar software confiable, a tiempo y apegado al presupuesto acordado con el cliente. La segunda cara de la moneda nos muestra la perspectiva del cliente, el cuál quiere saber con certeza que todo lo anterior se cumplirá. Por esto las organizaciones deben buscar una norma, estándar o modelo que pueda ayudarlas a conseguir su meta de calidad (competitividad). Sin embargo, la competitividad no es la única razón por la cuál se busque la calidad en el software. Se le debe dar importancia a cada programa que se desarrolla. Se debe tomar conciencia y responsabilidad de las consecuencias que un defecto en nuestro producto podría ocasionar. Algunos defectos de software han ocasionado serios daños y hasta perjudicado físicamente a personas. Gente ha muerto debido a software defectuoso [Humphrey, 2001].

La terminología para las características de la calidad del software, difieren de un modelo de calidad del software a otro. Cada modelo tiene un número diferente de niveles jerárquicos, así como también un número totalmente diferente de características.

Los modelos y estándares de la calidad del software forman parte de la ingeniería del software, es muy importante la implantación de los mismos debido a que la competencia es cada día más fuerte. Hoy en día los responsables expertos de compañías de todo el mundo industrializado reconocen que la alta calidad del producto se traduce en ahorre de costo y en una mejora general [Pressman, 2005].

Uno de los modelos de calidad del software más importantes es CMMI (Capability Maturity Model Integration), proporcionando una guía para el desarrollo de los procesos de software.

Otro de los estándares que ha tenido un impacto en el desarrollo y evolución del software es la familia de ISO 9001 (Organization for International Standarization) junto con sus pautas (ISO 9003-4). Este estándar ha sido adoptado por más de 130 países para su uso, se esta convirtiendo en el medio principal con el que los clientes pueden juzgar la competencia de

un desarrollador de software. El problema de ISO 9001 es que no está expresado en términos de software, sino de manera general [Pressman, 2005].

Existía un debate inicialmente acerca de cual modelo debería usarse por los ingenieros de software para asegurar la calidad, si ISO 9001 o CMMI, y como resultado se decidió que ambos son complementarios, y que teniendo la certificación de ISO 9001 puede ayudar grandemente a lograr el nivel más alto de madurez del modelo CMMI.

Como CMMI e ISO 9001, SPICE también es un modelo de calidad de suma importancia, se caracteriza por la evaluación a los procesos de software para la mejora continua, valoración de la capacidad de cada software y como base para el comercio internacional de cada software. Su trayectoria esta enmarcada a ejecutar, planificar, gestionar, controlar, y mejorar los procesos de adquisición, suministro, desarrollo, operación y soporte.

Para llevar a cabo el aseguramiento de la calidad del software, debe tenerse en cuenta un modelo de calidad ampliamente aceptado y orientado al desarrollo de productos de software, se estudiaron tres modelos ISO 9001, CMMI, y SPICE, se decidió optar por CMMI, por las siguientes razones:

- El modelo CMMI presenta cinco niveles, los cuales se utilizan para conocer la madurez de los procesos que se realizan para producir software.
  - ♦ *Nivel 1 o Inicial*: Este es el nivel donde están todas las empresas que no tienen procesos, los presupuestos se disparan, no es posible entregar el proyecto en fechas, el personal involucrado se tiene que quedar durante noches y fines de semana para terminar un proyecto. No hay control sobre el estado del proyecto, el desarrollo del proyecto es completamente opaco, no se sabe lo que pasa en él.
  - ♦ *Nivel 2 o Repetible*: Quiere decir que el éxito de los resultados obtenidos se pueden repetir, la principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

    Los procesos que hay que implantar para el desarrollo de este nivel son:

    Gestión de requisitos

Planificación del proyecto.
Seguimiento y control de proyecto
Gestión de proveedores
Aseguramiento de la calidad
Gestión de la configuración.

- *Nivel 3 o Definido*: Alcanzar este nivel significa que la forma de desarrollar proyectos está definida, esto quiere decir que está establecida, documentada y que existen métricas para la consecución de objetivos concretos.
  Los procesos que hay que implantar para el desarrollo de este nivel son:
  Desarrollo de requisitos
  Solución técnica
  Integración del producto
  Verificación
  Validación
  Desarrollo y mejora de los procesos de la organización
  Planificación de la formación
  Gestión de riesgos
  Análisis y resolución de toma de decisiones.
- *Nivel 4 o Cuantitativamente gestionado*: Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización.
  Los procesos que hay que implantar para el desarrollo de este nivel son:
  Gestión cuantitativa de proyectos
  Mejora de los procesos de la organización.
- *Nivel 5 u Optimizado*: Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras e incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puesta en práctica.
  Los procesos que hay que implantar para el desarrollo de este nivel son:

Innovación organizacional.

Análisis y resolución de las causas.

- Es un proceso largo y costoso que puede costar varios años de esfuerzo, aún así el beneficio obtenido para la empresa es mucho mayor que lo invertido.
- Una empresa que obtiene el nivel de madurez CMMI cuenta con una evidencia objetiva respecto a la efectividad de sus procesos de desarrollo, operación y mantenimiento del software, que es reconocida internacionalmente como una ventaja competitiva distintiva.
- CMMI es una recopilación de experiencias y soluciones a casos concretos compaginada como una colección estructurada de mejores prácticas.
- Durante la evaluación se verifica de que manera la organización ha implantado las mejores prácticas contenidas en el modelo, y se otorga un nivel de madurez que avala los procesos utilizados.
- CMMI destaca, en las empresas que desarrollan software, la voluntad de mejorar las cosas y reducir su dependencia de los héroes.

1.7 Factores que determinan la calidad del software

Un software de alta calidad es una de las metas más importantes. 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) [Pressman, 2005].

1.7.1 Clasificación de factores de calidad

Factores de McCall

McCall y sus colegas propusieron una útil clasificación de factores que afectan la calidad del software. Estos factores de calidad se concentran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad de cambios y su adaptabilidad a nuevos entornos. Cada una de estas características tiene un conjunto de

factores que las describen. Los factores de calidad descritos por McCall representan solo una de las muchas listas de comprobación sugeridas para la calidad del software [Pressman, 2005].

Características operativas:

- Corrección: Hasta donde satisface un programa su especificación y logra los objetivos expuestos por el cliente.
- Fiabilidad: Hasta donde se puede esperar que un programa lleve a cabo su función con la exactitud requerida.
- Eficiencia: La cantidad de recursos informáticos y de códigos necesarios para que un programa realice su función.
- Integridad: Hasta donde se puede controlar el acceso al software o a los datos por personas no autorizadas.
- Usabilidad (Facilidad de manejo): El esfuerzo necesario para aprender a operar con el sistema, preparar los datos de entrada e interpretar las salidas (resultados) de un programa.
- Facilidad de mantenimiento: El esfuerzo necesario para localizar y arreglar un error en un programa.
- Flexibilidad: El esfuerzo necesario para modificar un programa que ya está en funcionamiento.
- Facilidad de pruebas: El esfuerzo necesario para probar un programa y asegurar de que realice correctamente su función.
- Adaptabilidad a nuevos entornos:
- Portabilidad: El esfuerzo necesario para transferir el programa de un entorno a otro entorno diferente.
- Reusabilidad (Capacidad de reutilización): Hasta donde se puede volver a emplear un programa.
- Interoperabilidad: El esfuerzo necesario para acoplar un sistema con otro.

Factores FURPS

Existe un conjunto de factores a los cuales se les ha dado el acrónimo de FURPS: funcionalidad, facilidad de uso, fiabilidad, rendimiento y capacidad de soporte. Estos factores pueden usarse para establecer métricas de calidad para todas las actividades del proceso del software [Pressman, 2005].

- Funcionalidad: Se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.
- Facilidad de uso: Se valora utilizando factores humanos, la estética, la consistencia y la documentación general.
- Fiabilidad: Se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de las salidas, el tiempo de medio de fallos, la capacidad de recuperación de un fallo y la capacidad de predicción de un programa.
- Rendimiento: Se mide por la velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia.
- Capacidad de soporte: Combina la capacidad de ampliar el programa, adaptabilidad y servicios, así como la capacidad de hacer pruebas, compatibilidad, capacidad de configuración, la facilidad de instalación de un sistema y la facilidad con que se pueden localizar los programas.

Factores de calidad ISO 9126

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software [Pressman, 2005].

Funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento, portabilidad.

1.8 Métricas de calidad del software

Un elemento clave de cualquier proceso de ingeniería es la medición. Se emplean medidas para entender mejor los atributos de los modelos que se crean. Pero, fundamentalmente, se emplean las medidas para valorar la calidad de los productos de ingeniería o de los sistemas que se construyen.

La medición es el proceso por el que se asignan números o símbolos a los atributos de las entidades en el mundo real, de tal manera que las definan de acuerdo con unas reglas claramente definidas… En las ciencias físicas, medicina y, más recientemente, en las ciencias sociales, se es capaz de medir ya atributos que previamente se pensaba que no eran medibles… Por supuesto, tales mediciones no están tan refinadas como las de las ciencias físicas…, pero existen (y se toman importantes decisiones basadas en ellas). Se siente en la obligación de intentar << medir lo no medible>> para mejorar la comprensión de entidades particulares es tan poderosa en la ingeniería del software como en cualquier disciplina [Pressman, 2005].

Durante las pasadas tres décadas, muchos investigadores han intentado desarrollar una sola métrica que proporcione una medida completa de la complejidad del software. Aunque se han propuesto docenas de medidas de complejidad, cada una tiene un punto de vista diferente de lo que es la complejidad y qué atributos de un sistema llevan a la complejidad [Pressman, 2005].

Métricas del modelo de análisis

El trabajo técnico en la ingeniería del software empieza con la creación del modelo de análisis. En esta fase se obtienen los requisitos y se establece el fundamento para el diseño. Por tanto son deseables las métricas técnicas que proporcionan una visión interna a la calidad del modelo de análisis [Pressman, 2005].

La métrica de punto de función (PF) se puede usar como medio para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis [Pressman, 2005].

Al igual que la métrica punto de función, la métrica bang puede emplearse para desarrollar una indicación del tamaño del software a implementar como consecuencia del modelo de análisis. Esta métrica es una indicación independiente de la implementación del tamaño del sistema. Para calcular la métrica bang, el desarrollador de software debe evaluar primero un conjunto de primitiva (elementos del modelo de análisis que no se subdividen más en el nivel de análisis) [Pressman, 2005].

Davis y sus colegas proponen una lista de características que pueden emplearse para valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: especificidad, compleción, consistencia interna y externa, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización [Pressman, 2005].

Métricas del modelo de diseño

Las métricas de diseño para el software, como otras métricas del software, no son perfectas. Continúa el debate sobre la eficacia y como deberían aplicarse. Muchos expertos argumentan que se necesita más experimentación hasta que se puedan emplear las métricas de diseño. Y sin embargo, el diseño sin medición es una alternativa inaceptable [Pressman, 2005].

Las métricas del diseño de alto nivel se concentran en las características de la arquitectura del programa con especial énfasis en la estructura arquitectónica y en la eficiencia de los módulos. Estas métricas son de caja negra en el sentido que no requieren ningún conocimiento del trabajo interno de un módulo en particular del sistema [Pressman, 2005].

Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software e incluyen medidas de las "3C" (la cohesión, acoplamiento, y complejidad de módulo). Estas tres medidas pueden ayudar al desarrollador de software a juzgar la calidad de un diseño a nivel de componentes. Estas métricas son de caja blanca en el sentido de que requieren el conocimiento del trabajo interno del módulo en cuestión. Estas métricas se pueden aplicar una vez que se ha desarrollado un diseño procedimental. También se pueden retrasar hasta tener disponible el código fuente [Pressman, 2005].

Sears sugiere la conveniencia de la representación (CR) como una valiosa métrica de diseño para interfaces hombre-maquina. Una IGU (Interfaz gráfica de usuario) típica usa entidades de representación, iconos gráficos, texto, menús, ventanas y otras para ayudar al usuario a completar tareas. Para realizar una tarea dada usando IGU, el usuario debe moverse de una entidad de representación a otra [Pressman, 2005].

Métricas del código fuente

La teoría de Halstead de la ciencia del software es probablemente la más conocida y más minuciosamente estudiada... medidas compuestas de la complejidad (software). La ciencia del software asigna leyes cuantitativas al desarrollo del software de computadora, usando un conjunto de medidas primitivas que pueden obtenerse una vez que se ha generado o estimado el código después de completar el diseño [Pressman, 2005].

Métricas para pruebas

Aunque se ha escrito mucho sobre las métricas del software para pruebas, la mayoría de las métricas propuestas se concentran en el proceso de prueba, no en las características técnicas de las pruebas mismas. En general, los responsables de las pruebas deben fiarse de las métricas de análisis, diseño y de código para que les guíen en el diseño y ejecución de los casos de pruebas [Pressman, 2005].

Métricas del mantenimiento

El estándar IEEE 982.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software (basado en los cambios que ocurren con cada versión del producto) [Pressman, 2005].

Desarrollo de la métrica y de la OPM (Objetivo, pregunta, métrica)

Uno de los problemas al que se han enfrentado los trabajadores de las métricas durante las dos últimas décadas es la de desarrollar métricas que fueran útiles para el diseñador de software. Ha sido toda una historia de utilización de la m♪0trica dentro de los entornos industriales basada en el simple criterio de lo que era facilitar la medida, más que emplear cualquier criterio relacionado con la utilidad. El panorama de la última mitad de los años 80 y la primera mitad de la década de de los 90, constató el hecho de que mientras había sido desarrollado mucho trabajo en la validación de la métrica y en el esclarecimiento de los principios teóricos detrás de ella, muy poco había sido hecho para dotar al diseñador de software con herramientas para la selección o construcción de métricas.

La importancia de OPM proviene no solamente del hecho de que es uno de los primeros intentos de desarrollar un conjunto de medidas adecuado que pueda ser aplicado al

software, sino también al hecho de que está relacionado con el paradigma de mejora de procesos [Pressman, 2005].

1.8.1 Características fundamentales de las métricas del software

Se han propuesto cientos de métricas para el software, pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan medicines que son demasiado complejas, otras son tan esotéricas que poco profesionales tienen la esperanza de entenderlas y otras violan la nociones básicas intuitiva de lo que realmente es el software de alta calidad [Pressman, 2005].

Ejiogu [Eji, 1991] define un conjunto de atributos que deberían acompañar a las métricas efectivas del software. 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.

1.9 Calidad del software en Cuba.

El desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el estado cubano debido a la alta perspectiva económica que posee [Baeza, 1995], así como para el aseguramiento de un grupo importante de actividades del país. 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 [Febles, 2000]

Hoy en día la industria del software no ha acabado de salir de la fase artesanal debido a la desorganización y la falta de planificación. La disciplina ingeniería del software es relativamente reciente y sus conceptos más importantes están aun inmaduros. Se carece de un corpus de conocimiento aceptado, mayoritariamente que sirva de fundamentos, además de que se tiene una escasa presión del mercado [Buades, 2002].

En una organización inmadura existen procesos de software normalmente improvisados, y en caso de que se hayan especificado no se siguen rigurosamente. Presentan una organización reactiva (resolver crisis inmediatas). Los planes y presupuestos exceden sistemáticamente, al no estar basados en estimaciones realistas [Buades, 2002].

En las organizaciones inmaduras si existen plazos rígidos, se sacrifican funcionalidad y calidad del producto para satisfacer el plan [Buades, 2002].

1.10 Calidad del software en la Universidad de las Ciencias informáticas

La Universidad de las Ciencias Informáticas se ha concebido como agente de cambio en la industria cubana del software 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 la 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. Al cierre del curso pasado más del 90% de los estudiantes se encontraban vinculados a más de 200 proyectos de producción. En la UCI 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 [Febles, 2007].

