



Universidad de las Ciencias Informáticas

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

**Título: Definición de la estrategia para el aseguramiento
de la calidad en el proyecto Sistemas de Información
Geográfica para Móviles**

Autor: Yasmani Chapman Despaigne

Tutora: Ing. Yanet Edghill Martínez

Co-Tutora: Zayli Rodríguez

Ciudad de La Habana, 2011.

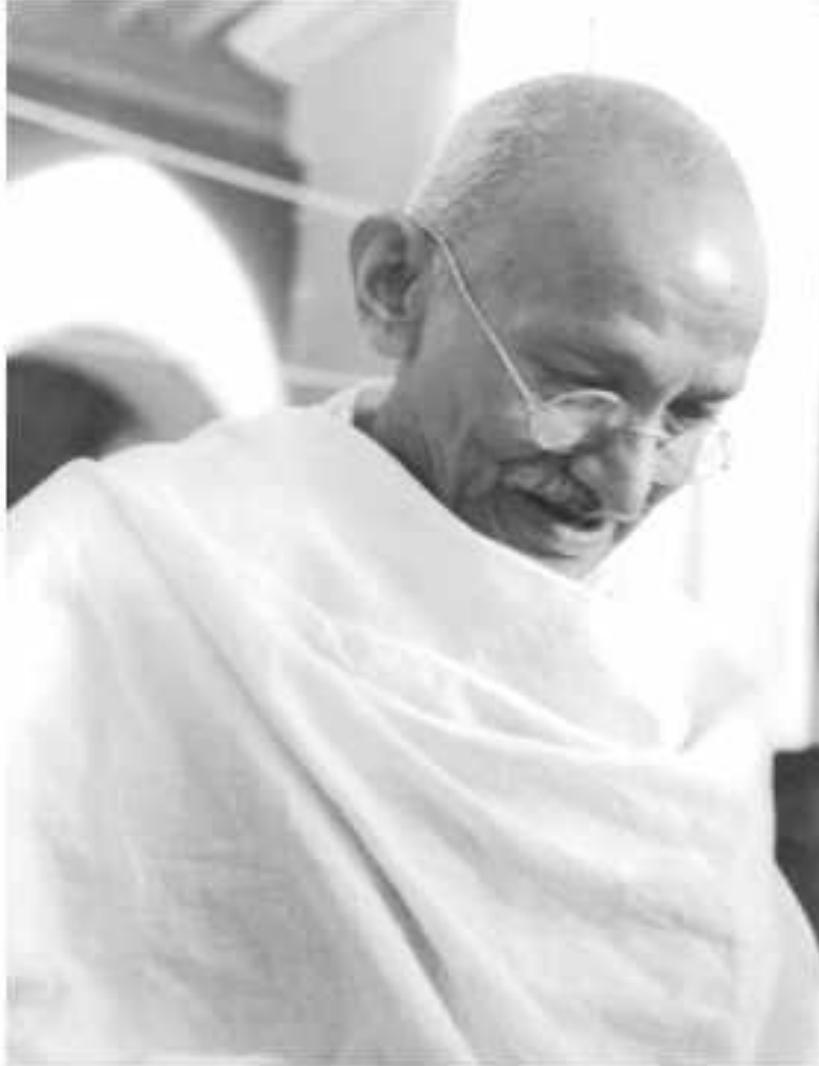
“Año 53 de la Revolución

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

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

“Yasmani Chapman Despaigne”

“Yanet Edghill Martínez”



“La voz interior me dice que siga combatiendo contra el mundo entero, aunque me encuentre solo. Me dice que no tema a este mundo sino que avance llevando en mí nada más que el temor a Dios.”

Mahatma Gandhi

DATOS DE CONTACTO

Tutor: Ing. Yanet Edghill Martínez.

Breve Currículum:

Ing. Yanet Edghill Martínez: Graduado de ingeniera en ciencias informáticas en el 2008.

El mismo tiene la categoría docente de instructor y cuenta con 3 años de experiencia en proyectos productivos.

Correo Electrónico: yedghill@uci.cu

Años de graduado: 2

Años de experiencia en el tema: 3

Co-Tutor: Zayli Rodríguez

Breve Currículo:

Correo Electrónico: zluis@uci.cu

Años de graduado:

Años de experiencia en el tema:

AGRADECIMIENTOS

Quisiera de forma general agradecerles a tantas personas que en verdad no sé por dónde empezar. Aspiro encontrar la palabra común, aquella que lleve a todos mi más sincera gratitud por todo su amor y su comprensión. “*Gracias*”, podría decirles a las personas que me subestimaron pues me dieron la fuerza para levantarme cada mañana y hacer de mi vida un reto hacia lo que resultó ser lo mejor de mí. A los que me brindaron la mano esperando vuelto a cambio, pues me mostraron que la diferencia de las personas no está en el color de piel sino en la sinceridad del accionar. Pero a ustedes amigos míos como decirles, si es que diciendo diría poco, con ustedes va todo mi sentir y el orgullo de tenerlos formando parte de mi universo.

Antes que todo agradecer a Dios por ayudarme alentarme y mostrarme el camino. A mi mamá todo el aprecio, el orgullo y el cariño, por mostrarme el camino, por enseñarme siempre a dar sin esperar nada a cambio, por enseñarme hacer siempre el bien, por convertirme en la persona que realmente deseo ser. A mi papá mi respeto, el cariño y la admiración por enseñarme lo que realmente significa la familia y las responsabilidades que esta implica, por estar siempre a mi lado a pesar de los obstáculos, por orientarme, por comprenderme y por tomar en cuenta mis decisiones y hacer que me sienta útil dentro de la misma.

A mi novia Yamaisi Reyes García todo mi amor por estar siempre a mi lado comprenderme y alentarme. A mi hermano Yusmani Chapman por estar siempre a mi lado y seguirme siempre, por escuchar lo que digo y saber tomar siempre lo positivo de cada situación. A mi tutora Yanet Edghill por brindarme su apoyo cuando lo necesité y por orientarme a lo largo de todo este proceso. A mis amigos Carlos Edgar Lorie, Carlos Merconchine, Yadir Marrero, Sergio Barrera, Franier González, Yanier Martínez, Lillian Menéndez, Lillian Landestoy por ayudarme, respetarme y por hacer más agradable mi estancia en la universidad

DEDICATORIA

Le dedico este trabajo, a esta Revolución que me ha dado la oportunidad de ser una mejor persona, de convertirme en un profesional capacitado, de tener fe en quién soy y la voluntad para ser quien quiera ser en el futuro. Este trabajo representa 5 años de mi vida, 5 años de abnegación, de esfuerzo, 5 años de sacrificio, por eso va dedicado especialmente a las personas que depositaron toda su fe en mi y confiaron plenamente en mi capacidad e intelecto.

A mi papá Ramón Reinaldo Chapman Céspedes para el que siempre he querido ser un orgullo, a mi mamá Rosaura Despaigne Despaigne por mostrarme el valor y la belleza de las pequeñas cosas, a mi hermano Yusmani Chapman Despaigne para el que siempre he querido ser un ejemplo, a mi tata Margarita Despaigne San Miguel por formar parte de mi pasado, presente y por darme la educación y el cariño cuando más lo necesité y a mi novia Yamaisi Reyes García por estar siempre a mi lado, por quererme, apoyarme y ayudarme en todo cuanto pudo.

RESUMEN

Debido a la necesidad de mejorar la calidad de los productos desarrollados por el proyecto "SIG-MÓVILES" ubicado en la facultad 6 de la Universidad de las Ciencias Informáticas (UCI), surgió el presente trabajo de diploma teniendo por título: "Definición de la estrategia de aseguramiento de la calidad para el proyecto "SIG-MÓVILES". El mismo tiene como propósito la definición de una estrategia para el aseguramiento de la calidad de los productos desarrollados en SIG-MOVILES. De igual manera se logrará optimizar y mejorar el trabajo durante el ciclo de vida del proyecto.

La estrategia propuesta sustenta el Plan de Aseguramiento de la Calidad en el cual se fundamentan todo el procedimiento concebido para la realización de la misma. Para el desarrollo de este trabajo se llevó a cabo la investigación y caracterización de modelos, técnicas y normas relacionadas con el proceso de aseguramiento de la Calidad.