Modelos de calidad que se siguen: CMMI e ISO 15504

Expectativas:

Programa efectivo en la mejora de procesos

Extender la gestión de la calidad hacia todos los procesos medulares de la universidad

Creación de red temática.

Conclusiones del capítulo I

- En este capítulo se trataron temas fundamentales que dieron paso al desarrollo y evolución del presente Trabajo de Diploma, se analizó las principales definiciones expuestas por personalidades expertas a lo que a calidad respecta, así como el estado del arte de la calidad del software.
- Según lo planteado por los diferentes autores se puede llegar a la conclusión de que el proceso de aseguramiento de la calidad del software dentro de un proyecto es un proceso sin fin, que debe llevarse paso a paso y del que no pueden esperarse resultados inmediatos.

CAPITULO 2: DISEÑO DEL PORCESO DE ASEGURAMIENTO DE LA CALIDAD PARA SISTEMA DE GESTION DE INVENTARIOS Y ALAMCENES (SIGIA).

Introducción

En este capítulo se abordarán temas que darán paso al desarrollo del presente trabajo de diploma. Para dar solución a los objetivos específicos se han realizado una serie de actividades que dan lugar al proceso de calidad llevado a cabo para el cumplimiento de los requisitos, tanto de la calidad como del proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

Para dar cumplimiento a los objetivos específicos y junto con ellos a los requisitos expuestos se trazaron las diferentes actividades:

Establecimiento de un Plan de Aseguramiento de la Calidad del software con el objetivo de proporcionar un mapa para institucionalizar la garantía de calidad del software, se utiliza como plantilla para actividades de aseguramiento y garantía de la calidad instituidas para el proyecto [Pressman, 2005].

Revisión de las actividades de Ingeniería del software para verificar su ajuste al proceso de software [Pressman, 2005].

Auditorias del control y la gestión de cambios para combinar los procedimientos humanos y las herramientas automáticas para proporcionar un mecanismo para el control de cambios [Pressman, 2005].

Definición de las métricas del software con el objetivo de proporcionar una manera sistemática de valorar la calidad del software basándose en un conjunto de reglas claramente definidas [Pressman, 2005].

2.1 Descripción del Proyecto productivo Sistema de Gestión de Inventarios y Almacenes (SIGIA)

El propósito del proyecto productivo Sistema de Gestión de Inventarios y Almacenes es obtener un sistema de gestión de inventarios, que permita controlar los Inventarios de los almacenes, la forma de almacenamiento de los productos, incluye además un módulo para la Gestión de Stock para saber si se necesita un producto, además del módulo de Gestión de Compras para la automatización de las compras de la entidad.

Una vez definida la necesidad, las metas y objetivos básicos del proyecto Sistema de Gestión de Inventario de Almacenes, se hace necesario el control y el seguimiento de las actividades.

Necesidad: En Cuba se han desarrollado numerosos sistemas para la gestión de almacenes, aunque la mayoría de estos están especializados para el funcionamiento de determinada empresa o tipo de empresas. Ejemplos de ellos los constituyen DRIM e IHMM, este último con el objetivo de usarse en instalaciones hoteleras. La ausencia de una aplicación que integrara todas las funcionalidades independientemente del tipo de empresa que haga uso de ella, sustentó la decisión de construir una alternativa con el fin de estandarizar el proceso de gestión de inventarios y almacenes, un software configurable a partir de las necesidades particulares de los clientes y que incorporara todas las regulaciones que establece el Ministerio de Finanzas y Precios.

Objetivos: El propósito del proyecto Sistema de Gestión de Inventario de Almacenes es obtener un Sistema de Gestión de Inventarios, que permita controlar los Inventarios de los almacenes, la forma de almacenamiento de los productos, incluye además un módulo para la Gestión de Stock para saber si se necesita un producto, además del módulo de Gestión de Compras para la automatización de las compras de la entidad [Repositorio, 2007].

Metas: Lo que se quiere lograr es un sistema informático seguro y estable que cumpla con los nuevos requisitos de interfaz y que permita:

Llevar el control de los activos de una empresa

Automatizar la gestión de los inventarios

Automatizar la gestión de las compras

Automatizar la generación de información actualizada de las entidades

2.1.1 Herramientas de desarrollo del proyecto productivo Sistema de gestión de Inventarios y Almacenes (SIGIA).

1. Para modelado se utiliza Rational Rose 2003 el cual demanda como mínimo 256 MB de RAM y recomendado 512 MB de RAM.
2. Para implementación como IDE de desarrollo Eclipse 3.2 el cual demanda como mínimo 768 MB de RAM y recomendado 1.0 GB de RAM al integrarlo con los plugins necesarios.
3. Para implementación los plugins serán:
   - WindowsBuilder, plugin que garantiza el trabajo con componentes visuales y reutilizables para la capa de presentación. Implementa el Framework Swing.
   - Capabilidades de Spring, para implementar el Framework Spring 2.0.
   - Hibernate Tool el cual facilita la ingeniería inversa y reversa de los ficheros de mapeo entre clases java y tablas de Base de Datos.
   - JUnit el cual facilita las pruebas de unidad.
   - SVNeclipse el cual facilita la integración con Subversión.
4. Como servidor de Base Datos SQL Server 2000. (256 MB de RAM).
5. Como servidor de control de versiones Subversión.

2.1.2 Papel del Administrador de calidad dentro del proyecto productivo Sistema de Gestión de Inventarios y Almacenes (SIGIA).

El administrador de calidad en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes comienza su trabajo con la confección del plan de aseguramiento de calidad de software, cerciorándose de que este se ajuste a los estándares y mecanismos de la calidad del software, además de monitorear las actividades referentes a la misma. Paralelamente y como soporte al plan de calidad realiza un plan de revisiones y auditorías, así como la realización de las misma para controlar las actividades de ingeniería de software del proyecto, teniendo en cuenta las herramientas (Listas de Chequeo), técnicas (RTF) y

metodologías (RUP) utilizadas. Otra de las actividades que complementan y apoyan el plan de aseguramiento de la calidad del software es la definición de las métricas de calidad, así como la coordinación y el control de cambios efectuados en el proyecto.

2.2 Proceso de Aseguramiento de la calidad del software en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

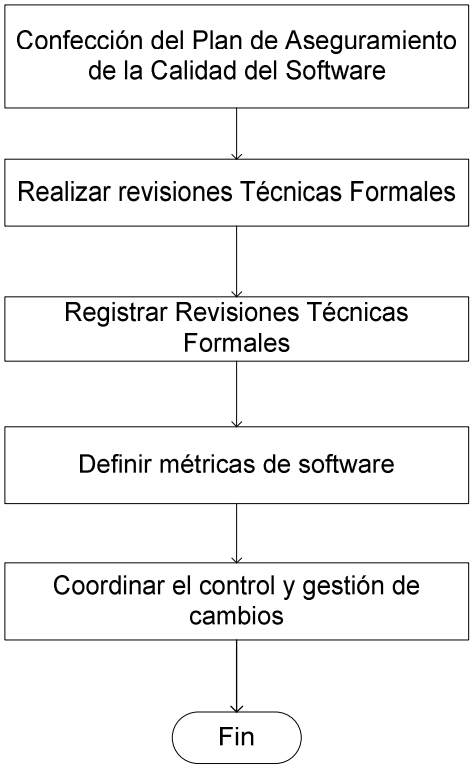


Diagrama 2.1 Proceso de Aseguramiento de la Calidad del Software para SIGIA

El proceso de Aseguramiento de la Calidad del Software en el proyecto Sistema de Gestión de Inventarios y Almacenes comienza con la confección del Plan de Aseguramiento de la Calidad del Software. Este plan incluye una serie de actividades que dan lugar al mismo. Seguido a la confección del plan se realizan las revisiones técnicas formales que se encuentran plasmadas en el plan de aseguramiento de la calidad del software a los diferentes artefactos generados en cada una de las iteraciones que se efectuarán en el proyecto (SIGIA). Estas revisiones técnicas formales son registradas con el fin de llevar un control y seguimiento de las mismas.

El proceso de aseguramiento de la calidad del software incluye la definición de las métricas de calidad con el propósito de mejorar el proceso sobre una base continua [Pressman, 2005].

Por último y no menos importante se coordina el control y la gestión de cambios del proyecto, para que este último fluya uniformemente durante su ciclo de vida.

2.3 Establecimiento del Plan de Aseguramiento de la Calidad del Software en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

El proceso de gestión de un proyecto de software se le da seguimiento con un conjunto de actividades que, globalmente, se denominan gestión de la calidad del software. La calidad del proyecto se basa en la determinación de las actividades y los recursos que se tienen que utilizar para que el producto final cumpla con los requisitos requeridos.

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, artefactos, y la calidad del proceso. Contiene el plan de revisiones y auditorias. Es mantenido durante todo el proceso de desarrollo del software [Rational Unified Process, 2003].

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.

2.3.1 Actividades que dan lugar al Plan de Aseguramiento de la Calidad del Software.

Para la obtención y seguimiento de las actividades de ingeniería del software, el proyecto Sistema de Gestión de Inventarios y Almacenes se vio en la necesidad de la realización del Plan de Aseguramiento de Calidad del Software para el control de dichas actividades. Para llevar a cabo el plan de aseguramiento de calidad del software se efectuaron diferentes actividades que apoyan el desarrollo del mismo. Estas actividades abarcan todos los puntos y fronteras del plan, dándole cumplimiento y seguimiento al proceso de desarrollo del

proyecto, de modo tal que el producto final SIGIA cumpla con las especificaciones dadas por el cliente y la calidad requerida. (Ver fig. 2.2).

Cada una de estas actividades es un paso importante en el plan de aseguramiento de la calidad del software propuesto por Rational Unified Process (RUP), al cual se le da cumplimiento en el Sistema de Gestión de Inventarios y Almacenes.

Fig. 2.2 Diagrama de las actividades del Plan de Aseguramiento de Calidad del Software.

Las actividades para llevar a cabo el plan de aseguramiento de calidad del software son:

1. *Coincidir en la misión:* Su propósito no es más que, los responsables de la ejecución del plan estén de acuerdo con el uso más eficaz de los recursos para cada iteración,

además de concordar con los objetivos de la iteración.

2. *Organización*: Esta actividad muestra la organización del proyecto Sistema de Gestión de Inventarios y Almacenes, con el objetivo de identificar claramente los involucrados en el desarrollo del mismo.
3. *Definir las tareas y responsabilidades*: Esta actividad tiene como objetivo identificar las tareas y las responsabilidades del grupo de aseguramiento de la calidad del software dentro del proyecto para llevar el control de pruebas, revisiones y auditorías que se ejecutan en cada iteración del proyecto.
4. *Muestra de la documentación utilizada*: En esta actividad se muestra una lista de los diferentes planes que se han realizado en el proyecto y a los cuales se va a hacer referencia en el plan de aseguramiento de la calidad del software.
5. *Resumen de estándares y guías*: El objetivo de esta actividad es mostrar los estándares y guías utilizados por el plan de SQA.
6. *Métricas*: En esta actividad se definen las métricas para la evaluación del proyecto.
7. *Plan de revisiones y auditorías*: Esta es la actividad más importante dentro del plan de aseguramiento de la calidad del software, ya que muestra el resumen de revisiones y auditorías que se van a ejecutar en el proyecto, así como también el momento preciso en el que se van a realizar.
8. *Organización y responsabilidades*: En esta sección se definen los involucrados en cada una de las revisiones y auditorías efectuadas.
9. *Definición de las herramientas, técnicas y metodología*: Esta actividad es la encargada de definir todas las herramientas y técnicas de detección de defectos utilizados para llevar a cabo las revisiones del software. Así como también la metodología de desarrollo utilizada para la realización del proyecto.
10. *Plan de desarrollo del software*: Su principal y único objetivo es coordinar el desarrollo de todos los planes y contenidos asociados. Esta actividad es esencial para el

desarrollo del Plan de Aseguramiento de Calidad del Software.

11. *Solicitud de cambios de revisión*: Tiene como objetivo determinar si la solicitud de cambio debe ser aceptada o rechazada. En caso de que sea aceptada esta actividad evalúa la prioridad, el esfuerzo, el horario, y así sucesivamente para determinar si el cambio está en el alcance para el lanzamiento en curso. Esta actividad es la última en el proceso para el desarrollo del Plan de Aseguramiento de Calidad del Software.

Con el fin de darle seguimiento a las actividades de ingeniería del software, con el propósito de que el producto final tenga la calidad requerida, se confecciona el plan de aseguramiento de la calidad, el cual tiene un propósito específico sistematizar y planificar todas las acciones realizadas en el proyecto Sistema de gestión de Inventarios y Almacenes. Su realización comienza desde el principio de desarrollo del software y termina con el producto final.

El equipo de aseguramiento de la calidad (SQA) del proyecto Sistema de Gestión de Inventarios y Almacenes (SIGIA), se trazó diferentes actividades para dar cumplimiento al plan.

En el comienzo del proyecto se realiza un plan de iteraciones, el cual muestra los recursos utilizados para cada una de ellas. Se realizó un estudio de las iteraciones expuestas por el plan, hasta concordar con los recursos utilizados por las mismas. Al haber analizado detenidamente cada una de las iteraciones, se decidió llevar a cabo la organización del proyecto, cada una de las personas del proceso de desarrollo junto con el rol que iban a desempeñar.

Al estar confeccionado el equipo de aseguramiento de la calidad se decidió aplicarle ciertas tareas y responsabilidades a las personas involucradas en el mismo, dichas tareas quedarían plasmadas en el plan de aseguramiento de la calidad del software. Estas tareas y responsabilidades responden a la puesta en marcha de un proceso de aseguramiento de la calidad para que se cumpla con los requisitos del proyecto, y estarían presente cada una de las personas del grupo de desarrollo del software junto con el equipo de calidad encargado de dicha actividad.

Para llevar a cabo las tareas y responsabilidades que no son más que las pruebas, revisiones y auditorías se utilizaron diferentes documentos que fueron de gran ayuda para el plan. Dentro de estos documentos se encuentra el Plan de pruebas, Plan de riesgos, Plan de gestión de Configuración, y el más importante el Plan de Desarrollo del Software el cual contiene todo lo referente al proyecto Sistema de Gestión de Inventarios y Almacenes.

Todavía el plan no se encontraba en su fase terminal pues le faltaban los estándares y guías que apoyarían al mismo estos estándares le darían validez al documento, junto con el Plan de Revisiones y Auditorías que se utiliza para realizar las actividades que controlarán el proceso de desarrollo de software a lo largo de su ciclo de vida. Este plan muestra el momento exacto en que se deben realizar las revisiones tanto a los artefactos generados como al mismo producto en general, para de esta forma garantizar que SIGIA se un producto confiable para los clientes.

Las revisiones y auditorías que tienen lugar en el proyecto son llevadas a cabo por el grupo de aseguramiento de la calidad junto con el personal del proyecto involucrado en la misma. Esto quiere decir que si se le está aplicando las revisiones a los requisitos en alguna u otra iteración se estaría llevando a cabo por el grupo de aseguramiento de la calidad del software junto con el personal del proyecto que llevó a cabo esta fase del proceso de desarrollo, así sucedería con las demás actividades de ingeniería del software expuestas por Rational Unified Process que es la metodología de desarrollo de software utilizada por el proyecto Sistema de Gestión de Inventarios y Almacenes.

Estas revisiones se realzarán utilizando diferentes herramientas y técnicas, en el proyecto se utilizan las revisiones técnicas formales como técnica y para poder realizar dichas revisiones se utilizan las listas de chequeo como herramienta fundamental.

En el momento que se realiza una solicitud de cambio por cualquiera del personal que trabaja en el proyecto, esta solicitud se analiza por la autoridad del control de cambios (ACC) que no es más que el grupo que toma la decisión final del estado y la prioridad del cambio, el mismo realiza las actividades pertinentes que darán paso al cambio o no, mientras que el equipo de aseguramiento de la calidad realiza las auditorías pertenecientes

al cambio para controlar los cambios ocurridos dentro del proceso. Estos cambios no pueden ocurrir sin una previa autorización por el ACC, el cual a su vez va a estar supervisado por el grupo de aseguramiento de la calidad (SQA).

2.2.2 Documento Plan de Aseguramiento de la Calidad del Software

Introducción

Propósito

El objetivo que se persigue con la realización del Plan de Aseguramiento de la Calidad del Software (SQA), 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.

Alcance

Sistema Gestión de Inventarios y Almacenes (SIGIA)

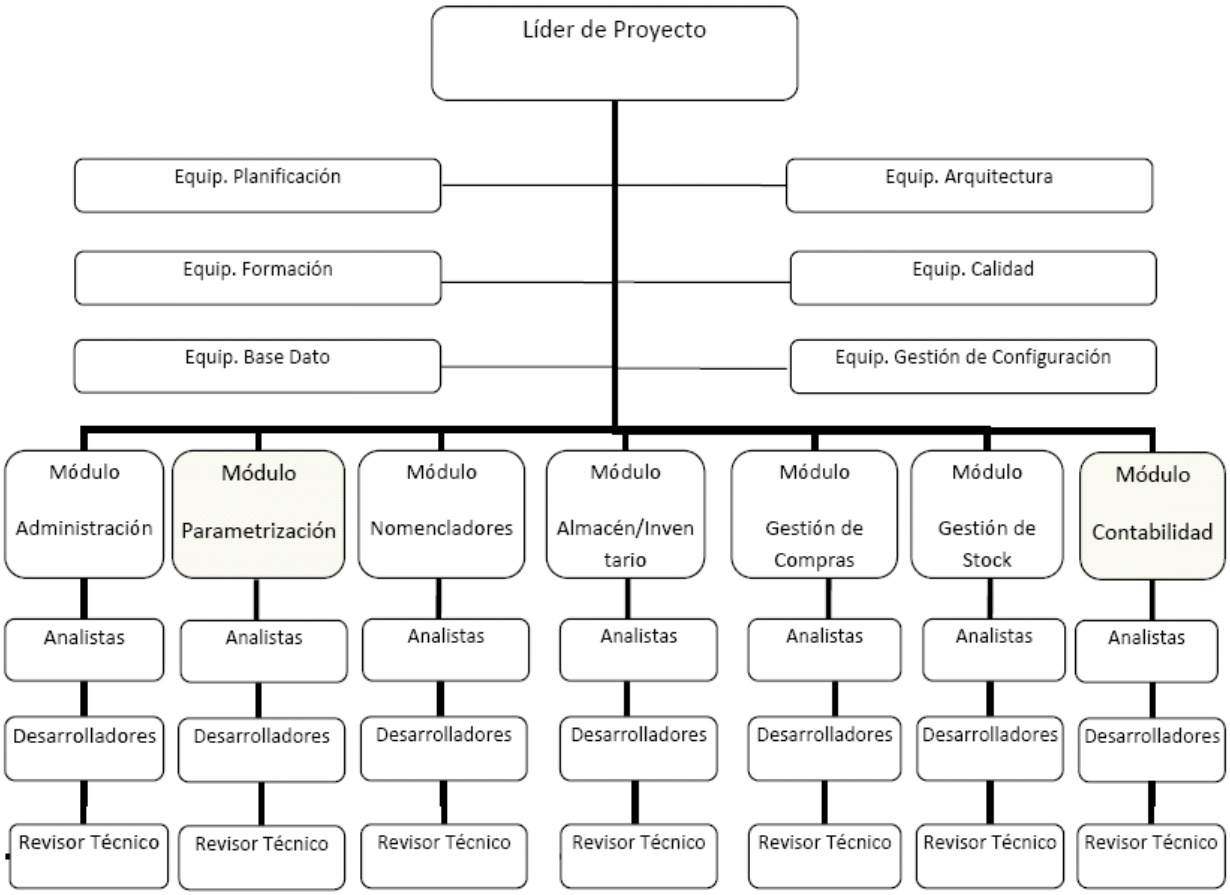
Definiciones, Acrónimos y Abreviaturas

SQA (Software Quality Assurance: Aseguramiento de la Calidad del software)

Referencias

Título	**Fecha**	**Autor**	**Ubicación (anexo, documento, etc.)**
Plan de pruebas	6/3/2007	Ing. Pruebas.	Documento
Plan de desarrollo del proyecto	6/12/2006	Equipo de planificación	documento
Plan de Gestión de Configuración	30/12/2006	Equipo Gestión de configuración	documento
Plan de gestión de riesgos	27/4/2006	Equipo de planificación	documento
Plan de Iteración	24/4/2006	Equipo de planificación	documento

Estructura del equipo de desarrollo



Definir tareas y responsabilidades

Tarea de Aseguramiento de calidad	**Precondición Al finalizar la fase:**	**Poscondición Antes de la fase:**	**Responsable**	**Comentarios**
Prueba1	Implementación de la iteración 1	Fase de inicio de la iteración 2	Ingeniero de pruebas	
Prueba 2	Implementación de la iteración 2	Fase de inicio de la iteración 3	Ingeniero de pruebas	
Prueba 3	Implementación de la iteración 3	Fase de inicio de la iteración 4	Ingeniero de pruebas	
Prueba 4	Implementación de la iteración 4	Fase de inicio de la iteración 5	Ingeniero de pruebas.	
Prueba 5	Implementación de la	Fase de inicio de la	Ingeniero de	

	iteración 5	iteración 6	pruebas	
Prueba 6	Implementación de la iteración 6	Fase transición	Ingeniero de pruebas	
Auditoria 1	Fase de implementación de la iteración 1	Fase de inicio de la iteración 2	Administrador de calidad	
Auditoría 2	Fase de implementación de la iteración 2	Fase de inicio de la iteración 3	Administrador de calidad	
Auditoría 3	Fase de implementación de la iteración 3	Fase de inicio de la iteración 4	Administrador de calidad	
Auditoría 4	Fase de implementación de la iteración 4	Fase de inicio de la iteración 5	Administrador de calidad	
Auditoría 5	Fase de implementación de la iteración 5	Fase de inicio de la iteración 6	Administrador de calidad	
Auditoría 6	Fase de implementación de la iteración 6	Fase de transición	Administrador de calidad	
Revisión 1	Levantamiento de requisitos (fase inicio) de la iteración 1	Arquitectura (fase de elaboración) de la iteración 1	Administrador de calidad	
Revisión 2	Arquitectura (fase de elaboración) de la iteración 1	Análisis y diseño (fase de elaboración) de la iteración 1	Administrador de calidad	
Revisión 3	Análisis y diseño (fase de elaboración) de la iteración 1	Levantamiento de requisitos (fase de incio) de la iteración 2	Administrador de calidad.	
Revisión 4	Levantamiento de requisitos (fase de incio) de la iteración 2	Arquitectura (fase de elaboración) de la iteración 2	Administrador de calidad	
Revisión 5	Arquitectura (fase de elaboración) de la iteración 2	Análisis y diseño (fase de elaboración) de la iteración 2	Administrador de calidad	
Revisión 6	Análisis y diseño (fase de elaboración) de la iteración 2	Levantamiento de requisitos (fase de incio) de la iteración 3	Administrador de calidad	
Revisión 7	Levantamiento de requisitos (fase de incio) de la iteración 3	Arquitectura (fase de elaboración) de la iteración 3	Administrador de calidad	

Revisión 8	Arquitectura (fase de elaboración) de la iteración 3	Análisis y diseño (fase de elaboración) de la iteración 3	Administrador de calidad	
Revisión 9	Análisis y diseño (fase de elaboración) de la iteración 3	Levantamiento de requisitos (fase de incio) de la iteración 4	Administrador de calidad	
Revisión 10	Levantamiento de requisitos (fase de incio) de la iteración 4	Arquitectura (fase de elaboración) de la iteración 4	Administrador de calidad	
Revisión 11	Arquitectura (fase de elaboración) de la iteración 4	Análisis y diseño (fase de elaboración) de la iteración 4	Administrador de calidad	
Revisión 12	Análisis y diseño (fase de elaboración) de la iteración 4	Levantamiento de requisitos (fase de incio) de la iteración 5	Administrador de calidad	
Revisión 13	Levantamiento de requisitos (fase de incio) de la iteración 5	Arquitectura (fase de elaboración) de la iteración 5	Administrador de calidad	
Revisión 14	Arquitectura (fase de elaboración) de la iteración 5	Análisis y diseño (fase de elaboración) de la iteración 5	Administrador de calidad	
Revisión 15	Análisis y diseño (fase de elaboración) de la iteración 5	Levantamiento de requisitos (fase de incio) de la iteración 6	Administrador de calidad	
Revisión 16	Levantamiento de requisitos (fase de incio) de la iteración 6	Arquitectura (fase de elaboración) de la iteración 6	Administrador de calidad	
Revisión 17	Arquitectura (fase de elaboración) de la iteración 6	Análisis y diseño (fase de elaboración) de la iteración 6	Administrador de calidad	
Revisión 18	Análisis y diseño (fase de elaboración) de la iteración 6	Fase de transición	Administrador de calidad	
Certificación final del producto	Prueba	Implantación	Administrador de calidad	
Certificación por parte del Cliente	Certificación final del producto		-Administrador de calidad, -Líder del proyecto -Clientes	
Evaluación de la	Un mes después de		-Administrador	

satisfacción del cliente	la implantación del producto		de calidad -Clientes	

Plan de revisiones y Auditorías

Tareas generales de Revisiones y Auditorias.

Las revisiones del software sirven para purificar, las actividades de ingeniería del software que suceden como resultado del análisis, el diseño y la codificación.

- **Revisión de los Requerimientos:** Asegurar que el sistema tenga condiciones o capacidades para satisfacer las exigencias expuestas por el cliente.
  Artefactos: Lista de Requerimientos
- **Revisión de la Arquitectura:** Garantizar un diseño preliminar.
  Artefactos: Arquitectura.
- **Revisión del análisis y diseño:** Asegurar que el modelo se acerque lo mas posible a la implementación para tener una idea mas clara de lo que se quiere en el sistema.
  Artefactos: Modelo de casos de uso del sistema.
  Actores que se relacionan con el modelo.
  Clases del diseño.
  Base de datos.
- **Auditoria de la Configuración Funcional:** Para verificar que todos los requerimientos han sido cumplidos
  Artefactos: Capa de presentación
  Capa de negocio
  Capa de acceso a datos
- **Auditoria de la Configuración Física:** Para verificar que el software y su documentación están completos y listos para entregar.
  Artefactos: Todos los documentos que se utilizaron en el proyecto para un mejor funcionamiento del software
- **Auditoria del proceso**

- **Revisión del proceso**
- **Revisión Administrativa:** Revisión de aprobación del proyecto, Revisión de la planificación del proyecto y Revisión del plan de iteración.

Revisión post-mortem: Revisión de la aceptación de la iteración y revisión de la aceptación del proyecto.

No. de Revisión	Tipo	Objetivos	Descripción (Iteración)	Fase del Proyecto	Responsable
1	Revisión administrativa	1. Revisión de la aprobación del proyecto. 2. Revisión del plan de iteraciones	Inicio del proyecto	inicio	-Administrador de calidad. -Equipo de gestión de proyecto
2	Revisión post-mortem	Revisión de la aceptación de la iteración.	Inicio de la iteración 1	inicio	Administrador de calidad
3	Revisión del Modelo del Negocio	Artefacto: Diagrama de casos de uso del negocio	Inicio de la iteración 1	Inicio	Administrador de Calidad
4	Revisión de los requerimientos	1. Módulo Almacén/Inventario 2. Módulo Nomencladores	Iteración 1	inicio	Administrador de calidad
5	Revisión de la arquitectura	Garantizar diseño preliminar Artefacto: Arquitectura	Iteración 1	elaboración	Administrador de calidad
6	Revisión análisis y diseño	1. Módulo almacén/Inventario 2. Modulo nomencladores	Iteración 1	elaboración	Administrador de calidad
7	Auditoría configuración funcional	1. Capa de presentación. 2. Capa de negocio. 3. Capa de acceso a datos.	Iteración 1	construcción	Administrador de calidad
8	Revisión post-mortem	Revisión de la aceptación de la iteración.	Inicio de la iteración 2	inicio	
9	Revisión de los requerimientos	Módulo Administración	Iteración 2	inicio	Administrador de calidad

10	Revisión de la arquitectura	Artefacto: Arquitectura	Iteración 2	elaboración	Administrador de calidad
11	Revisión análisis y diseño	Módulo Administración	Iteración 2	elaboración	Administrador de calidad
12	Auditoria de la configuración funcional	1. Capa de presentación. 2. Capa de negocio. 3. Capa de acceso a datos.	Iteración 2	construcción	Administrador de calidad
13	Revisión post-mortem	Revisión de la aceptación de la Iteración.	Inicio de la iteración 3	inicio	Administrador de calidad.
14	Revisión de los requerimientos	Módulo Gestión de compras.	Iteración 3	inicio	Administrador de calidad
15	Revisión de la arquitectura	Artefacto: Arquitectura	Iteración 3	elaboración	
16	Revisión análisis y diseño	Módulo Gestión de compras.	Iteración 3	elaboración	Administrador de calidad
17	Auditoría de la configuración funcional.	1. Capa de presentación 2. Capa de negocio. 3. Capa de acceso a datos.	Iteración 3	construcción	Administrador de calidad
18	Revisión post-mortem	Revisión de la aceptación de la iteración.	Inicio de la iteración 4		Administrador de calidad
19	Revisión de los requerimientos	Módulo gestión de stock.	Iteración 4	Inicio	Administrador de calidad
20	Revisión de la arquitectura	Artefacto: Arquitectura	Iteración 4	elaboración	Administrador de calidad
11	Revisión análisis y diseño	Módulo gestión de stock.	Iteración 4	elaboración	Administrador de calidad
22	Auditoría de la configuración funcional.	1. Capa de presentación. 2. Capa de negocio. 3. Capa de acceso a datos.	Iteración 4	construcción	Administrador de calidad
23	Revisión post-mortem	Revisión de la aceptación de la iteración	Inicio de la iteración 5		Administrador de calidad

24	Revisión de los requerimientos	Módulo Parametrización	Iteración 5	Inicio	Administrador de calidad
25	Revisión de la arquitectura	Artefacto: Arquitectura	Iteración 5	elaboración	Administrador de calidad
26	Revisión análisis y diseño	Módulo Parametrización	Iteración 5	elaboración	Administrador de calidad
27	Auditoría configuración funcional	1. Capa de presentación 2. Capa de negocio 3. Capa de acceso a datos	Iteración 5	construcción	Administrador de calidad
28	Revisión post-mortem	Revisión de la aceptación de la iteración	Inicio de la iteración 6		Administrador de calidad
29	Revisión de los requerimientos	Módulo contabilidad	Iteración 6	Inicio	Administrador de calidad
30	Revisión de la arquitectura	Artefacto: Arquitectura	Iteración 6	elaboración	Administrador de calidad.
31	Revisión de análisis y diseño	Módulo contabilidad	Iteración 6	elaboración	Administrador de calidad
32	Auditoría de la configuración funcional	1. Capa de presentación. 2. Capa de negocio 3. Capa de acceso a datos.	Iteración 6	construcción	Administrador de calidad
33	Auditoría de la configuración física	Verificación de toda la documentación del software. Verificación del producto final (SIGIA)		Transición	Administrador de calidad

Organización y responsabilidades

Tipo de revisión	**Personal involucrado en las actividades de revisión y auditorias.**	**Descripción de las tareas y responsabilidades de cada uno de ellos.**
Revisión administrativa	-Líder del proyecto	1. Inicia el proyecto, e identifica los riesgos y posibilidades. 2. Planifica el desarrollo, lo

		monitorea y lo controla. 3. Define los roles y la organización del proyecto. 4. Controla las iteraciones y las fases. 5. Resuelve problemas en el equipo. 6. Coordina y asigna las actividades para el trabajo.
Revisión post-mortem	-Analista del sistema -Líder del proyecto	1. Desarrollar el Plan de Administración de Requerimientos. 2. Desarrollar Documento Visión 3. Identificar las demandas de los stakeholders. 4. Definir un vocabulario común con el cliente. 5. Buscar Actores y Casos de Uso. 6. Estructurar el modelo de Casos de Uso del sistema.
Revisión de los requerimientos.	-Arquitecto del software -Analista del sistema	1. Prioriza los casos de uso 2. Realiza el análisis de la arquitectura y prueba su viabilidad. 3. Estructura el modelo de análisis, implementación y despliegue. 4. Incorpora elementos ya construidos. 5. Identifica los mecanismos y algunos elementos de diseño
Revisión de análisis y diseño.	-Diseñador de interfaz de usuario.	1. Realizar el prototipo de la interfaz de usuario.