PALABRAS CLAVES

Aseguramiento de la calidad de procesos y productos, calidad, técnicas, normas.

Contenido

INTRODUCCIÓN	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA ESTRATEGIA PARA EL ASEGURAMIENTO DE LA CALIDAD	15
INTRODUCCIÓN	15
1.1 CALIDAD DE SOFTWARE.....	15
1.2 GESTIÓN DE LA CALIDAD	16
1.2.1 Planificación de la Calidad	16
1.2.2 Control de la Calidad	17
1.2.3 Aseguramiento de la Calidad	20
1.2.4 Mejora de la Calidad.....	21
1.3 AUDITORÍAS.....	21
1.4 ESTÁNDARES	22
1.5 MODELOS	24
1.5.1 CMMI	24
1.5.2 SPICE	26
1.6 REVISIONES.....	27
1.7 MÉTRICAS DEL SOFTWARE	29
1.8 ESTADO ACTUAL DE LA CALIDAD DE SOFTWARE	30
1.8.1 Calidad de software en el Mundo	30

1.8.2 Calidad de software en Cuba	30
1.8.3 Calidad de software en la UCI	31
CONCLUSIONES DEL CAPÍTULO.....	31
CAPÍTULO 2: PROPUESTA DE LA ESTRATEGIA PARA EL ASEGURAMIENTO DE LA CALIDAD.....	32
INTRODUCCIÓN	32
2.1 ESTADO ACTUAL DEL PROYECTO SIG-MÓVILES.....	32
2.2 ESTRUCTURA Y COMPOSICIÓN.....	32
2.2.1 Roles y responsabilidades definidos	33
2.2.2 Soporte tecnológico del proyecto	33
2.3 ESTRATEGIA DEL ASEGURAMIENTO DE LA CALIDAD.....	33
2.3.1 Estructura de la Estrategia	34
3. CONTROL	41
CONCLUSIONES DEL CAPÍTULO.....	44
CAPÍTULO 3: VALIDACIÓN DE LA ESTRATEGIA	45
INTRODUCCIÓN	45
3.1. RESULTADOS OBTENIDOS DE LA APLICACIÓN DE LA PROPUESTA.....	45
3.1.1. Definición del Equipo de Calidad.....	45
3.1.2. Entrenamiento del equipo de desarrollo	46
3.1.3. Resultado de la aplicación de las revisiones.....	48
CONCLUSIONES DEL CAPÍTULO.....	51
CONCLUSIONES GENERALES.....	52

RECOMENDACIONES.....	53
BIBLIOGRAFÍA.....	54
TRABAJOS CITADOS.....	56
ANEXOS.....	58
GLOSARIO DE TÉRMINOS.....	63

INTRODUCCIÓN

Actualmente, debido al desarrollo creciente que presentan las Tecnologías de la Información y las Comunicaciones TICs, el mundo se ha visto inmerso en un proceso de transformaciones afectando prácticamente todas las esferas de la sociedad. Estas tecnologías constituyen cada vez más una necesidad en el ámbito social donde los cambios, el aumento del conocimiento y las demandas de una educación de alto nivel se convierten en una exigencia permanente.

El avance tecnológico es un componente que está insolublemente ligado al desarrollo de cualquier país del mundo. La presencia de la computación como elemento indispensable dentro de este progreso, juega un papel determinante en los sectores productivos. Esto se debe a que los costos, la duración de los procesos son superiores a los estimados y la satisfacción de las expectativas de los clientes todavía no alcanza los niveles de conformidad requeridos.

Numerosa es la cifra de naciones que han destinado considerables esfuerzos a la capacitación del personal calificado en el desarrollo de software, con el fin de desarrollar productos que puedan ser de utilidad para el progreso del país. Actualmente uno de los principales problemas existentes en este ámbito es la calidad del software. Este tema ha sido objeto de estudio para un gran número de ingenieros y especialistas en el desarrollo del software a partir del cual se han llevado a cabo cuantiosas investigaciones.

Según la Lic. Fernanda Scalone (12) en su tesis para optar por el título de máster, destaca que la calidad del producto de software se diferencia de la calidad de otros productos de fabricación industrial, debido a que:

- ✓ El software es un producto mental, no restringido por las leyes de la física o por los límites de los procesos de fabricación. Es algo abstracto, y su calidad también lo es.
- ✓ Se desarrolla, no se fabrica.
- ✓ El software no se deteriora con el tiempo. No es susceptible a los efectos del entorno, y su curva de fallos es muy diferente a la del hardware. Todos los problemas que surjan durante el mantenimiento estaban desde el principio, y afectan a todas las copias del mismo; no se generan nuevos errores.

Su trascendencia es tal, que pasó de ser una ventaja competitiva a una condición necesaria que debe cumplir cada organización que desarrolle software para sobrevivir en el mercado. A medida que se desarrollan las TICs los problemas van siendo más complejos.

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. La UCI y el sistema de empresas cubanas vinculadas a este trabajo jugarán un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización (1).

La meta de transformar la informática en un campo próspero internacionalmente y en una de las ramas más productivas de Cuba, requiere inevitablemente el respaldo de un sólido sistema de educación superior. Corresponde a las universidades cubanas, garantizar la formación de profesionales en la informática y la computación, listos para enfrentar la producción de software con responsabilidad y creatividad. La UCI debe alcanzar este propósito con la ejecución de ambiciosos programas curriculares y de producción con la aplicación de las más modernas tecnologías en la docencia (2).

Por su proyección, se define como un centro experimental, docente–productor. En ella discurren los vínculos formativos, productivos e investigativos. Ha sido de interés de la institución modificar sustancialmente sus estructuras y promover una formación desde la producción. Lo anterior ha traído consigo que desde el punto de vista organizacional, los grupos a partir del cuarto año de la carrera se modifiquen y de brigadas docentes pasen paulatinamente a funcionar como proyectos productivos. La universidad al irse consolidando, deberá poseer una visión apreciable sobre los grupos creados, pues en gran parte, de ellos se conformarían los activos intangibles que generarán competencias a largo plazo (3).

Esta universidad tiene centradas fundamentalmente sus fuerzas en la formación de un personal comprometido con los principios de la Revolución y la producción de software con una alta calidad. De igual manera contribuye hacia el desarrollo de soluciones tecnológicas integrales para la economía nacional y la exportación, mediante un amplio grupo de programas. Está compuesta por diez facultades, siete en la Sede Central y tres Facultades Regionales, donde se llevan a cabo una serie de proyectos productivos desarrollados por estudiantes, profesores y trabajadores. En cada una de las diez facultades de la Sede Central, existe un personal que se dedica a la evaluación de la calidad de los proyectos productivos así como los productos que se obtienen en ellos.

Entre los proyectos que se desarrollan se encuentra “SIG-MÓVILES”. El cual tiene como propósito el desarrollo de Sistemas de Información Geográfica (SIG) para móviles. Este proyecto no cuenta con un procedimiento que garantice la calidad de los procesos que en él se desarrollan. Lo cual evidencia que no existe una correcta asignación de los roles, debido a que el equipo de desarrollo no tiene implícito ningún rol relacionado con la administración de la calidad de sus productos. No se han definido mecanismos para la evaluación objetiva, entre los que se encuentran revisiones, pruebas, entre otros.

Además tampoco se tienen en cuenta modelos y estándares sobre los cuales debe de estar regido todo el trabajo del proyecto. Esto trae consigo que se incumpla con los lineamientos mínimos de calidad a considerar en la producción de software expuestos por el Centro de Calidad para soluciones Tecnológicas (Calisoft), los cuales aseguran que el proceso de desarrollo cumpla con los requerimientos establecido.

La deficiencia antes mencionada influye de manera directa en la calidad que debe tener el producto final. Además es la consecuencia de que el proyecto encontrándose en la etapa de diseño e implementación aún no se le haya aplicado algún procedimiento que permita detectar y corregir cualquier error presente en el mismo. Los diferentes aspectos a los que se hacen referencia influyen de manera negativa en la planificación del cronograma de trabajo, en el cumplimiento de los requisitos de calidad, en la aceptación y confianza que se debe lograr en el cliente.