	-Administrador de base de datos. -Revisor técnico -Analista del sistema -Arquitecto del software	1. Construir el diseño de la base de datos. 1. Este relacionado con la revisión técnica a los artefactos generados durante el proceso (modelo de negocio, de sistema, análisis, diseño…) 2. Implementar y ejecutar la colección de pruebas al sistema. 3. Analizar las fallas para encontrar los errores y solicitar los cambios.
Auditoria de la configuración funcional	-Ingeniero de implementación -Arquitecto del software	1. Implementar los elementos diseñados. 2. Desarrollar y ejecutar las pruebas de implementación sobre las clases programadas.
Auditoria de la configuración física.	-Analista del sistema -Arquitecto del software -Líder del proyecto	1. Verificar que todos los documentos realizados en el proyecto estén completos y listos para entregar

Documentación utilizada por el plan

- ✓ Documento Plan de Pruebas versión 1.0, Sistema Gestión de Inventarios y Almacenes (SIGIA)
- ✓ Documento Plan de Desarrollo del proyecto versión 1.0, Sistema Gestión de Inventarios y Almacenes (SIGIA)
- ✓ Documento Plan de Gestión de Configuración versión 1.0, Sistema Gestión de Inventarios y Almacenes (SIGIA)

- ✓ Documento Plan de Riesgos versión 1.0, Sistema Gestión de Inventarios y Almacenes (SIGIA)
- ✓ Documento Plan de resolución de problemas versión 1.0, Sistema Gestión de Inventarios y Almacenes (SIGIA)

Resumen de estándares y guías

Estándar	Ubicación	Comentarios
Rational Rose 2003	Plan de desarrollo de software	
Eclipse 3.2	Plan de desarrollo del software	
WindowsBuilder	Plan de desarrollo del proyecto	
Capabilidades de Spring		
Hibernate Tool		
JUnit		
SVNeclipse		
Casos de pruebas	Plan de pruebas	
Manuales	Manual de usuario	
Base Datos SQL Server 2000	Plan de desarrollo del software	
Subversión	Plan de desarrollo del software y Plan de gestión de Configuración del software	

2.4 Revisión de las actividades de ingeniería del software

El proceso de revisiones o inspección como suele llamarse en muchas de las bibliografías consultada, tiene gran importancia dentro del proceso de Aseguramiento de la Calidad del Software, pues éste ayuda a detectar defectos en etapas tempranas del desarrollo de los proyectos de software, lo que constituye una garantía de la calidad del producto final.
Las revisiones técnicas formales tienen tres elementos:

- Informe escrito del estado del producto revisado.
- La participación activa y abierta de todos los del grupo de revisión.
- Total responsabilidad de todos los participantes en la calidad de la revisión.

2.4.1 Ventajas de las revisiones técnicas formales

- *Reduce sustancialmente el costo del software*: Esto provoca que las pruebas realizadas al producto final sean menos costosas ya que va a hacer la menor cantidad de errores cometidos por parte de los participantes en el proyecto de desarrollo del software. En otras palabras, las revisiones efectuadas a lo largo del proceso de desarrollo tienen más efectividad que las realizadas solamente en el momento de la implantación del producto.
- *Tiene gran valor educativo para los participantes*: Esto permite que los participantes en el proceso de desarrollo no cometan los mismos errores a la hora de realizar un nuevo software.
- *Se utiliza para comunicar la información técnica*: Cuando se efectúan las revisiones, es un momento en que la información que fluye durante el proceso de desarrollo sea conocida por todos los involucrados en el proyecto.
- *Fomenta la seguridad y la continuidad*: Estas revisiones dan cierto grado de seguridad y continuidad para el producto, ya que deben de cumplir con diferentes parámetros o factores de calidad del software que permiten el desarrollo continuo y seguro del proyecto.

2.4.2 Revisiones Técnicas Formales

Con la realización del Plan de Aseguramiento de la calidad, el cual lleva incluido un plan de revisiones y auditorías, queda por definido la actividad más importante para el control y mantenimiento de la calidad del software dentro del proyecto Sistema de Gestión de Inventarios y Almacenes con el proceso de revisiones del mismo.

Las revisiones del software surgen a partir de la necesidad de producir software de calidad. El proceso de revisiones es de gran importancia para el proceso de Aseguramiento de Calidad del software, pues este ayuda a detectar defectos en etapas tempranas del desarrollo de los proyectos de software, lo que constituye una garantía de calidad del producto final.

No obstante, hay varios momentos en los que no deberían omitirse; el primero al finalizar la etapa de determinación de los requerimientos, para detectar lo antes posible defectos relacionados con los requerimientos contratados, las necesidades de los usuarios e involucrados en el proyecto y otros que pueden dar al traste con un software que no represente adecuadamente las expectativas de los clientes.

Otras inspecciones deben realizarse al concluir las etapas de análisis, diseño e implementación respectivamente, para garantizar eliminar la mayor cantidad posible de defectos antes de pasar a la fase siguiente.

La última inspección debe llevarse a cabo, una vez terminada la primera versión del producto, de manera tal que se verifique si efectivamente el producto terminado posee todos los requisitos técnicos y funcionales necesarios que le permitan cumplir con los objetivos y requerimientos contratados y satisfacer adecuadamente las necesidades del los clientes [Delgado, 2006]. El proyecto productivo Sistema de Gestión de Inventarios y Almacenes utiliza para garantizar la calidad del producto SIGIA las Revisiones Técnicas Formales (RTF).

El objetivo primario de las Revisiones Técnicas Formales es encontrar errores durante el proceso, de forma que no se conviertan en defectos después de la entrega del producto. El

beneficio obvio de estas revisiones es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software [Pressman, 2005].

Las revisiones técnicas formales (RTF) es una actividad de garantía de calidad del software llevada a cabo por los ingenieros del software (y otros).

Objetivos de las revisiones técnicas formales:

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 de forma uniforme.
5. Hacer que los proyectos sean más manejables.

Estas revisiones también sirven de entrenamiento permitiendo que los ingenieros más jóvenes puedan observar los diferentes enfoques de análisis, diseño e implementación del software. Las Revisiones Técnicas Formales se utilizan también para promover la seguridad y la continuidad, ya que varias personas se familiarizarán con partes del software que, de otro modo, no hubieran visto nunca [Pressman, 2005].

La cantidad de revisiones que deben ser realizadas a un proyecto depende de las características de éste y es una decisión que debe ser tomada por el Administrador de Aseguramiento de Calidad conjuntamente con el Líder de Proyecto y documentada debidamente en el Plan de Revisión del Proyecto de Software.

2.4.3 Procedimiento para introducir las revisiones en el proceso de desarrollo de software.

- Construir grupo de Aseguramiento de Calidad del Software, en caso de que el proyecto sea pequeño definir Administrador de Calidad que se encargaría de las funciones correspondientes a este grupo.
- Definir roles adecuados para el grupo de desarrollo del proyecto, entre los que

deben estar: Líder del proyecto, Desarrolladores y Administrador de Calidad.

- Ejecutar revisión del producto terminado que ha desarrollado el proyecto.
- Definir e introducir métricas que permitan analizar la efectividad de las revisiones.
- Introducción de herramientas automatizadas para planificación y seguimiento de las revisiones.

2.4.4 Procedimiento para introducir las revisiones en el proceso de desarrollo del proyecto Sistema de Gestión de Inventarios y Almacenes

En el proyecto productivo Sistema de Gestión de Inventarios y Almacenes para introducir las revisiones en el proceso de desarrollo lo primero que se efectúo fue la planificación de las mismas, o sea se desarrolló dentro del plan de aseguramiento de la calidad del software un plan de revisiones que le dan seguimiento a cada una de las actividades de ingeniería del software, con el objetivo de que estas cumplan con las especificaciones expuestas por los clientes.

Estas revisiones efectuadas en a lo largo del proceso de desarrollo del software permiten un mejor entendimiento de las actividades de ingeniería del software efectuadas en SIGIA. Luego de una buena planificación de las revisiones se establecen las herramientas con las cuales se efectuarán las revisiones, en este caso serían las lista de chequeo. A continuación se procede a efectuar la reunión y se mantiene un registro de la misma (Fig. 2.3).

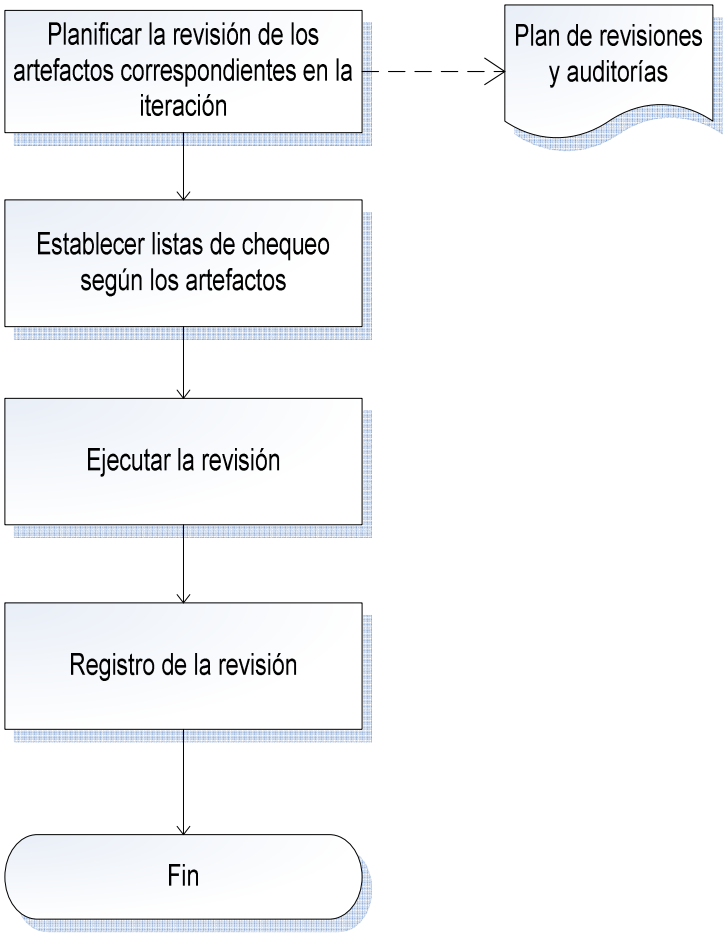


Fig. 2.3 Diagrama del proceso general de las revisiones que se efectuarán en SIGIA

Una vez llegado el momento de la revisión definido en el plan de revisiones y auditorías del proyecto, se deja constancia de la misma en un formato (Ver Anexo 1).

Una vez concluida la Revisión es importante dar el seguimiento adecuado a los defectos, tarea que corresponde al Administrador de Calidad del proyecto. No obstante el resultado satisfactorio o no de la revisión reporta experiencia, no solo para los desarrolladores del proyecto en cuestión sino también para el Administrador de Calidad que deberá sacar experiencia de cada revisión y analizar dónde puedan estar los problemas y utilizar la información resultante de las inspecciones para el mejoramiento continuo del proceso de desarrollo del software.

Esto sólo es posible si se registran los defectos adecuadamente y si se definen métricas que permitan saber la eficiencia de las inspecciones, entre otros aspectos.

2.4.5 Herramienta para efectuar las Revisiones Técnicas Formales (RTF) en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

Las herramientas utilizadas para llevar a cabo el proceso de revisiones en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes son las listas de chequeo, estas listas de chequeo se utilizan para evaluar los atributos de calidad que asegurarán que el producto final cumpla con los requisitos establecidos. El proyecto se encuentra en la primera iteración por tanto las revisiones realizadas a lo largo del proceso serán solamente a los módulos almacén/inventario y nomencladores respectivamente. Las listas de chequeo se llevaran a cabo desde el negocio hasta la implementación de esos módulos.

Listas de chequeo utilizadas en el proceso de desarrollo en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

1. Lista de chequeo para modelo de negocio.

   En el proyecto Sistema de Gestión de Inventarios y Almacenes se realiza el modelo del negocio con el fin de que todos los involucrados en el mismo comprendan los procesos y las automatizaciones que se van a realizar en la entidad. En este caso se concluye que el objetivo principal del modelado del negocio que no es más que describir cada proceso que se va a realizar en el negocio, los roles que intervienen en las actividades que complementan el proceso, así como también definir las diferentes reglas.

   Al mismo tiempo que se realiza el modelado del negocio se aplica una revisión al mismo, utilizando la herramienta lista de chequeo para modelo del negocio, esta lista de chequeo permite que todos los casos de uso, así como también el flujo de trabajo del modelo del negocio, los actores que intervienen en el flujo, tengan una mejor comprensión para el personal que va automatizar la entidad.

La lista de chequeo del modelo del negocio también se aplica al diagrama de actividades del negocio, con el objetivo de que dicho diagrama brinde el momento exacto en que ocurren las actividades del negocio.

En el proyecto productivo Sistema de Gestión de Inventarios y Almacenes se realizó una revisión al modelo del negocio utilizando la lista de chequeo referente a este artefacto.

La lista de chequeo de modelo del negocio fue de gran importancia para el proceso de desarrollo, ya que dio lugar a un mejor entendimiento del negocio, como también a una mejor captura de requisitos.

2. Lista de chequeo para etapa de levantamiento de requisitos.

En el caso de la etapa de levantamiento de requisitos fue un poco mas intensa la aplicación de las listas de chequeo ya que es la etapa más importante dentro del proceso de desarrollo de software, debido a que si no se hace una buena captura de requisitos pues entonces no se cumple con las especificaciones del cliente.

La lista de chequeo perteneciente a esta etapa del desarrollo del software responde a una serie de parámetros que permiten evaluar la calidad de los requisitos dando lugar a la calidad del software, que no es otra cosa que cumplir con las exigencias expuestas por los clientes. Los parámetros que llevan consigo esta lista de chequeo son:

- Claridad: Este parámetro persigue que los requisitos, la estructura y formato del documento sean comprensibles tanto para el cliente como para el personal del equipo de desarrollo.
- Concisión: Este parámetro permite que las descripciones, y las tablas, cada frase expuesta aporten algo a la especificación de los requisitos.
- Completitud interna: El objetivo de este parámetro es que n las tablas, diagrama, anexos, las terminologías, las figuras y la tabla de contenido estén bien definidas y referenciadas en el documento.