Teniendo en cuenta la situación problemática anteriormente planteada se traza el siguiente **Problema a Resolver**: ¿Como contribuir a garantizar la calidad de los productos del proyecto SIG-MÓVILES en aras de satisfacer los lineamientos mínimos de calidad establecidos?

Basado en lo antes expuesto se ha definido como **Objeto de Estudio**: El proceso de aseguramiento de la calidad del software, el cual deriva como **Campo de Acción** el proceso de aseguramiento de la calidad en el proyecto "SIG-MÓVILES".

Para darle solución al problema planteado se propone como **Objetivo General**: Definir una estrategia para el aseguramiento de la calidad en el proyecto "SIG-MÓVILES".

Como **Idea a Defender** el trabajo plantea que: Con la definición correcta de una estrategia para el aseguramiento de la calidad de software se logrará el cumplimiento de las metas trazadas en el proyecto, así como la entrega del proyecto "SIG-MÓVILES" en tiempo y con los lineamientos mínimos de calidad establecidos.

Para darle solución al problema y cumplir el objetivo trazado se proponen las siguientes **Tareas de la Investigación**:

1. Investigar la documentación correspondiente con los procesos o etapas de la gestión de la calidad, específicamente el aseguramiento de la calidad.
2. Caracterizar los modelos y estándares que existen nacional e internacionalmente para el proceso de aseguramiento de la calidad de un software.
3. Caracterizar el proyecto "SIG-MÓVILES".
4. Definir la estrategia de aseguramiento de la calidad del software para el proyecto.
5. Validar la estrategia propuesta.

6. Analizar los resultados obtenidos a partir de la propuesta.

En el presente trabajo, se utilizaron diversos **métodos investigativos** entre ellos:

- ✓ El método teórico **histórico-lógico** con el objetivo de investigar la documentación referente a los temas relacionados con la planificación, el control y el aseguramiento de la calidad.
- ✓ El método teórico **analítico-sintético** con el fin de analizar la implantación de diversas estrategias aplicadas en distintos proyectos de similar desarrollo en la universidad.
- ✓ El método empírico de la **entrevista**, la cual se le realizó al jefe del proyecto con el fin de determinar la situación actual existente en el proyecto.
- ✓ El método empírico de la encuesta, la cual se le realizó a 6 estudiantes pertenecientes al grupo de desarrolladores del proyecto, con el fin de determinar el nivel de aceptación que tuvieron los cursos impartidos.

Este trabajo está compuesto por tres capítulos de los cuales se abordará:

- En el Capítulo I, la definición de la teoría relacionada con la calidad del software, así como, de varios conceptos necesarios para la realización de una estrategia para el aseguramiento de la misma.
- En el Capítulo II se llevará a cabo de forma general la propuesta de una Estrategia para el Aseguramiento de la Calidad del proyecto SIG-MÓVILES, partiendo de un análisis de la situación actual del proyecto la cual servirá como base para la aplicación de la estrategia.
- En el Capítulo III se procederá a validarla propuesta realizada en el capítulo anterior a partir de las metas y prácticas que propone el modelo y las normas seleccionadas para asegurar la calidad en el proyecto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA ESTRATEGIA PARA EL ASEGURAMIENTO DE LA CALIDAD

Introducción

La calidad del software es un término que ha adquirido gran relevancia con el paso del tiempo, ya que se considera como uno de los principales activos con los que cuenta un país para mejorar su posición competitiva global. Para conseguir una buena calidad del software es esencial utilizar los modelos y métodos apropiados para controlar el proceso de desarrollo. Con el objetivo de lograr un mejor entendimiento del tema, en el presente capítulo se reflejan varios conceptos asociados a la calidad y el proceso de evaluación. Se describen distintos modelos, métricas y técnicas para asegurar la calidad en los productos software.

1.1 Calidad de Software

Según el Diccionario de la Real Academia Española (DRAE), es una propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie. De acuerdo con la ISO 9000:2000, es el grado en el que un conjunto de características inherentes cumplen con los requisitos. El Instituto de Ingeniería de Software (SEI) en su modelo CMMI define la calidad como: Grado en el cual un sistema, componente o proceso cumple con los requisitos especificados (4).

El producto de software posee un conjunto de cualidades que pueden ser detectadas por el usuario o, únicamente por profesionales de la informática con acceso al código. Estas cualidades pueden ser denominadas como factores de calidad externos o internos. La clave para obtener los factores externos radica en los internos pues para que los usuarios disfruten de las cualidades visibles, todo el equipo del proyecto debe aplicar técnicas internas que aseguren las cualidades ocultas.

La calidad del software puede medirse una vez terminado el producto. Esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en algunas de las etapas anteriores, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control y aseguramiento durante todo el ciclo de vida del software. Una vez analizado los conceptos se concluye que para que un software se desarrolle o tenga calidad debe de estar de acorde a las normas métricas y estándares establecidos, no solo este como producto terminal sino todo el proceso de desarrollo del mismo.

1.2 Gestión de la Calidad

La norma ISO 9000 define la Gestión de la Calidad como: “Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. Se implanta por medios tales como la planificación, el control, el aseguramiento (garantía) y la mejora de la calidad, en el marco del sistema de calidad” (5).

Los procesos de Gestión de la Calidad incluyen todas las actividades de la organización ejecutante que determinan las políticas, los objetivos y las responsabilidades relativas a la calidad de modo que el proyecto satisfaga las necesidades por las cuales se emprendió. (6)

Por lo abordado anteriormente se concluye que la gestión de la calidad puede definirse como: “El proceso principal a realizar por cualquier empresa desarrolladora de software en función de asegurar la calidad de sus productos, ejecutada a nivel de empresa y basada en la determinación y aplicación de las políticas de calidad de la misma”. Este proceso se encuentra formado por un conjunto de actividades definidas como Planificación, Aseguramiento, Control y Mejora de la Calidad.

1.2.1 Planificación de la Calidad

Según Juran (1993) “La planificación de la calidad es el proceso de establecer objetivos de calidad de largo alcance y definir un enfoque para satisfacer objetivos.” La Planificación Estratégica de la Calidad es el proceso por el cual una empresa define su razón de ser en el mercado, su estado deseado en el futuro y desarrolla los objetivos y las acciones concretas para llegar a alcanzar el estado deseado. Se refiere, en esencia al proceso de preparación necesario para alcanzar los objetivos de la calidad (7).

Los objetivos perseguidos con la Planificación Estratégica de la Calidad son:

- Proporcionar un enfoque sistemático
- Fijar objetivos de calidad.
- Conseguir los objetivos de calidad.
- Orientar a toda la organización.
- Válida para cualquier periodo de tiempo

Partiendo del criterio de que la planificación requiere considerablemente de la participación de los directivos, debido a que son estos los que determinan los objetivos que se deben incluir en el plan del negocio y quienes tienen la tarea de transmitir esta información hacia niveles inferiores de la organización con la función de identificar las acciones, asignar recursos y responsabilidades necesarias para el cumplimiento de estas acciones sería conveniente decir que mediante este proceso

se lograría disminuir el tiempo de desarrollo del producto ,daría la posibilidad de establecer calendarios precisos y sobre todo los mas importante, cumplirlos.

James (1997) añade que la planificación de la calidad va más allá de la planificación estratégica porque se le añade un requerimiento de mejora a este concepto, lo importante es que la dirección y el personal trabajen juntos para asegurar la mejora continua usando el plan como vehículo (8).

Propósitos del Plan de Calidad

El plan de calidad se centra en detallar las normas para el proyecto y los criterios que se utilizan para medir y determinar si los resultados son los esperados, además de crear y documentar un plan para cumplir con esas normas (9).

Este proceso se centra en las políticas de calidad propias de la organización, tiene como objetivo principal el desarrollo del Plan de Calidad en el mismo se deben establecer:

- Los estándares, normas de calidad y regulaciones que afectan al proyecto.
- Los estándares que deberán desarrollarse específicamente para el proyecto.
- La manera de asegurar la conformidad con dichos estándares.
- Los procesos y planes de mejora continua.
- Las métricas que se utilizarán para medir los resultados del proyecto.
- Los procesos que se utilizarán para aplicar dichas métricas.
- El grado de calidad del producto y cualidades que deben ser poseídas por los entregables del proyecto.