- Completitud externa: El propósito de este parámetro es que los requerimientos de funcionalidad tanto de software como de hardware, las auditorias, las restricciones de diseño, las interfaces externas entre otros requerimientos estén relacionados con los requisitos del software. Este parámetro tiene como segundo objetivo verificar los cambios efectuados en el proyecto.
- Consistencia interna: permite que los requerimientos no entren en conflicto.
- Consistencia externa: que los requerimientos son consistentes con el negocio y las justificaciones del proyecto.
- Trazabilidad: el propósito de este parámetro es que cada requerimiento obedezca a una necesidad específica del usuario.
- Verificabilidad: el objetivo de este parámetro es que cada requerimiento al ser implementado se pueda verificar su funcionamiento.
- Modificabilidad: Este parámetro se refiere a que el documento de los requisitos este expresado de una forma clara y lógica, además de que las redundancias sean mínimas.
- Priorización: Que cada requerimiento tenga asignado un nivel de prioridad para cliente o usuario.

La lista de chequeo en cuestión fue realizada en el proyecto Sistema de Gestión de Inventarios y Almacenes, la misma muestra los resultados de la primera iteración del proyecto (Ver anexo 2)

3. Lista de chequeo para casos de uso del sistema.

   Teniendo en cuenta que los requisitos cumplan con las especificaciones de los clientes, se agrupan en casos de uso que no son más que las operaciones que se van a implementar en el sistema, estos casos de uso responden a un requisito funcional. A este artefacto generado por la fase de especificación de requisitos se le aplica la lista de chequeo que supervisa el buen funcionamiento del mismo. El

objetivo principal de la lista de chequeo de los casos de uso esta centrado mayormente en las especificaciones de los casos de uso.

En el proyecto productivo Sistema de Gestión de Inventarios y Almacenes para la primera iteración del proyecto se tienen una serie de casos de uso que dan lugar a los módulos de Almacén/Inventario y Nomencladores, a estos dos módulos es a los que se le aplicaron las listas de chequeo para los casos de uso, verificando que la iteración en el proyecto pasara a la fase de análisis y diseño con la menor cantidad de errores.

4. Lista de chequeo para análisis y diseño.

   En el momento en que los casos de uso están acoplados por requisitos y las descripciones están completas y revisadas por el equipo de aseguramiento de la calidad del software del proyecto productivo Sistema de Gestión de de Inventarios y Almacenes, comienza su desarrollo en la fase de análisis y diseño donde se realizan las clases y los diagramas de interacción con el propósito de proporcionar al desarrollador de software una guía para la implementación del sistema.

   En esta fase el proyecto efectuó diferentes revisiones al modelo de análisis junto a los diagramas de interacción, estas revisiones se realizaron con el objetivo de minimizar los errores a la hora de efectuar las pruebas en la implementación del sistema.

5. Lista de chequeo para arquitectura.

   A medida que aumenta la complejidad de las aplicaciones, y que se extiende el uso de sistemas distribuidos y basados en componentes, los aspectos arquitectónicos del desarrollo de software están recibiendo un interés mayor tanto de la comunidad científica como desde la propia industria del software. En el proyecto Sistema de Gestión de Inventarios y Almacenes el eslabón principal es la arquitectura, para ello también se realizan listas de chequeo con el objetivo de que esta sea robusta y flexible.

6. Lista de chequeo para la interfaz

   Las listas de chequeo realizadas a la interfaz fue para apoyar las pruebas de caja negra que realizaron el equipo de prueba del proyecto Sistema de Gestión de Inventarios y Almacenes, y se realizaron con el objetivo de que la interfaz tuviera los requisitos mínimos e indispensables para la utilización de la misma por los clientes de la entidad, además que cumpliera con los atributos de calidad como son usabilidad, seguridad, confiabilidad, eficiencia y portabilidad.

7. Lista de chequeo para el producto.

   Esta lista de chequeo aún no se ha podido realizar debido a que el proyecto se encuentra en la primera iteración del ciclo de vida.

2.4.6 Proceso de revisión en el proyecto Sistema de Gestión de Inventarios y Almacenes.

Inicialmente para efectuar una revisión hay que tener en cuenta el plan de revisiones y auditorías. Este plan proporciona el momento exacto en que se va a efectuar cada una de las revisiones y las auditorías por parte del administrador de calidad al proyecto Sistema de Gestión de Inventarios y Almacenes.

En estos momentos el proyecto se encuentra en la primera iteración la cual esta contenida por los módulos Almacén/Inventarios y el módulo Nomencladores.

La primera revisión efectuada fue la revisión administrativa, con el objetivo de supervisar la aceptación del proyecto y la revisión al plan de iteraciones del mismo, de modo que el personal involucrado en el proyecto esté de acuerdo con las iteraciones realizadas al proyecto e incluidas en el plan de aseguramiento de la calidad del software, así como los objetivos que estas iteraciones persiguen. Consecutivamente a esto se comenzó con la revisión a los artefactos generados por el proceso de desarrollo del proyecto iniciando con el modelo de casos de uso del negocio, para realizar esta revisión se utilizó la lista de chequeo perteneciente a este artefacto, en caso de que se encontrase alguna anomalía pues se enviaba nuevamente a los analistas del sistema con el objetivo de que se vuelva a efectuar el modelado del negocio pero esta vez asegurándose de corregir todos los errores. Una vez

que se tenga el modelo del negocio en óptimas condiciones, pues se pasa a la otra fase que es la revisión a la etapa de levantamiento de requisitos, se utiliza para ello la lista de chequeo propuesta por Rational Unified Process para este artefacto (lista de requisitos), si el listado de los requisitos para la iteración no se encuentra correctamente pues las analistas del sistema tienen que realizarlo nuevamente corrigiendo lo errores registrados por el administrador de calidad en la plantilla de registros de errores (Ver anexo 3). Para completar las revisiones realizadas a los requisitos pues se hace una revisión a los casos de uso de sistema verificando que los mismos cumplan con las especificaciones hechas por los clientes, se utiliza nuevamente para esta fase la lista de chequeo (Ver anexo 3) que es la herramienta que se va a utilizar en el proceso de desarrollo para las revisiones. En caso de que en esta revisión se encuentren errores pues se pasa el registro a los analistas para que estos corrijan los errores detectados.

La siguiente revisión dedicada a las actividades de ingeniería del software es la referente a la arquitectura del proyecto. Esta revisión se realiza con el arquitecto y el administrador de calidad. Se utiliza para ello la lista de chequeo dedicada a la arquitectura del software y aplicada al proyecto. En caso de que este artefacto fundamental en el proceso de desarrollo no proporcione las exigencias para mantener una organización del software o sea que brinde la solidez necesaria al desarrollo del producto SIGIA pues se registra dicha información y se arreglan los errores cometidos. En caso de que la arquitectura se encuentre en óptimas condiciones para el soporte de todo el proceso de desarrollo, pues se pasa a la revisión del análisis y diseño del producto. En esta fase se tiene en cuenta los diagramas de interacción, así como también el modelo de casos de uso del sistema. En esta revisión se aplica la lista de chequeo referente al análisis y diseño (Ver anexo 4) y se verifica que los artefactos generados en esta etapa del proceso de desarrollo cumplan con los parámetros destinados al mejoramiento de la calidad del software. Luego de haberle aplicado la lista de chequeo de deja registro de la revisión (Ver anexo 5). Al aplicar la lista de chequeo y verificar que todo esté correctamente pues se pasaría a la revisiones de las diferentes interfaces del software con el objetivo de realizar un aporte a las pruebas de caja

negra efectuadas por el ingeniero de prueba del proyecto. Por ultimo se realiza una revisión general producto con el fin de que este cumpla con los requerimientos definidos por los clientes en la primera fase del ciclo de vida del proyecto (Ver anexo 6).

Este proceso de revisiones se realiza en cada una de las iteraciones del software con el fin de que este último salga con la calidad esperada.

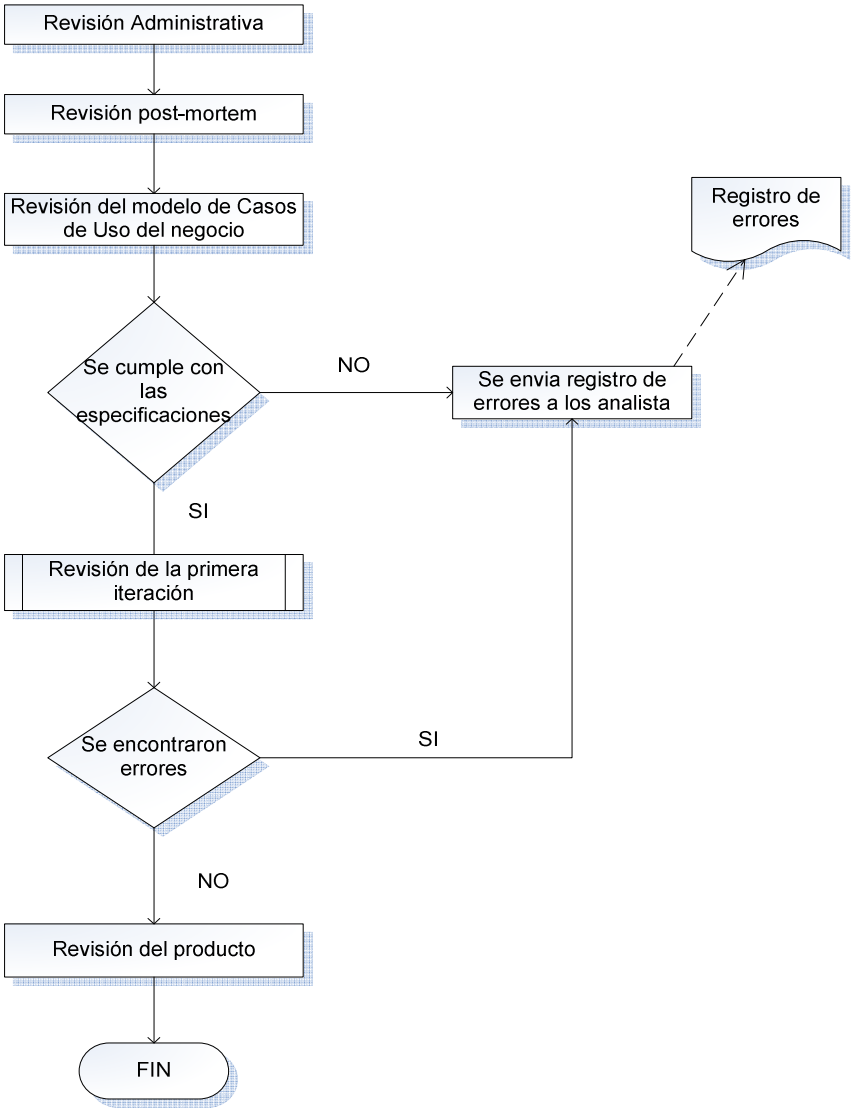


Fig. 2.4 Diagrama general de las revisiones de las iteraciones en SIGIA

2.4.7 Revisiones realizadas a la primera iteración del proyecto Sistema de Gestión de Inventarios y Almacenes.

Las revisiones realizadas al proyecto Sistema de Gestión de Inventarios y Almacenes para la primera iteración de su ciclo de desarrollo comenzaron con la revisión a los requerimientos.

La primera iteración esta compuesta por los módulos nomencladores y almacén/inventario. Mediante la lista de chequeo destinada a la etapa de levantamiento de requisitos se midieron diferentes factores de calidad, que son de suma importancia para el entendimiento del producto final. En caso de que los requisitos no cumplieran los objetivos expuestos por calidad pues se envía el registro de la revisión a los analistas y se realiza nuevamente la descripción de los requisitos, esta revisión se realiza varias veces dentro del ciclo de desarrollo del proyecto hasta que todos los requerimientos y casos de uso asociados a los mismos se encuentren en óptimas condiciones. Si por el contrario la lista de requisitos y las descripciones de los casos de uso cumplen con los parámetros de calidad expuestos pues automáticamente se pasa a la revisión de la arquitectura verificando que esta sea un soporte a la organización del proceso de desarrollo, esta revisión la realiza el administrador de calidad junto con el arquitecto del software de SIGIA. Se le aplica la lista de chequeo verificando el cumplimiento de los parámetros de calidad si la arquitectura no cumple con dichos parámetros pues el arquitecto debe revisarla nuevamente pues la arquitectura de software es la base de todo proceso de desarrollo. Al pasarle la lista de chequeo a la arquitectura y esta cumplir con todos los parámetros establecidos pues se pasa a la revisión de la fase de análisis y diseño donde se revisan todos los casos de uso, las clases de análisis y diseño respectivamente, y los diagramas de interacción si estos artefactos no cumplen las especificaciones expuestas por la lista de chequeo destinada a ellos pues los analistas del sistema para que revisen y corrijan los errores encontrados por el administrador de calidad. En caso de que los artefactos cumplan con la lista de chequeo expuesta pes se pasaría a la revisión de las interfaces. Esta revisión se realiza después de haber realizado las pruebas de caja negra realizadas por el ingeniero de pruebas, o sea son

un apoyo a las pruebas, midiendo de esta forma que cumplan con los casos de pruebas realizados y que las interfaces sean lo más amigable posible para los usuarios finales además de cumplir con los requisitos de calidad expuestos por la lista de chequeo destinada a este artefacto del proceso de desarrollo de software. En caso de que no se cumpla lo antes expuesto pues la información pasaría a los desarrolladores del software y el proceso comienza de nuevo, por otra parte si el artefacto en cuestión se encuentra en óptimas condiciones pues termina el proceso de revisiones para la primera iteración del proyecto Sistema de Gestión de Inventarios y Almacenes.

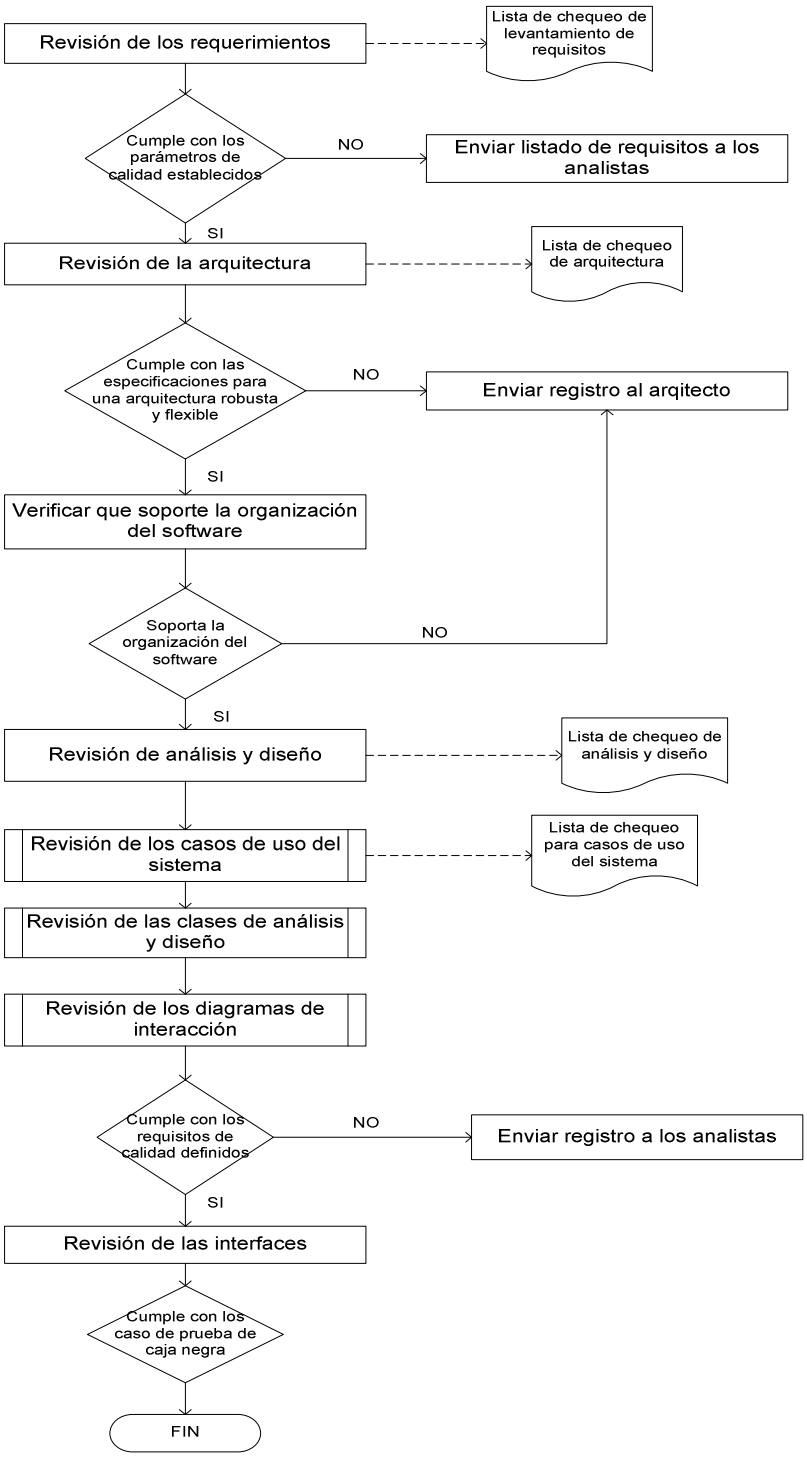


Fig. 2.5 Diagrama del proceso de revisión a la iteración 1

Modelo del negocio (Fase inicio)

El proyecto Sistema de Gestión de Inventarios y Almacenes concluyó su primera iteración (módulo nomencladores y módulo Almacén/Inventario) a la cual, a medida que se iban desarrollando cada una de las fases del proceso de desarrollo se le iban ejecutando las revisiones correspondientes.

En la fase de inicio se realizaron dos revisiones diferentes al modelo del negocio, con el fin de que todos los involucrados en el proyecto tuvieran conocimiento de los procesos y las automatizaciones que se iban a realizar en SIGIA. A esta fase se le aplicó la lista de chequeo destinada al modelo del negocio, de ella se sacaron las siguientes conclusiones:

- Los casos de uso del negocio describen el flujo de trabajo, produciendo un resultado de valor para un actor del negocio.
- Los casos de uso del negocio ejecutan únicamente las actividades que forman parte del negocio.
- Las relaciones existentes entre los casos de uso del negocio solo son relaciones de inclusión, de extensión y de generalización.
- Todos los casos de uso están asociados a un actor, en caso de que no existiera eso serían los casos de uso de extendidos, incluidos y generalizados.
- Cada actor del negocio se corresponde con un rol y no con una persona física.
- Ningún trabajador del negocio es considerado como un actor y viceversa.
- En el diagrama de casos de uso del negocio se indica quien inicia la comunicación entre actores y casos de uso.
- En los diagramas de actividad existe un y solo un estado inicial y al menos un estado final.
- De una actividad siempre se transita a otra actividad, una decisión, una barra de sincronización, y a un estado final.
- En todos los diagramas queda claro el orden en que ocurren las actividades y además queda claro cuando el orden de las actividades no es significativo.
- Todas las actividades descritas dentro del diagrama de actividad son posibles.

- El flujo de las actividades es claro y fácil de comprender.
- El los diagramas de de objeto solo se incluyen las entidades del negocio, la relación que existe entre ellas y los trabajadores del negocio que interactúan con dichas entidades.

Etapa de levantamiento de requisitos (Fase inicio)

En la etapa de levantamiento de requisitos para los módulos Nomencladores y módulo Almacén/Inventario del proyecto productivo Sistema de gestión de Inventarios y Almacenes, se llevó a cabo un proceso de revisión, en el mismo se evaluaron los atributos de calidad exactitud, facilidad de mantenimiento, y eficiencia (Ver anexo 2) y se registró en una propuesta de formato (Ver anexo 3) para tener constancia de las deficiencias sostenidas en el proceso de levantamiento de requisitos.

Cada uno de los atributos de calidad consta de diferentes parámetros los cuales son evaluados con el fin de cumplir con las exigencias realizadas por los clientes y para tener un mejor entendimiento del sistema que se está implementando.

Etapa de análisis y diseño (Fase elaboración)

En esta etapa se realizó el modelo de clases de análisis del sistema y diseño incluido el documento de la arquitectura del software. Estos artefactos fueron revisados por el grupo de aseguramiento de la calidad del software, para ello utilizaron como técnica las revisiones técnicas formales y como herramienta las listas de cheque pertenecientes a esta etapa del proceso de desarrollo del software. En el caso específico del modelo de clases de análisis y diseño se tuvo en cuenta dos atributos de calidad brindados por la lista de chequeo los cuales son la forma de medir lo realizado en el proceso. Estos atributos de calidad son, exactitud y facilidad de mantenimiento junto con los parámetros que llevan incluidos (Ver anexo 4).

Los atributos de calidad medidos en esta fase, fueron cumplidos correctamente permitiendo un modelo de clases de análisis y diseño que fueron el punto de partida para la implementación del sistema, ya que son la guía para los desarrolladores de software.

Por otra parte se puede decir que la arquitectura del sistema juega un papel importante durante todo el proceso de desarrollo del software pero fundamentalmente en la fase de elaboración y construcción, el grado de independencia y paralelismo que se encuentra en estas fases no puede ser posible si la arquitectura del software no es estable. Para ello se realiza la revisión de la misma comprobando que lo planteado anteriormente se cumpla. La importancia de una arquitectura estable no se puede exagerar, en caso de que la arquitectura no sea estable pues es mejor detener la fase de construcción para errores menores. Al aplicar la herramienta de revisión a la arquitectura se obtuvo los siguientes resultados:

1. La complejidad del sistema está acorde con la funcionalidad que proporciona.
2. La complejidad conceptual brinda las habilidades y experiencias apropiadas a los usuarios, operadores y diseñadores.
3. La arquitectura coherente
4. El nombre y los tipos de componentes son razonables.
5. Todos los componentes de seguridad trabajan para salvaguardar el sistema.
6. La arquitectura permite recuperar el sistema en caso de un fracaso dentro de tiempo requerido.
7. El diseñador del sistema puede entender la arquitectura, diseñar el elemento y desarrollarlo con éxito.
8. El empaquetamiento reduce la complejidad y mejora la comprensión.
9. El documento de la arquitectura de software está actualizado.
10. Los riesgos técnicos se han mitigado o se han enviado a un plan de contingencia, en caso de que surja un nuevo riesgo se documenta y se analiza para su impacto potencial.

Revisión de la interfaz de la primera iteración de Sistema de Gestión de Inventarios y Almacenes

En la revisión de la interfaz realizada se detectaron deficiencias a la hora de ejecutar el programa (Ver anexo 5).

2.4.8 Métricas para el proceso de revisiones del proceso de desarrollo del software.

En estudios realizados se ha demostrado que las inspecciones a software son un potente método para la detección de defectos, encuentran de un 60 a un 90% de todos los defectos, así como proveen retroalimentación que permitirá a los desarrolladores de software evitar la inyección de defectos de trabajos futuros [Schulmeyer, 1997].

Los Líderes de Proyectos que recopilen y usen efectivamente la información de las inspecciones pueden:

- Asignar recursos.
- Controlar, según los procedimientos establecidos por la proyecto.
- Determinar la calidad del software inspeccionado.
- Medir la efectividad de las Inspecciones o Revisiones.

Para mejorar el proceso de revisiones de forma tal que se eleve la calidad del producto final, es necesario tener alguna medida del mejoramiento de las Inspecciones y de su eficiencia entre otros aspectos. Para ello se han definido algunas métricas que dotarán a al proyecto de información necesaria para dar seguimiento al Proceso de Revisiones y adoptar las medidas necesarias que contribuyan a mejorarlo.

1. Efectividad de eliminar los defectos en una Inspección o Revisión:

$$\mathrm{ERD_i} = \frac{\mathrm{DE_i}}{\mathrm{DL}} * 100$$

$$\mathrm{DL} = \mathrm{DE_i} + \mathrm{DEP}$$

DEi: Cantidad de defectos detectados durante la revisión i,

DEP: Cantidad de defectos encontrados posterior a la revisión i,

DL: Cantidad total de defectos presentes en el producto.

Si se representan en un gráfico los valores resultantes de la métrica de efectividad se puede analizar qué las inspecciones realizadas al Proyecto de Software resultan poco

efectivas y así valorar la posibilidad de incluirla o no en la misma fase del Desarrollo del Proyecto en otros con características similares al analizado o tomar cualquier otra decisión que contribuya al mejoramiento del Proceso de Revisiones [Schulmeyer, 1997].

2.5 Auditorías de los productos de software designados para verificar el ajuste con los requisitos definidos como parte del proceso del software

Un buen sistema de control de calidad en un proyecto de software constituye una garantía de tranquilidad tanto para el productor, como para el cliente interesado en que producto llegue conforme a lo requerido.

Es importante que tanto el cliente como el desarrollador evalúen de manera independiente la idoneidad del servicio de control de calidad prestado para determinar su capacidad real de poder cumplir con requisitos del producto.

Las diferentes auditorías previstas en el plan de aseguramiento de la calidad del software se llevan a cabo para verificar que se hayan hecho las correcciones.

Durante el proceso de auditoría el grupo de SQA revisa las actividades de control de calidad de manera independiente observando los controles realizados en cada una de las etapas de proceso, confirmando que el personal de calidad dirija su trabajo a cumplir con las especificaciones del cliente.

2.5.1 Auditorías del control y la gestión de cambios

Cuando se construye software de computadora, los cambios son inevitables. Además, los cambios aumentan el grado de confusión entre los ingenieros de software que están trabajando en el proyecto. La confusión surge cuando no se ha analizado los cambios antes de analizarlos, no se han registrado antes de implementarlos, no se le han comunicado aquellas personas que necesitan saberlo o no se han controlado de manera que mejoren la calidad y reduzcan los errores [Pressman, 2005].

El control de cambios es vital. Pero las fuerzas que lo hacen necesario también lo hacen molesto. Nos preocupamos por le cambio porque una diminuta perturbación en el código puede crear un gran fallo en el producto. Pero también puede reparar, un gran fallo o habilitar excelentes capacidades nuevas. Nos preocupamos por el cambio porque un

desarrollador pícaro pude hacer fracasar el proyecto; sin embargo las brillantes ideas nacidas en la mente de esos pícaros, y un pesado procesos de control de cambios puede disuadirle de hacer un trabajo creativo [Bauer, 1969].

La identificación, el control de versiones y el control de cambios ayudan al equipo de desarrollo de software a mantener una orden que, de otro modo, llevaría a una situación caótica y sin salida. Sin embargo, incluso los mecanismos más correctos del control de cambios hacen un seguimiento al cambio solo hasta que se ha generado la orden de cambio de ingeniería (OCI). Para asegurarse de que el cambio ha sido implementado correctamente se tienen las Revisiones técnicas formales y auditorías de la configuración del software [Pressman, 2005].

Las revisiones técnicas formales se centran en la corrección técnica del elemento de configuración que ha sido modificado. Los revisores evalúan el ECS para determinar consistencia con otros ECSs, las omisiones o los posibles efectos secundarios. Se debe llevar a cabo una revisión para cualquier cambio que no sea trivial.

Por otro lado una auditoría de la configuración del software complementa la revisión al comprobar características que generalmente no tienen en cuenta la revisión.

2.5.2 Proceso de auditoría de solicitud de cambios en el proyecto productivo Sistema de Gestión de Inventarios y Almacenes.

El personal involucrado en el proceso de desarrollo del software del proyecto Sistema de Gestión de Inventarios y Almacenes, detecta la necesidad de un cambio en alguno de los elementos de configuración del software, el mismo realiza la petición de cambio y el desarrollador evalúa el cambio con el objetivo de calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y sobre otros elementos de la configuración. Los resultados de la evaluación por parte del desarrollador se presentan como un informe de cambio, el cual es enviado al equipo de gestión de configuración del software del proyecto, quien se encarga de aceptar o denegar el cambio. En caso de que el equipo de gestión de configuración deniegue el cambio se le informa de esto al usuario que lo solicito y termina el proceso, pero si por el contrario el equipo de

autoridad del control de cambios (ACC) acepta la petición de cambio, esta queda pendiente y se genera la orden de cambio de ingeniería (OCI). La OCI describe el cambio que se va a realizar, las restricciones que se deben respetar y los criterios de revisión y de auditoria. Al aceptar la solicitud de cambio se le deben dar de baja a los elementos de configuración que se vana cambiar y se realiza el cambio. El equipo de aseguramiento de calidad del software realiza una auditoria para la revisión del cambio, con el propósito de que se haya hecho el cambio que se especificaba en la orden de cambio de ingeniería, se haya seguido el proceso del software y se hayan aplicado correctamente los estándares de ingeniería del software, que se haya seguido el proceso de gestión de configuración para señalar el cambio, registrarlo y divulgarlo y luego se hayan actualizado correctamente todos los elementos de configuración. Luego de el equipo de aseguramiento de calidad del software realizar las tareas pertinentes para que el cambio se realice con la calidad requerida el elemento de configuración se le da de alta en la base de datos y se utilizan los mecanismo de control de versiones apropiados para crear la siguiente versión del software (Ver Fig.5)

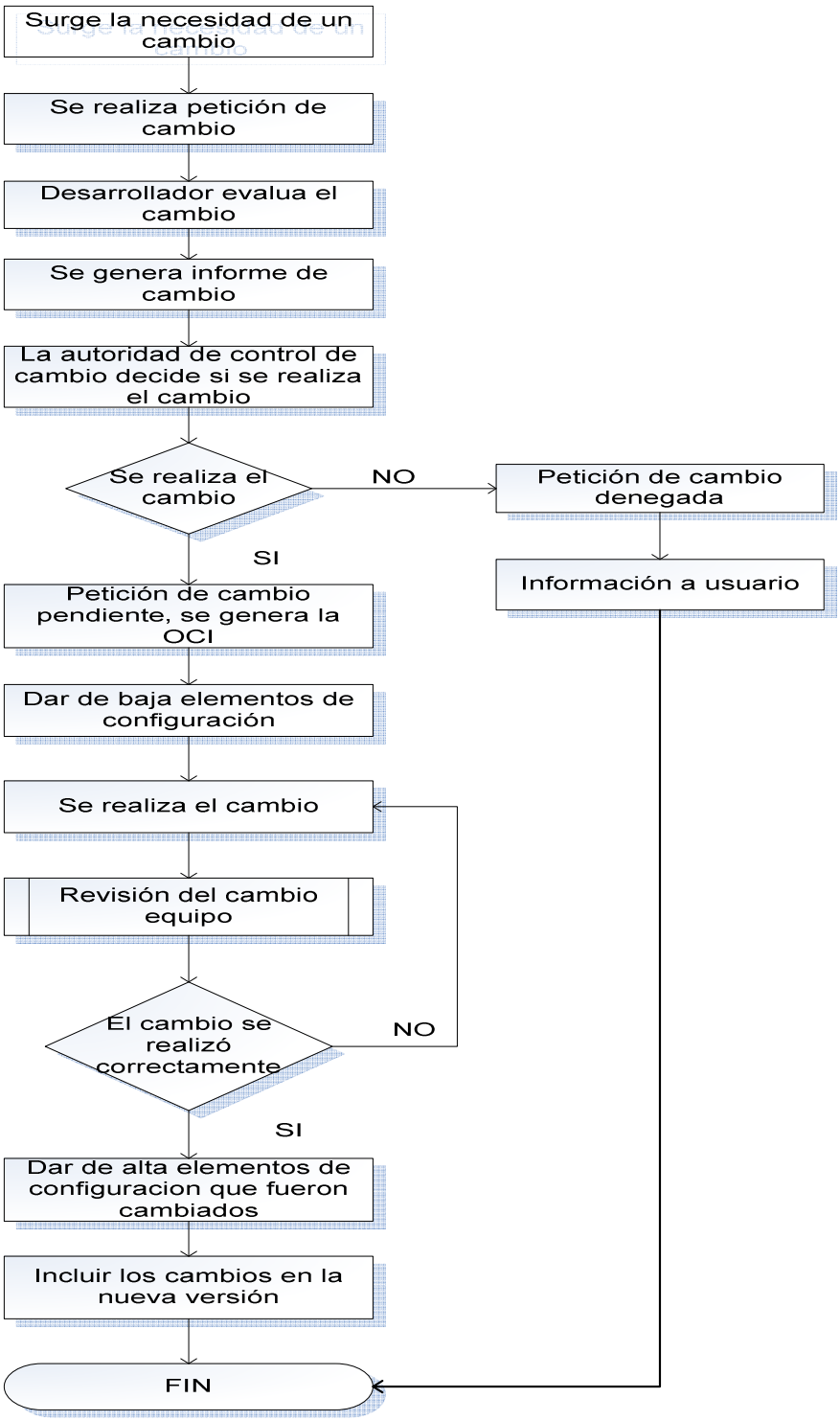


Fig. 2.6 Diagrama del proceso de Solicitud de Cambios en SIGIA

2.6 Procedimiento de revisiones para la solicitud de cambios en SIGIA

En el momento en que se realiza el cambio el equipo de SQA verifica que este cambio cumpla con el procedimiento establecido.