Por lo abordado anteriormente se concluye que la Planificación de la Calidad puede definirse como el proceso en el cual se ejecutan una serie de actividades encaminadas a exponer los objetivos de la calidad y las acciones necesarias para satisfacer dichos objetivos.

1.2.2 Control de la Calidad

El control de la calidad del software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centrados en dos objetivos fundamentales (5):

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

Las distintas actividades para la implantación del control de la calidad en el desarrollo de software son (10):

1. Aplicación de metodología y técnicas de desarrollo.
2. Reutilización de procesos de revisión formales.
3. Prueba del software.
4. Ajustes a los estándares de desarrollo.
5. Control de cambios, mediciones y recopilación de información.
6. Gestión de informes sobre el control de calidad.

El aspecto a considerar en el Control de la Calidad del Software es la “Prueba del Software”.

Pruebas de software

Las pruebas de software pueden considerarse como “una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza la evaluación de algunos aspectos” (11).

Los objetivos de las pruebas de software según Myers (1979) son:

1. El proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si detecta un error no detectado hasta entonces.

Especificación de los niveles de prueba

- Pruebas unitarias

Prueba el diseño y el comportamiento de cada uno de los componentes una vez construidos. Una prueba unitaria (12) es una forma de probar el correcto funcionamiento de un módulo. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. El objetivo de estas pruebas es aislar cada parte del programa y mostrar que las partes individuales son correctas.

- Pruebas de integración

Comprueba la correcta unión de los componentes entre sí a través de sus interfaces, y si cumplen con la funcionalidad establecida. Las pruebas de integración se realizan durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

- Pruebas del sistema

Las pruebas del sistema (12) tienen como objetivo encontrar defectos en el funcionamiento del sistema completo. Dentro de estas pruebas se incluyen las siguientes:

- Pruebas funcionales: dirigidas a asegurar que el sistema realiza correctamente todas las funciones detalladas en los requerimientos.
- Pruebas de rendimiento: orientadas a determinar si los tiempos de respuesta se encuentran dentro de los límites establecidos en las especificaciones del sistema.
- Pruebas de facilidad de uso: comprueban la experiencia del usuario en el uso del sistema, asegurando que el mismo se acomode al modo de trabajo de los usuarios, y que le aporta elementos que faciliten el ingreso y la recuperación de los datos.

- Pruebas de implantación y aceptación

El objetivo de las pruebas de implantación (12) es comprobar el correcto funcionamiento del sistema en su entorno operativo. Estas pruebas permiten que el usuario realice la aceptación del sistema desde el punto de vista operativo, lo que incluye la instalación y el mantenimiento del sistema.

- Pruebas de regresión

El objetivo de estas pruebas es comprobar que los cambios sobre un componente del sistema no introducen errores en otros componentes. Las pruebas de regresión se llevarán a cabo cada vez que se haga un cambio sobre un componente, ya sea para corregir un error como para realizar una mejora en el mismo. Dichas pruebas consistirán en repetir las pruebas unitarias y de integración, y/o las pruebas del sistema.

Las técnicas para la aplicación de cada una de estas pruebas son las siguientes:

- La prueba de especificación (12), o "prueba de caja negra": Verifica el comportamiento de la interfaz.
- La prueba de estructura, o "prueba de caja blanca": Verifica la implementación interna de la unidad.

De esta forma se concluye que el Control de la Calidad puede definirse como el proceso en el cual se comprueba de forma sistemática que el producto de software este de acorde a las especificaciones de diseño establecidas.

1.2.3 Aseguramiento de la Calidad

El aseguramiento de la calidad es un conjunto de actividades planificadas, sistemáticas y necesarias para aportar la confianza de que el producto de software satisfará los requisitos dados de calidad (13). Según la norma ISO 9000-3 el propósito del aseguramiento de la calidad es proporcionar una adecuada seguridad de que los productos de software y los procesos en el ciclo de vida del proyecto están conformes con sus requerimientos específicos y se ajustan a sus planes establecidos

El aseguramiento de la calidad aborda principalmente:

- Un enfoque de gestión de la calidad.
- Métricas del software.
- Verificación y validación a lo largo del ciclo de vida del software, incluyendo pruebas, procesos de revisión y auditorías.
- Gestión de configuración del software.
- El control de la documentación del software.
- Un procedimiento que asegure los ajustes a los estándares en el proceso de desarrollo de software siempre que esto sea posible.

El aseguramiento de la calidad, como actividad de soporte en el proceso de desarrollo, comprende procedimientos para la aplicación efectiva de métodos y herramientas, revisiones y estrategias de prueba, control de cambios, aseguramiento de ajuste a los estándares entre otras. El Instituto de Ingeniería del Software (SEI) recomienda la aplicación de las siguientes actividades:

- Establecimiento del Plan de Aseguramiento de la Calidad para un proyecto.
- Participación en el desarrollo de la descripción del proceso de software.
- Revisión de las actividades de ingeniería del software.
- Auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de software.
- Registrar lo que no se ajuste a los requisitos e informar a los superiores.
- Coordinar el control de cambio.

El Aseguramiento de la calidad es un modo planificado y sistemático de asegurar a la gerencia que se aplican los estándares, prácticas, procedimientos y métodos definidos del proceso (4).

1.2.4 Mejora de la Calidad

Mejora de la calidad: es la parte de la Gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos establecidos. Significa lograr un cambio ventajoso, convirtiéndose en un sinónimo de avance. La Mejora es la parte de la Gestión que contribuye, por medio de las mediciones, a los análisis de los datos y auditorías, a efectuar mejoras en la calidad del software (14).

1.3 Auditorías

Las auditorías según el estándar ISO 19011:2002 se define como: proceso sistemático, independiente y documentado para evaluar el estado actual (evidencias de la auditoría) y evaluarlas de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría (13).

Una Auditoría de Calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectarla adversamente. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos (14).

Las razones para realizar una auditoría son:

- Establecer el estado del proyecto.
- Verificar la capacidad de realizar o continuar un trabajo específico.
- Verificar qué elementos aplicables del programa o Plan de Aseguramiento de la Calidad han sido desarrollados y documentados.

- Verificar la adherencia de esos elementos con el programa o Plan de Aseguramiento de la Calidad.

Este proceso necesita del conocimiento de varios puntos claves como: establecer la infraestructura, conocer que proyectos de mejora existen, instaurar equipos para los proyectos, proporcionar recursos a los equipos, a la formación y a la motivación para lograr así diagnosticar las causas, fomentar remedios, y establecer controles para conservar los beneficios (15). La Mejora de la Calidad es un proceso que toda empresa debe implementar, ya que día a día los productos necesitan una mejor adecuación al uso, porque el cliente cada vez exige más y la competencia también exige que produzcamos con mayor calidad e innovación (16).

1.4 Estándares

Los Estándares de Calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software (17). La utilización de estándares como parte del proceso de aseguramiento de la calidad está presente en todo el proceso de desarrollo, y siendo más precisos, en todo el ciclo de vida. La presencia de estándares asociados directa o indirectamente al desarrollo de software es abundante.

El estándar IEEE 730 es una recomendación para la elaboración de un Plan de Aseguramiento de la Calidad del Software (SQAP) para proyectos de desarrollo de software. Cuando en un proyecto de desarrollo de software se incluye un plan de estos, las decisiones relacionadas con la calidad deben ser tomadas con anticipación y por lo tanto, deben ser estudiadas y razonadas suficientemente antes de iniciar el desarrollo (18).

La Organización internacional de estándares, en inglés International Standard Organization(ISO) ha aportado estándares para la industria del software. Algunos de los más importantes son:

- ISO 9001. *Quality Systems Model for Quality Assurance in Design, Development, Production, Installation and Servicing.*
- ISO 90003. *Guidelines for Application of ISO 9001 to the Development, Supply and Maintenance of Software.*
- ISO 90042. *Quality Management and Quality Systems Elements.*

La calidad desempeña un rol determinante para la competitividad de la empresa. Cuando una empresa está funcionando y decide implantar un Modelo o un Estándar de Calidad del Software, es señal que la empresa tiene el propósito de permanecer y crecer en el mercado, ser competitiva, proteger los

intereses de los accionistas, cuidar la fuente de trabajo y mejorar la calidad de vida de su personal (14).

Implantar Modelos o Estándares de Calidad tiene como objetivo principal que las empresas desarrollen sistemáticamente, productos, bienes y servicios de mejor calidad y cumplan con las necesidades y deseos de los clientes. Para esto, se requiere de un Modelo o Estándar que: permita:

1. Unir la misión de la empresa y el esfuerzo de cada área en una sinergia de resultados hacia la competitividad y la calidad de clase mundial.
2. Tener procesos y procedimientos ágiles; y comprensibles para todos los involucrados, pasando por las etapas de desarrollo, prueba, producción y satisfacción del cliente.

Normas ISO 9000

Las normas ISO 9000, son un conjunto de enunciados que conforman un modelo para desarrollar un sistema de gestión de la calidad en una empresa. La familia de estándares ISO 9000, creada por ISO (Organización Internacional para la Estandarización) establece los lineamientos y guía el aseguramiento de calidad de los productos, servicios y relaciones con el cliente.

Cabe señalar que las normas ISO 9000 se refieren al sistema de gestión de calidad y no a las especificaciones para la elaboración de un producto. Es decir, lo que se logra es homologar sistemas de calidad para los productos. Según lo expresado en las normas, la participación de recursos para transformar requisitos de las partes interesadas en productos de acuerdo a sus demandas representa un enfoque basado en procesos.

Norma ISO/IEC 9126

La norma ISO/IEC 9126 está enfocada a la calidad de Producto y consta de las siguientes partes:

- 1) Modelo de Calidad, 2) Métricas externas, 3) Métricas internas, 4) Calidad en el uso de métricas.

Esta primera parte de la ISO 9126 describe el modelo de calidad del producto de software. Especifica seis características de calidad interna y externa, las cuales están divididas en sub características, son manifestadas externamente cuando el software es utilizado como parte de un sistema, siendo un resultado de los atributos internos del software. La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas.

Esta norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte,

mantenimiento, aseguramiento de la calidad, y auditoría del software. Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsable de especificar y evaluar la calidad del software. En el trabajo se decide adoptar la Norma ISO 9126 para regir los atributos de calidad siguiendo la propuesta realizada en el documento: "Lineamientos mínimos de calidad".

1.5 Modelos

Los Modelos de Calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos claves y permiten medirlos avances en calidad (10).

1.5.1 CMMI

CMMI es un modelo que estudia los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de cinco niveles. La madurez de un proceso es un indicador de la capacidad para construir un software de calidad. Fue desarrollado por el *Software Engineering Institute (SEI) perteneciente a Carnegie Mellon University*.

CMMI provee a las organizaciones de software una guía de cómo realizar un control de los procesos de desarrollo y mantenimiento de software. Permite seleccionar estrategias de mejoras determinando la madurez de los procesos existentes e identificando factores críticos respecto de la calidad del software.

El modelo está formado por dos representaciones, el continuo y el escalonado. Para la representación continua se utilizan los términos nivel de capacidad o áreas de procesos de capacidad. Para la representación escalonadas e utiliza los términos nivel de madurez o madurez organizacional. CMMI cuenta con cinco niveles de capacidad y veintidós áreas de procesos. Estas áreas de procesos están divididas en categorías (Gestión de Procesos, Gestión de Proyectos, Ingeniería y Soporte).

En la categoría de Soporte se encuentra el área de proceso que le compete a este trabajo, Garantía de la Calidad del Procesos y del Productos (PPQA). Esta área de proceso se encuentra ubicada en el nivel 2 de madurez, en el cual el proceso está administrado.

Área de proceso Aseguramiento de la calidad de proceso y de producto según modelo CMMI v1.2 (4), involucra:

- Evaluar objetivamente los procesos, los productos de trabajo y los servicios ejecutados frente a las descripciones de proceso, los estándares y los procedimientos aplicables.
- Identificar y documentar las no conformidades.
- Proporcionar realimentación al equipo del proyecto y a los gerentes sobre los resultados de las actividades de aseguramiento de la calidad
- Asegurar que sean tratadas las no conformidades.

Algunos ejemplos de formas de realizar evaluaciones objetivas son:

- Auditorías formales por organizaciones de aseguramiento de la calidad separadas desde el punto de vista organizativo.
- Revisiones entre pares que pueden ser ejecutadas en varios niveles de formalidad.
- Revisiones en profundidad del trabajo en el lugar donde se realiza (es decir, auditorías “de escritorio”).
- Revisiones y comentarios distribuidos de productos de trabajo.

Las no conformidades son problemas identificados en las evaluaciones que reflejan la falta de adherencia a los estándares, descripciones de proceso o procedimientos aplicables. El estado de las mismas proporciona una indicación de las tendencias de calidad. Los problemas de calidad incluyen no conformidades y resultados del análisis de tendencia.

Algunos ejemplos de formas de resolver las no conformidades dentro del proyecto son:

1. Corregir la no conformidad.
2. Cambiar las descripciones de proceso, estándares, o procedimientos que fueron incumplidos.
3. Obtener una excepción para cubrir la no conformidad.

Establecer y mantener registros de las actividades de aseguramiento de la calidad.

Productos de trabajo típicos

1. Registros de evaluación.
2. Informes de aseguramiento de la calidad.
3. Informes del estado de las acciones correctivas.
4. Informes de las tendencias de calidad.

1.5.2 SPICE

El modelo ISO/SPICE proporciona un marco para la evaluación de los procesos de software. Este marco puede ser utilizado por organizaciones cuya actividad incluya la planificación, gestión, control o mejora de los procedimientos de adquisición, suministro, desarrollo, operación, evolución y soporte de software. ISO/IEC TR 15504 provee un marco de trabajo para la valoración del proceso de software y parte de los mínimos requerimientos para la realización de una valoración para asegurar consistencia y repetitividad de las mediciones. El modelo ISO/SPICE es tridimensional, la primera dimensión es funcional (procesos), la segunda es capacidades (niveles), la tercera de adecuación o efectividad (calificaciones).

Dimensión de Procesos

La dimensión está organizada jerárquicamente de la siguiente manera:

1) Categorías de procesos, que agrupan procesos comunes. 2) Procesos, que logran propósitos técnicos. 3) Prácticas base, operaciones que conforman un proceso.

El Aseguramiento de la Calidad según el modelo SPICE

El aseguramiento de la calidad es tratado dentro de la categoría de proceso de soporte. Los procesos de soporte pueden ser empleados en varios puntos del ciclo de vida y pueden ser realizados por la organización que los emplea, el cliente o por una organización independiente.

Actividades realizadas como parte del aseguramiento de la calidad según SPICE:

- ✓ Seleccionar los estándares del proyecto.
- ✓ Revisar las actividades de ingeniería del software frente a los planos, estándares y procedimientos seleccionados.
- ✓ Auditar los productos de software frente a los estándares y procedimientos seleccionados.
- ✓ Informar de los resultados de las actividades anteriores, en particular, de las desviaciones, a los niveles apropiados de dirección y plantilla.
- ✓ Tratar las desviaciones: Las desviaciones se tratan en el nivel de gestión apropiado, escalando el siguiente nivel, cuando sea necesario, hasta resolverlas.

Aspectos fundamentales que conllevaron a la elección del modelo CMMI.