La primera actividad que se realiza es comprobar que el cambio que está efectuando sea el mismo expuesto en la Orden de Cambio de Ingeniería, previniendo que no existan errores mayores, en caso de que esté incorrecto el cambio se envía al equipo encargado de realizar el cambio para que este compruebe y corrija los errores. Si por el contrario el cambio coincide con el expuesto por la OCI pues se procede a confirmar que se hayan aplicado correctamente los estándares de ingeniería, si no se aplican los estándares de ingeniería pues el cambio es enviado nuevamente al equipo de desarrollo encargado de realizar el cambio. En caso de que estén aplicados correctamente los estándares de ingeniería pues se procede a ejecutar la siguiente actividad. Esta actividad es verificar que se haya llevado a cabo correctamente el proceso de gestión de configuración establecido por el equipo de gestión de configuración y aprobado por el equipo de desarrollo del software del proyecto. Al verificar que el proceso de gestión de configuración sea el correcto pues se actualizan los elementos de configuración involucrados en el cambio, dando de baja y alta a los mismos (Ver Fig. 6).

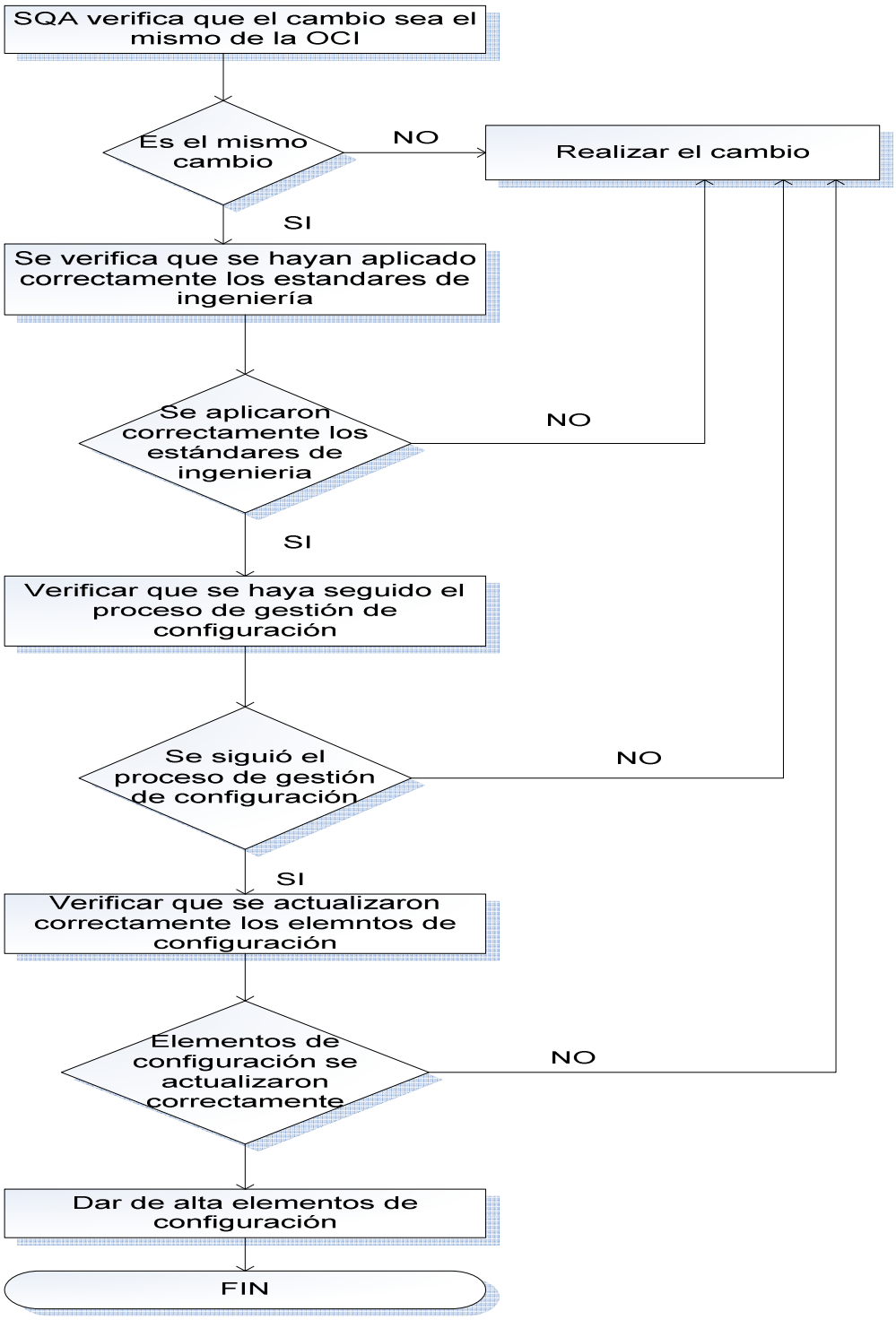


Fig. 2.7 Diagrama del proceso de revisión de la Solicitud de Cambio en SIGIA

2.7 Recopilar y analizar las métricas del software

El objetivo primordial de la ingeniería del software es producir un sistema, aplicación o producto de alta calidad. Para lograr eso objetivos los ingenieros de software deben aplicar métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo de software. Además un buen ingeniero de software debe medir si la alta calidad se va a llevar a cabo.

La calidad de un sistema, aplicación o producto, es tan bueno como los requisitos que describen el problema, el diseño que modela la situación, el código que conduce un programa ejecutable, y las pruebas que ejercitan el software para detectar errores. Un buen ingeniero de software utiliza mediciones que evalúan la calidad del análisis y modelos de diseño, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería de software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad [Pressman, 2005].

2.7.1 Métricas para levantamiento de requisitos.

Factor	Métricas Asociadas	Valor
Factores de Completitud		
Factor 1. ¿Han sido involucradas todas las áreas funcionales relevantes a las cuales apoyará el sistema?	Métrica 1: Número de áreas funcionales relevantes omitidas	3%
Factor 2. ¿Han sido involucradas todas las áreas funcionales secundarias a las cuales apoyará el sistema?	Métrica 2 : Número de áreas funcionales secundarias omitidas	5%
Factor 3. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/ modificar o consultar información?	Métrica 3: Número de roles relevantes omitidos	0%
Factor 4. ¿Han sido definidos todos los roles secundarios de usuario encargados de generar/modificar o consultar información?	Métrica 4 : Número de roles secundarios omitidos	2%
Factor 5. ¿Han sido considerados	Métrica 5: Número de	0%

todos los sistemas externos con los cuáles interactuará el sistema?	sistemas externos omitidos	
Factor 6. ¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso del negocio?	Métrica 6: Número de casos de uso que no tiene descripción resumida	0%
Factor 7. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	Métrica 7: Número de requisitos omitidos por caso de uso Métrica 8: Número de casos de uso que tienen requisitos omitidos	0%
Factor 8. ¿Existen requisitos que no han sido considerados en algún caso de uso?	Métrica 9: Número de requisitos que no son considerados en ningún caso de uso.	0%
Factor 9. ¿Han sido definidos todos los roles de usuario encargados de actividades de soporte/ mantenimiento / auditoría?	Métrica 10: Número de roles de soporte / mantenimiento / auditoría omitidos.	10%
Factor 10. ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del negocio?	Métrica 11: Número de casos de uso que no poseen una descripción extendida.	0%
Factor 11. ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?	Métrica 12: Número de acciones del flujo de eventos que no están redactadas en función del responsable. Métrica 13: Número de casos de uso que no describen condiciones de excepción relevantes	0%
Factor 12. ¿Se describen las condiciones de excepción relevantes que debe contemplar cada flujo de eventos?	Métrica 14: Número de casos de uso que tienen un nombre incorrecto	0%
Factor 13. ¿Todos los casos de uso del negocio han sido clasificados de acuerdo a su relevancia (primario / secundario / opcional)?	Métrica 15: Número de casos de uso que no representan una interacción observable por un actor	0%

		96%
Factores de Consistencia		
Factor 14. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	Métrica16: Número de casos de uso que se solapan	0%
Factor 15. ¿Representa el caso de uso una interacción observable por un actor?	Métrica17: Número de acciones del flujo de eventos que no se corresponde la definición de las con el responsable	0%
Factor 16. ¿No existe solapamiento en la funcionalidad que representan los diferentes casos de uso?	Métrica 18: Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.	0%
Factor 17. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?	Métrica 19: Grado de adecuación de la descripción del flujo de eventos para un caso de uso Métrica 20: Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.	0%
Factor 18. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?	Métrica 21: Grado de adecuación de la descripción del flujo de eventos para un caso de uso Métrica 22: Número de casos de uso no aceptados	0%
Factor 19. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema	Métrica 23: Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una	0%

claramente identificable?	condición monitoreada por el sistema	
Factor 20. Si en el caso de uso interviene mas de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?	Métrica 24: Número de casos de uso con más de un actor que no describe cuál es el actor iniciador	0%
Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 25: Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos	0%
		100%
Factores de Correctitud		
Factor 22. ¿Existe para cada caso de uso de negocio por lo menos un usuario responsable?	Métrica 26: Número de casos de uso que no tienen un usuario responsable	0%
Factor 23. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 27: Grado en que los requisitos representados por el caso de uso son comprensibles por el usuario Métrica 28: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario	0%
Factor 24. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?	Métrica 29: Grado en que se ajusta el diagrama del caso de uso a la metodología.	0%
Factor 25. ¿Las interacciones definidas describen la funcionalidad requerida del sistema?	Métrica 30: Grado en que las interacciones definidas describen la funcionalidad solicitada por el usuario Métrica 31: Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema	0%

Factor 26. ¿Las interacciones definidas introducen mejoras al proceso actual?	Métrica 32: Número de casos de uso que deben ser modificados para mejorar el proceso actual	5%
Factor 27. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?	Métrica 33: Grado en que se ajusta el diagrama del caso de uso a la metodología.	0%
		99.16%

Factores de Complejidad		
Factor 28. ¿En sistemas relativamente grandes se ha realizado una agrupación de los casos de uso en paquetes?	Métrica 34: Se hizo partición por paquetes Métrica 35: Grado en que es adecuada la partición por paquetes	0%
Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	Métrica 36: Número de elementos del diagrama que requieren reubicación	3%
		98.7%

Tabla 2.1 Métricas para los requerimientos

2.7.2 Métricas para modelo de análisis

En el proyecto productivo Sistema de Gestión de Inventarios y Almacenes se definió la métrica del punto de función para el modelo de análisis.

Los puntos de función se derivan con una relación empírica según las medidas contable (directas) del dominio de información del software y las evaluaciones de la complejidad del software.

Se utilizó esta métrica debido a que está diseñada para aplicaciones de sistemas de información de gestión aunque se utiliza también para otro tipo de sistemas, además de que es muy fácil de realizar.

Esta métrica se calcula completando la tabla 2.2.

Los valores de dominio de información se definen de la siguiente forma:

- Número de entrada de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada [Pressman, 2005].
- Número de salidas de usuario. Recuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto las salidas se refiere a informes, mensajes de error, etc.
- Número de archivos. Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente.)
- Numero de interfaces externas. Se cuenta todas las interfaces legibles por la máquina.

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad, este valor de complejidad es subjetivo [Pressman, 2005].

Parámetro de medición		Cuenta	Simple	Medio	Complejo		
Nro entradas	X		3	4	6	=	
Nro salidas	X		4	5	7	=	
Nro de peticiones	X		3	4	6	=	
Nro archivos	X		7	10	15	=	
Nro interfaces externas	X		5	7	10	=	
Cuenta total							

Tabla 2.2 Cálculo de punto de función de la métrica del modelo de análisis

Para calcular puntos de función (PF), se utiliza la relación siguiente:

PF= cuenta-total X [0.65+0.01x 6(Fi)]

En donde cuenta-total es la suma de todas las entradas obtenidas de la figura.

Fi (i= 14) son <<valores de ajuste de complejidad según la respuesta de las siguientes preguntas:

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Es crítico el rendimiento?
4. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?
5. ¿Requiere el sistema entrada de datos interactivas?
6. ¿Requiere la entrada de datos interactivos que las transacciones de entradas se lleven a cabo sobre múltiples pantallas u operaciones?
7. ¿Se actualiza los archivos maestros de una forma interactiva?
8. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
9. ¿Es complejo el procesamiento interno?
10. ¿Se ha diseño el código para ser utilizable?
11. ¿Están incluidas en el diseño la conversión y la instalación?
12. ¿Se ha diseñado el sistema para múltiples instalaciones en diferentes organizaciones?
13. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

2.7.3 Métricas para el diseño

En el proyecto se decidió utilizar la métrica para el diseño de interfaz.

Para una representación específica (por ejemplo, el diseño de una IGU interfaz gráfica de usuario específica), se pueden asignar costos a cada frecuencia de acciones de acuerdo con la siguiente relación:

Costos= Σ [frecuencia de transición (k) X costos de transición (k)].

Donde k es la transición específica de una entidad de representación a la siguiente cuando se realiza una tarea específica. Esta suma se da con todas las transiciones de una tarea en particular o conjunto de tareas requeridas para conseguir alguna función de la aplicación. El costo puede estar caracterizado en términos de tiempo, retraso del proceso o cualquier otro

valor razonable, tal como la distancia que debe moverse el ratón entre entidades de la representación [Pressman, 2005].