Después de estudiados los modelos anteriores se pueden señalar varios puntos críticos que permiten elegir CMMI ante las Normas ISO y SPICE, algunos de estos puntos son:

- CMMI establece una estructura conceptual de referencia para medir la mejora continua de procesos, mientras que ISO 9000 solamente especifica un nivel aceptable mínimo de calidad para procesos de desarrollo de software.
- CMMI y SPICE son modelos muy similares sin embargo, un factor determinante para la elección del mismo debido a las condiciones actuales de la Universidad lo representa el hecho de que:
 - CMMI sea un modelo altamente recomendable para organizaciones donde las características de los procesos son inmaduros, carentes de formas sistemáticas de producción de software, donde el éxito del desarrollo depende más del heroísmo de los desarrolladores. Mientras que SPICE generalmente aplicado en organizaciones en vueltas con planificación, gestión, seguimiento, control y mejora de procesos de adquisición, suministro, desarrollo, operación, evolución y soporte de software.
- CMMI fue concebido para la producción de software y tiene técnicas para ser independiente e incremental en su avance, lo cual motiva al personal. Mientras que ISO es muy general, no proporciona información de cómo aplicarlo a empresas de menor tamaño, tampoco proporciona directrices para su implementación en varias industrias

1.6 Revisiones

Las revisiones según el estándar ANSI/IEEE Std 10281988 se define como: evaluación de un elemento para determinar diferencias con los resultados planeados y recomendar mejoras. Las revisiones del software son un "filtro" para el proceso de Ingeniería del Software, se aplican en varios momentos del desarrollo del software y sirven para detectar errores y defectos que pueden ser eliminados. Las revisiones del software sirven para "purificar" las actividades de la Ingeniería del Software que suceden como resultado del análisis, diseño y codificación (13).

Existen varios tipos de revisiones que pueden llevarse a cabo durante el proceso de desarrollo de software (13).

- Reuniones Informales: Una reunión que tiene lugar alrededor de una máquina, u otro espacio en cual se discuten problemas técnicos.
- Presentación formal: Una presentación formal de un diseño de software, una aplicación o cualquier producto de trabajo que se realiza bajo la audiencia de de clientes, ejecutivos o personal técnico.
- Revisiones técnicas formales: Actividad desarrollada por los ingenieros del software.

La revisión técnica formal (RTF), a veces llamada inspección, es el filtro más efectivo desde el punto de vista del aseguramiento de la calidad y es un medio efectivo para mejorar la calidad del software. El defecto se define como una anomalía del producto. Dentro del contexto del proceso del software, los términos defecto y fallo son sinónimos. Ambos implican un problema de calidad que es descubierto después de entregar el software a los usuarios finales. El objetivo principal de las RTF es encontrar errores durante el proceso, de forma que se conviertan en defectos después de la entrega del software. El beneficio de estas RTF es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software.

Las actividades de diseño introducen entre el 50 y 65% de todos los errores durante el proceso de software. Sin embargo, se ha demostrado que las RTF son efectivas en un 75% a la hora de detectar errores. Con la detección y la eliminación de un gran porcentaje de errores, el proceso de revisión reduce substancialmente el coste de los pasos siguientes en las fases de desarrollo y mantenimiento.

Los objetivos de la RTF son:

- Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.
- Verificar que el software bajo revisión alcanza sus requisitos.
- Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos,
- Conseguir un software desarrollado en forma uniforme.
- Hacer que los proyectos sean más manejables.

La RTF sirve 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. Es una clase de revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupo de evaluaciones técnicas del

software. Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida.

1.7 Métricas del software

El glosario de términos de estándares de IEEE define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento. Estas pueden definirse como: Una serie de medidas o pasos que ayudan a definir con mayor exactitud el desarrollo y calidad de un producto.

Las métricas pueden agruparse según varias clasificaciones (métricas orientadas al software, proyecto, proceso y personas de forma genérica), a continuación se enuncian algunas de las clasificaciones para las métricas (13).

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

De esta forma se concluye que las métricas pueden definirse como la actividad en la cual se aplican una serie de pasos en los que se incluyen la utilización de medidas cuantitativas las cuales permiten al personal del proyecto planificar y evaluar la calidad del producto.

1.8 Estado actual de la Calidad de Software

1.8.1 Calidad de software en el Mundo

La calidad en las empresas ha evolucionado, si se analiza desde los inicios de los procesos de industrialización a mediados del siglo XIX hasta cerca de 1940. Este es un término que se relacionaba con la inspección en los productos con el propósito de detectar errores, de ésta fecha hasta los años 80`s el control de calidad se convirtió en un ejercicio de control estadístico cuyo propósito era impedir que el producto defectuoso llegara al cliente y a partir de los 80`s se inician procesos de gestión de calidad total, que busca garantizar la calidad por medio de la planificación y la creación de modelos de forma permanente.

Las teorías de gestión de procesos de software son una síntesis de los conceptos de gestión de la calidad trabajadas por Deming, Crosby y Juran entre otros; que en los últimos 30 años han sido usadas para resolver problemas comunes de muchas organizaciones. Se han descubierto soluciones pero existe una gran brecha entre el estado de la práctica y el estado del arte; sin embargo muchos de estos conceptos y soluciones han sido usados para construir lo que hoy se conoce como modelos de mejora de procesos (19).

La calidad del software es objeto de estudio por parte de diversos grupos de investigación nacional e internacional, sin embargo existen organizaciones académicas y científicas que supervisan y promueven los estudios en esta área; entre las principales organizaciones están el instituto de Ingeniería de Software SEI y el Grupo de Gestión de Objetos OMG.

1.8.2 Calidad de software en Cuba

En estudios realizados por el Centro de Estudios de Ingeniería de Sistemas (CEIS) en empresas nacionales se detectaron problemas entre los que se encuentran: los resultados alcanzados no cubren las expectativas, la productividad es baja, la cantidad real de recursos a consumir en tiempo principalmente es casi impredecible, el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida y los proyectos están excesivamente tarde (20).

Los resultados de este estudio arrojan la necesidad de establecer políticas de calidad en las organizaciones del país con el fin de mejorar la productividad de las empresas y lograr una adecuada

satisfacción de los clientes tanto el mercado nacional como el internacional. Establecer un adecuado Sistema de Aseguramiento de la Calidad contribuye considerablemente a elevar la calidad en el o los productos resultantes del proyecto de software (20).

1.8.3 Calidad de software en la UCI

La UCI es uno de los principales centros desarrolladores de software en los cuales se realizan todo un conjunto de actividades para lograr la implantación de un sistema de calidad. El Centro para la excelencia en el desarrollo de proyectos tecnológicos (CALISOFT) ubicado en la misma es una organización enfocada a contribuir al desarrollo de la industria cubana de software facilitando la implementación de las mejores prácticas en el proceso de desarrollo y/o mantenimiento de software (21).

Esta entidad es responsable de la evaluación y certificación de productos, procesos y organizaciones según normas nacionales e internacionales y de la asesoría, adiestramiento y formación continua de los especialistas en el país en los temas de calidad e ingeniería de software. Esta organización se encuentra actualmente en el proceso de certificación del nivel dos de CMMI, debido a esto la universidad ha sido objeto de de pruebas pilotos en numerosos proyectos.

Conclusiones del capítulo.

En el este capítulo se llevó a cabo un estudio de los conceptos básicos relacionados con la calidad del software, haciendo énfasis en el Área de Aseguramiento de la Calidad por ser el tema que ocupa el trabajo de diploma. El análisis de estas definiciones ha permitido realizar una valoración de los mismos, haciendo posible destacar relaciones y diferencias entre estos. Teniendo en cuenta este factor y tras haber realizado un estudio exhaustivo de los temas a nivel nacional como internacional, se decide adoptar el modelo CMMI para desarrollar la propuesta de solución, según las características y ventajas que proporciona el mismo. Guiados por este, se hace evidente la utilización de listas de chequeo para evaluar objetivamente el producto, detectando y procurándole seguimiento a las no conformidades detectadas.

CAPÍTULO 2: Propuesta de la Estrategia para el Aseguramiento de la Calidad.

Introducción

Con el objetivo de realizar la definición satisfactoria de la estrategia para asegurar la calidad en el proyecto SIG-MÓVILES, se realiza un estudio detallado que comprende las características, estructura y situación actual que presenta el mismo. Respaldo por la aplicación de mecanismos para la obtención de información como son la entrevista y la lista de chequeo, ambas aplicadas al jefe del proyecto por ser este la figura clave dentro del mismo. Procurando proveer de esta manera una visión general del proyecto y del producto actual en desarrollo.

Luego se realizará un análisis y descripción de los procesos de Aseguramiento de la Calidad el cual será guiado por las prácticas que propone el Modelo de Mejora CMMI. [Ver Entrevista](#) y [Ver Lista de Chequeo](#).

2.1 Estado actual del proyecto SIG-MÓVILES

El proyecto Sistema de Gestión Geográfica para Móviles, perteneciente a la línea SIG-Móviles que a su vez pertenece al Departamento de Geoinformática de la facultad 6 tiene entre sus líneas fundamentales el desarrollo de herramientas para el montaje de SIG a diversos entornos sociales y gubernamentales. En el mismo se apoya en gran medida la toma de decisiones sobre varios procesos de negocio y fomentan el camino de la informatización de la sociedad cubana sobre tecnologías libres. Sistema de Gestión Geográfica para Móviles es un proyecto de investigación y desarrollo (I+D) e incluye varias líneas de la rama de la geomática que apoyan directamente la formación de estudiantes en el segundo perfil del polo. Es un equipo de trabajo que une a equipos de desarrollo del Centro de Telecomunicaciones de la facultad 2 que tiene como meta fortalecer la experiencia y los productos con la realización de un SIG para Móviles y abrir un espacio sólido en el mercado de aplicaciones de esta rama.

2.2 Estructura y Composición

En el presente sub-epígrafe se abordarán temas sobre la estructura y composición del proyecto SIG-MÓVILES, los diferentes roles desempeñados dentro del proyecto, sus responsabilidades y el soporte tecnológico con que este cuenta.

2.2.1 Roles y responsabilidades definidos

En el proyecto SIG-MÓVILES se encontraban inicialmente definidos los roles y responsabilidades [ver Anexo1](#). Luego de haber analizado la tabla anterior, se considera que la asignación de roles y responsabilidades que presenta el proyecto no es suficiente para asegurar la calidad de los procesos y productos que el mismo desarrolla debido a los siguientes factores:

- El proyecto solo presenta un único rol responsable de la calidad.
- Las tareas asignadas a dicho rol solo tratarían de manera superficial e insuficiente el aseguramiento de la calidad de los productos y procesos.

Por esta razón se realiza una propuesta de cómo deben quedar definidos los roles encargados de asegurar la calidad dentro del proyecto SIG-MÓVILES, esperando alcanzar un mayor nivel de eficiencia y organización.

2.2.2 Soporte tecnológico del proyecto

El proyecto cuenta con un total de 7 computadoras en perfecto estado en las cuales trabajan 19 personas en total, 14 de ellos estudiantes y 5 profesores situados de la manera más eficiente posible para lograr aprovechar al máximo el horario laboral. Actualmente se trabaja sobre dos líneas de Investigación-Desarrollo: (1) Línea I+D Protocolo: WAP, para la cual se utiliza la arquitectura Cliente-Pasarela (Codificador y Decodificador)-Servidor de Información. (2) Línea I+D GVSIG Mobile y GVSIG Mini soportado bajo una arquitectura perteneciente a dichas plataformas.

Para realizar el diseño se utiliza Visual Paradigm, como gestor de base de datos se utiliza PostgreSQL, como lenguaje de modelación se utiliza UML, como herramienta de desarrollo GVSIG Mobile, GVSIG MINI, GeoServer y MapServer y como IDE se utiliza Eclipse. La metodología para el desarrollo de los productos utilizada es RUP. Para el almacenamiento de la información se utiliza el sistema de Gestión Documental Exscriba.

2.3 Estrategia del Aseguramiento de la calidad

Para mayor entendimiento sobre el trabajo en cuestión es necesario procurar una breve explicación acerca del significado del término estrategia. **Una estrategia** no es más que el proceso seleccionado mediante el cual se espera lograr alcanzar un estado futuro. Es un conjunto de acciones planificadas en el tiempo que se llevan a cabo para lograr un determinado fin (22).

A partir de los conceptos tratados en el capítulos anterior, las definiciones que ofrece CMMI , se procede a diseñar una estrategia que sea aplicable al proyecto SIG-MOVILES de la facultad, teniendo en cuenta la situación actual del proceso productivo del mismo. Para la definición de la estrategia se propone la siguiente **estructura**:

2.3.1 Estructura de la Estrategia

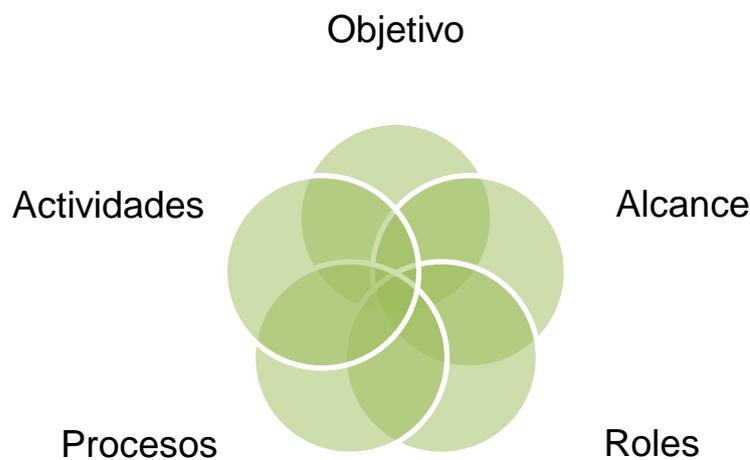


Ilustración 1 Estructura de la Estrategia

Objetivo

El propósito de esta estrategia es proporcionar al equipo de trabajo y a la dirección visibilidad objetiva de los procesos y de sus productos asociados. Esto incluye revisar las actividades y los productos de trabajo frente a los criterios que minimizan la subjetividad.

Alcance

El alcance de un proyecto es la suma total de todos los productos y sus requisitos o características. Se utiliza a veces para representar la totalidad de trabajo necesitado para dar por terminado un proyecto. La realización de esta estrategia está encaminada hacia el aseguramiento de la calidad de los productos y procesos del proyecto SIG-MÓVILES.

Roles

Un rol en el proceso de desarrollo de software puede verse como un papel a desempeñar por parte de la persona perteneciente al proyecto de acorde a sus habilidades, experiencias y actitudes el cual encierra todo un conjunto de tareas y responsabilidades a desempeñar durante el desarrollo del proyecto. Los roles que a continuación se definirán no necesariamente deberán ser asignados en su totalidad, estos serán adaptables de acorde al grupo interno de calidad de cada proyecto que se desee formar, según las características del proyecto en cuestión.

➤ Alta Gerencia

Cuando se asigna este rol, se refiere a todo aquel personal no perteneciente al grupo interno del proyecto, un ejemplo del mismo les sería asignado a los jefes de departamento y de centro al cual tributa el proyecto. La persona a la cual se le asigne este rol debe de poder:

- Analizar los Reportes del Área de Aseguramiento de la Calidad y tomar decisiones estratégicas a partir de la información de estos reportes.
- Publicar los reportes de PPQA.

➤ Administrador de la Calidad.

La persona a la cual se le asigne este rol debe de poder:

- Mantener actualizado el Registro de evaluaciones del proyecto
- Monitorear el cumplimiento de las acciones correctivas y el estado de las no conformidades.
- Notificar periódicamente a los Revisores Líderes de las evaluaciones el estado de las no conformidades.
- Notificar al Coordinador de PPQA solicitudes de escalamiento de no conformidades.

➤ Coordinador de PPQA

La persona a la cual se le asigne este rol debe de poder:

- Revisar el expediente de las evaluaciones antes de archivarlos.
- Redactar solicitudes de mejora de procesos y productos
- Realizar la evaluación periódica del desempeño de los revisores líderes
- Recepcionar solicitudes de escalamiento.

- Enviar la notificación de escalamiento al nivel correspondiente
- Documentar los permisos y comunicarlos a los proyectos.
- Realizar los reportes de PPQA e informar a la Alta Gerencia sobre las solicitudes de mejora, las evaluaciones de desempeño de los revisores la tendencia de calidad.