La conveniencia de la representación se define como:

CR = 100 X [(Costos de la representación óptima CR)/ (costos de la representación propuesta)]

Donde CR =100 para una representación óptima [Pressman, 2005].

La CR se emplea para valorar diferentes distribuciones propuestas de IGU y la sensibilidad de una IGU representación en particular los cambios en las descripciones de tareas. El diseñador de interfaz puede emplear el cambio en la conveniencia de la representación como una guía en la elección de la mejor representación de IGU para una aplicación en particular [Pressman, 2005].

2.7.4 Métricas para el código fuente

La ciencia del software asigna leyes cuantitativas al desarrollo del software de computadora, usando un conjunto de medidas primitivas que pueden obtenerse una vez que se ha generado o estimado el código después de completar el diseño. Estas medidas se listan a continuación:

N1: el número de operadores diferentes que aparecen en el programa

N2: el número de operando diferentes que aparecen en el programa

N3: el número total de veces que aparece el operador

N4: el número total de veces que aparece el operando

Halstead usa las medidas primitivas para desarrollar expresiones para la longitud global del programa; volumen mínimo potencial para un algoritmo; el volumen real (número de bits requeridos para especificar un programa); el nivel del programa (una constante para un lenguaje dado); y otras características tales como esfuerzo de desarrollo, tiempo de desarrollo e incluso el número esperado de fallos en el software [Pressman, 2005].

Halstead expone que la longitud N se puede estimar como:

N= n1log2n1+n2log2n2

Y el volumen de programa se puede definir como:

V= N log2 (n1+n2)

Se tiene en cuenta que V varía con el lenguaje de programación y representa el volumen de información en (bits) necesarios para especificar un programa. Teóricamente debe existir un volumen mínimo para un algoritmo. Halstead define una relación de volumen de la forma más compacta de un programa con respecto al volumen real de un programa. Por tanto, L debe ser siempre menor de 1. En términos de medidas primitivas, la relación de volumen se puede expresar como:

L=2/n1 X n2/N2

2.7.4 Métricas para pruebas

Las métricas basadas en la función pueden emplearse para predecir el esfuerzo global de las pruebas. Se pueden juntar y correlacionar varias características a nivel de proyecto (por ejemplo, esfuerzo y tiempo para las pruebas, errores descubiertos, número de casos de prueba producidos).

La métrica bang puede proporcionar una indicación del número de casos de pruebas necesarios para examinar las medidas primitivas. El número de primitivas funcionales (PFu), elementos de datos (DE), objetos (OB), relaciones (RE), estados (ES), y transiciones (TR) pueden emplearse para predecir el número y tipo de prueba del software de caja negra y de caja blanca [Pressman, 2005]

2.7.5 Métricas del mantenimiento

El estándar IEEE 682.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software (basado en cambios que ocurren con cada versión del producto). Se determina la siguiente información:

Mt= número de módulos en la versión actual

Fc= número de módulos en la versión actual que se han cambiado

Fa= número de módulos en la versión actual que se han añadido

Fd= número de módulos en la versión anterior que se han borrado en la versión actual.

El índice de la madurez del software se calcula de la siguiente manera

IMS= [Mt-(Fa+Fc+Fd)]/Mt

A medida que el IMS se aproxima a 1.0 el producto se empieza a estabilizar. El IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software. El tiempo medio para producir una versión de una producto de software puede correlacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento.

2.8 Valoración final de la solución propuesta

La solución que este Trabajo de Diploma propone para el diseño del proceso de Aseguramiento de la Calidad del Software para el Sistema de gestión de Inventarios y Almacenes comienza a partir de la confección del Plan de Aseguramiento de la Calidad del Software el cual está aprobado por el líder del proyecto y el jefe del proyecto SIGIA, este plan es el encargado de la planificación y seguimiento de la calidad del software en SIGIA. Este documento está contenido principalmente por el plan de revisiones y auditorías, el cual constituye un apoyo al documento principal de este Trabajo de Diploma, así como también con la definición de las métricas que se van a desarrollar para la medición del software, la coordinación y el control de los cambios que se presenten durante el proceso de desarrollo del software.

El plan de revisiones y auditorías construido en el proyecto Sistema de Gestión de Inventarios y Almacenes y contenido en el plan de Aseguramiento de la Calidad del Software presenta diferentes tipos de revisiones efectuadas tales como revisión a los requerimientos, revisión al análisis y diseño, revisión a la arquitectura, auditoría a la configuración funcional, etc.. . Estas revisiones fueron realizadas al proyecto y se encuentran bien definidas y explicadas en este trabajo. Se llevó a cabo las revisiones y estas últimas se realizaron con el fin de darle seguimiento a las actividades de ingeniería del software, y que el proceso de aseguramiento de la calidad cumpliera su principal objetivo que no es más que el cumplimiento con los requisitos del cliente.

Para la planificación del proceso de aseguramiento de la calidad del software se tuvo en cuenta el plan de aseguramiento de la calidad que propone RUP (Proceso Unificado de Software), en el se deja plasmado todo lo que se va a realizar para el proceso de calidad que se va a llevar a cabo en el proyecto. La planificación expuesta en este Trabajo de

Diploma y aprobada por el jefe de proyecto, el líder del proyecto y los diferentes desarrolladores del mismo, tiene gran importancia para el proceso de aseguramiento de la calidad ya que representa una guía para las actividades de ingeniería del software que se realizan en SIGIA. Los puntos que se consideran que son los más importantes dentro del plan de aseguramiento de la calidad son: plan de revisiones y auditorías, métricas del software, y auditoría al proceso de solicitud de cambios.

Para dar comienzo al primer punto del plan de aseguramiento de la calidad, que no es más que el plan de revisiones y auditorías se puede decir que este último cuenta con diferentes revisiones y se realizan con el fin de desarrollar un proceso de aseguramiento de la calidad con la calidad requerida. Estas revisiones comienzan con la Revisión Administrativa la cual está destinada a la revisión del plan de iteraciones del proyecto, así como también a la aprobación del mismo, comprobando que las iteraciones sean las más efectivas y que todos los involucrados en el proceso de desarrollo estén de acuerdo con ellas. Consecutivamente a esto se realiza la revisión a la iteración en la que se encuentra en el proyecto, en este caso a la primera iteración ya que el proyecto se encuentra actualmente en la misma. En esta iteración se revisa la etapa de levantamiento de requisitos, análisis y diseño, codificación, la interfaz, la solicitud de cambios en caso de que existan, arquitectura y por último se realiza una revisión al producto resultante del proceso de desarrollo para esa iteración. En cada una de las revisiones se deja un registro de las mismas con el objetivo de no cometer los errores nuevamente. Luego de haber realizado las revisiones y haber comprobado que todo estuviera correcto pues se pasaría a la siguiente iteración y se lleva a cabo el mismo proceso. Estas revisiones son de gran importancia para el proceso desarrollo de software, son un filtro para el software procurando que el producto final salga con la calidad esperada, además ayuda a los desarrolladores a educarse ya que ellos no volverán a cometer esos errores nuevamente, además de que reducen los costos.

Como otro de los puntos fundamentales tratado en el plan de aseguramiento de la calidad se tiene la definición de métricas de software para el proyecto, estas métricas se utilizan para valorar la calidad de los productos de ingeniería. En este Trabajo de Diploma

simplemente se definen las métricas que se van a utilizar, las que se creen que son más importantes. Se debe realizar un estudio más exhaustivo acerca de este tema, ya que al existir métricas para cada momento de la ingeniería del software no se pudo tener en cuenta muchas métricas que también son relevantes para el proceso de desarrollo.

Como un punto esencial en el proceso de desarrollo de software se tiene la revisión y auditoría a proceso de gestión de configuración. El proceso evaluado por el equipo de aseguramiento de la calidad del software fue solicitud de cambio, este proceso es muy importante ya que los cambios realizados en el proyecto son la base para la obtención del producto. Finalmente con un proceso de solicitud de cambios bien efectuado y cumpliendo con los requisitos propuestos por gestión de configuración y la revisión correspondiente efectuada al mismo, se puede decir que la identificación, y el control de cambios ayudan al equipo de desarrollo de software a mantener una orden que, de otro modo, llevaría a una situación caótica y sin salida.

Uno de los aspectos más importante en el proceso de calidad efectuado es la utilización de herramientas para que este proceso sea efectivo. Las herramientas utilizadas fueron las listas de chequeo, las mismas son de vital importancia para el proceso de desarrollo del software.

EL proceso de aseguramiento de la calidad del software e el proyecto Sistema de Gestión de Inventarios y Almacenes no cuenta aún con valiosos resultados debido a que el proyecto se encuentra en la primera iteración de su ciclo de desarrollo. Con las revisiones y auditorías desarrolladas en SIGIA se puede llegar a la conclusión de que si se aplica correctamente el proceso de Aseguramiento de la Calidad del Software en el proyecto Sistema de Gestión de Inventarios y Almacenes, teniendo en cuenta todo lo planteado en el Plan de Aseguramiento de Calidad del Software el producto final obtenido poseerá la calidad esperada por parte de los clientes dando así cumplimiento a los requisitos presentados por los mismos.

Conclusiones del capítulo II

La solución propuesta en este capítulo está contenida por la realización de un documento, cual es obligatorio su refinamiento y seguimiento por parte del Administrador de Calidad.

También se realizan las revisiones previstas por el plan de revisiones y auditorías realizadas al proyecto Sistema de gestión de Inventarios y Almacenes, estas revisiones se desarrollaron a la primera iteración del ciclo de desarrollo y los resultados de estas revisiones y auditorías se encuentran registrados para darle cumplimiento y seguimiento a las actividades de ingeniería del software.

En este capítulo se realiza la definición de las métricas del software, las mismas son de gran importancia para la medición y análisis de SIGIA.

Se realiza el proceso de revisión para el control y la coordinación de los cambios ocurridos en el proyecto con el fin de que el proceso de desarrollo del software en el proyecto tenga un carácter uniforme.

CONCLUSIONES GENERALES

- El proceso de Aseguramiento de la calidad del software representa un factor importante dentro del proceso de desarrollo de un proyecto asegurando que este último sea favorecido con el cumplimiento de los requisitos del mayor beneficiado (el cliente).

- El estudio y desarrollo detallado del principal aspecto del proceso de aseguramiento de la calidad del software (planificación), es la causa trascendental para que el producto final sea el esperado tanto por el cliente como por los mismos desarrolladores del software.

- Las principales tareas llevadas a cabo por el administrador de calidad para el proceso de aseguramiento de calidad del software constituyeron una vía para el seguimiento de errores dentro del proceso de desarrollo.

- El Plan de Aseguramiento de la Calidad del Software vinculó todos los movimientos que se efectuaron el Sistema de Gestión de Inventarios y Almacenes para los cuales se tenía un único fin (cumplimiento con los requisitos de cliente).

- Uno de los puntos más importantes dentro del plan de aseguramiento de la calidad del software es el plan de revisiones y auditoría que señala el momento exacto en que se van a realizar las mismas. Este plan es de suma importancia ya que sin él no se lograrían los objetivos del proceso de aseguramiento de la calidad del software.

- Para las revisiones llevadas a cabo en SIGIA se definió como técnica las Revisiones Técnicas Formales y como herramienta se utilizaron las listas de chequeo. Estas últimas miden una serie de atributos de calidad que están enfocadas al mejoramiento del producto de software, y el principal beneficiado no sería otro que el cliente.

- Las métricas permiten establecer una medición a las actividades de ingeniería del software para ver hasta que punto estas actividades fueron cumplidas por el proyecto Sistema de Gestión de Inventarios y Almacenes.

- El proceso propuesto de revisión de cambios asegura que todos los parámetros establecidos por el proceso de gestión de configuración en SIGIA para la coordinación y el control de los cambios sean cumplidos de manera eficiente, de forma tal que las medidas y decisiones tomadas sean las correctas para la obtención de un producto seguro.

RECOMENDACIONES

- Dar seguimiento a las actividades del plan de aseguramiento de la calidad, durante todo el ciclo de vida del proyecto.
- Cumplir al plan de revisiones y auditorías durante todo el ciclo de desarrollo.
- Analizar y desarrollar las métricas definidas en este trabajo de diploma, así como también analizar y estudiar otras métricas que sean de mejor entendimiento.
- Realizar las revisiones al proceso de solicitud de cambios con el fin de controlar y coordinar lo que ocurre el proyecto.

REFERENCIAS BIBLIOGRAFICA

[Baeza, 1995] Baeza Yates, R. A. y otros. Computing in Chile: The Jaguar of the Pacific Rim Communications of the ACM. Vol. 38, No 9, 1995.

[Buades, 2002] Buades Rubio Gabriel, "Calidad en ingeniería del Software", 2002.

[Bauer, 1969] Bauer. Dr. F. L *"Software Quality". Report on a conference sponsored by the NATO SCIENCE COMMITTEE Garmisch, Germany, 7th to 11th October 1968. January 1969.*

[Cockburn, 2000] Cockburn, Alistair. "*Software Quality Process*", Cockburn*Highsmith Series Editors, 2000.

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

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

[Delgado, 2006] Delgado, Martha Dunia, Las revisiones dentro del proceso de aseguramiento de la calidad del software, CEIS, 2006.

[Durán, 2000] 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.*

[Febles, 2000]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, 2002] Febles Estrada Ailyn,"El proceso de enseñanza aprendizaje y la vinculación universidad-empresa, una experiencia en una Universidad Cubana", Instituto Superior Politécnico "José Antonio Echeverría", 2002.

[Febles, 2007] 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, 2005] Ferré Grau, Xavier; *Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software*. Tesis Doctoral. 2005.

[Humphrey, 2002] Humphrey Watts S, “Introducción al Proceso de Software Personal (PSP)”. Addison-Wesley. 2002.

[IEEE, 1990] IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12–1990, Institute of Electrical and Electronics Engineers, 1990.

[IEEE, 2004] IEEE Standard Glossary of Software Engineering Terminology, Guide to the Software Engineering Body of Knowledge, Los Alamitos, California, 2004

[ISO 8402, 1994] sitio web http://fabetsia.dmpa.upm.es/solo_alumnos/sp2/Tablon_sp2/TransparenciasCALIDAD06.pdf (1/02/2007)

[ISO 9000, 2000] sitio web http://fabetsia.dmpa.upm.es/solo_alumnos/sp2/Tablon_sp2/TransparenciasCALIDAD06.pdf

[Jackson, 2001] Jackson, M.: Software Requirements and Specifications, Addison-Wesley, 2001.

[Juran, 1979] sitio web http://www.juran.com/ 13/2/2007

[Nogueira, 2004] Nogueira Rivera Dianelys, “Control de gestión: Evolución, dimensiones y diagnostico”, 2004

[Pressman, 2005] Pressman Roger S. “Ingeniería del Software, Un enfoque práctico”, Félix Varela, La habana, 2005.

[Reuse, 2006] The Reuse Company. “Presente y Futuro de la Reutilización de Software”. Última Modificación: 20/06/2006.

[Reynolds, 1995] Reynolds, G.: “Information Systems for Managers”, West Publishing Company, 3ra Ed, 1995.

[RUP, 2003] Rational Unified Process®. Versión 2003.06.00.

[Schulmeyer, 1997] Schulmeyer, G.: “Handbook of Software Quality Assurance”, 1997

[Wieringa, 1996] Wieringa, R. J. “Requirements Engineering: Frameworks for Understanding” JohnWiley & Sons, 1996.

GLOSARIO DE TERMINOS

SIGIA Sistema de Gestión de Inventarios y Almacenes

SQA Software Quality Assurance (Aseguramiento de la calidad del software)

OCI Orden de Cambio de Ingeniería

ECS Elementos de Configuración del Software

ACC Autoridad del control de cambios