➤ Revisor Líder

La persona a la cual se le asigne este rol debe de poder:

- Efectuar la reunión de inicio de la evaluación.
- Organizar al equipo revisor y distribuir las tareas.
- Participar en la evaluación aplicando listas de verificación, entrevistas, etc.
- Redactar el Informe final de la evaluación.
- Efectuar la reunión de cierre de la evaluación.
- Realizar las evaluaciones de desempeño de los revisores.
- Comunicar los resultados y el expediente de la evaluación al Coordinador de PPQA.

➤ Revisor Técnico

La persona a la cual se le asigne este rol debe de poder:

- Ejecutar las RTF
- Elaborar el informe de las RTF.
- Participar en la reunión de apertura y cierre.

➤ Probador

La persona a la cual se le asigne este rol debe de poder:

- Ejecutar las pruebas a los sistemas y documentarlas.

➤ Diseñador de Pruebas.

El modelo en el cual se hizo referencia para la asignación de los roles anteriormente mencionados, adaptados a las características de los proyectos del departamento, no contempla la asignación de este rol, sin embargo se ha decidido incluir debido a la importancia que representan las actividades concedidas al mismo.

1. Definir el método de prueba y asegurar su implementación exitosa.
2. Identificar técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias y encauzar los recursos adecuados para las pruebas.
3. Definir los enfoques de prueba.
4. Examinar la arquitectura de software.
5. Definir la configuración del ambiente de prueba.
6. Estructurar la implementación de pruebas.
7. Definir los elementos de prueba.

Procesos

Los procesos definidos en esta estrategia no son más que las fases por las cuales se transitará para cumplir con la estrategia.

1. Planificación.
2. Aseguramiento.
3. Control.

1. Planificación

Proceso en el cual se llevará a cabo la planificación ordenada de todos los elementos que conducirán la estrategia, estos elementos se explicaran a través de las actividades y su explicación pertinente.

Incluye las actividades correspondientes a:

- Definir los objetivos de calidad para el proceso de desarrollo: En esta actividad se propone que el administrador de la calidad especifique los objetivos de calidad que va a tener el proyecto.
- Definir roles y responsabilidades a desempeñar en el proyecto: Para evitar la subjetividad o parcialidad, asegurar que las personas a las que se ha asignado responsabilidad y autoridad para llevar a cabo el aseguramiento de la calidad de proceso y de producto puedan realizar sus evaluaciones con la suficiente independencia y objetividad.
- Planificar como se llevará a cabo la capacitación del personal: Formar, según sea necesario, a las personas para realizar o dar soporte al proceso de aseguramiento de la calidad de proceso y de producto. Se deberá realizar una planificación de los cursos de capacitación los cuales serán impartidos al personal del proyecto por parte del Coordinador de la Calidad.

Algunos ejemplos de temas de formación son:

- Introducción a la Calidad de Software
 - Auditorías y Revisiones de Software
 - CMMI
 - Gestión de Calidad
 - Pruebas de Software
- Especificar sobre las herramientas, técnicas y metodologías que se utilizarán en el proyecto para la aplicación de la estrategia: Se deberá especificar por parte del Administrador de Calidad las herramientas que se utilizarán tales como: listas de chequeo, normas, estándares, metodologías, modelos etc.
- Establecer los artefactos que deberán ser generados por cada actividad a desarrollar en el proyecto: Deberá quedar constancia de los artefactos que se generan luego de realizadas las auditorías y revisiones, así como también se propone que luego de realizadas estas se cree un informe que incluya las no conformidades encontradas. Esta actividad estará asignada al revisor líder.
- Especificar y documentar los criterios a tener en cuenta durante la revisión al proyecto: La intención de esta actividad es proporcionar criterios, sobre la base de las necesidades del negocio, tales como:
- ¿Qué será evaluado?
 - ¿Cuándo o con qué frecuencia será evaluado un proceso?
 - ¿Cómo se llevará a cabo la evaluación?
 - ¿Quién debe estar involucrado en la evaluación?
- Especificar los artefactos que se revisarán, durante que etapa en el ciclo de vida del producto y por quién: Se deberá confeccionar por parte del Administrador de la Calidad un cronograma donde se especificará qué proceso, artefacto, producto o parte del mismo será revisado, en qué fecha y quiénes serán los responsables de dicha revisión.
- Especificar cómo serán organizadas estas revisiones: Debe de realizarse una estrategia por parte del administrador de la calidad en la que se explique la forma en la que serán realizadas las auditorías y revisiones.
- Se deberá elaborar el Plan de Aseguramiento de la Calidad: En la fase de Planificación, un artefacto fundamental lo constituye el Plan de Aseguramiento de la Calidad. En dicho

documento se especifican todos los datos y actividades a realizar por parte del proyecto en este sentido y el mismo debe estar contenido en el Expediente de Proyecto.

2. Aseguramiento

El aseguramiento de la calidad como actividad de protección está presente en los métodos y herramientas de análisis, diseño, programación, prueba, inspecciones técnicas formales, ajustarse a los estándares, control de la documentación del software y de los cambios realizados durante todo el proceso de desarrollo (13). Con el objetivo de asegurar la calidad del proyecto a continuación se especifican algunas herramientas que serán utilizadas durante este proceso para evaluar objetivamente los procesos y productos de mismo.

Auditorías

Se propone que las auditorías al proyecto se lleven a cabo por parte de los especialistas y siguiendo el calendario de la entidad CALISOFT.

Revisiones

Existen varios tipos de revisiones que pueden llevarse a cabo durante el proceso de desarrollo de software de estas se aplicarán en la estrategia, las que a continuación se anuncian.

Revisiones técnicas formales: Actividad desarrollada por los ingenieros del software que tiene como objetivo detectar los errores durante el proceso, siendo esto un evidente beneficio, pues se evita que se propaguen a las siguientes fases.

Revisiones Pass Around: Revisión informal en la cual el autor distribuye el documento a varias personas para su revisión. La efectividad de este método depende del conocimiento y motivación de los revisores.

Actividades que deberán ser consumadas durante la revisión.

- Realizar las revisiones de acuerdo al plan de proyecto.
- Definir el producto a revisar.
- Definir día y hora para la revisión.
- Determinar que es necesario y quien debe hacerlo.
- Definir el tipo de revisión a realizar.

- Identificar a las personas que deben participar e invitarlas indicándole su responsabilidad en la revisión.
- Si se realiza una reunión definir quien la organizará. Desarrollar una agenda de la reunión.
- Definir qué se debe hacer durante la revisión y quien debe hacerlo.
- Definir los criterios de éxito para la revisión (cuando se puede decir que la revisión puede finalizar).
- Se deben definir las listas de comprobación para dar soporte a la actividad de aseguramiento de la calidad.
- Se registran los defectos como parte del informe de la revisión entre pares y se realiza su seguimiento y escalado fuera del proyecto cuando sea necesario.

A continuación se muestra un conjunto mínimo de directrices para las revisiones técnicas formales:

1. Revisar al producto, no al productor.
2. Fijar una agenda y mantenerla.
3. Limitar el debate y las impugnaciones.
4. Enunciar áreas de problemas, pero no intentar resolver cualquier problema que se ponga de manifiesto.
5. Tomar notas escritas.
6. Limitar el número de participantes e insistir en la preparación anticipada.
7. Desarrollar una lista de chequeo para cada producto que vaya de ser revisado.
8. Disponer recursos y una agenda para las RTF.
9. Llevar a cabo un buen entrenamiento de todos los revisores.
10. Repasar las revisiones anteriores.

El nivel de escalamiento de las no conformidades se efectuará teniendo en cuenta los siguientes niveles de modo que el último nivel al cual se escalará será el nivel 1.

Nivel 1: Director General de la IP y Vicerrector de Producción.

Nivel 2: Directores de las direcciones y centros de servicio y soporte de la IP.

Nivel 3: Jefes de Grupos de las direcciones y centros de servicio y soporte de la IP.

Nivel 4: Decanos, directores de Áreas, jefes de centros de desarrollo.

Nivel 5: Gerentes, jefes de polos y líneas de producción.

Según el modelo que guía la aplicación de esta estrategia CMMI las revisiones se deberán llevar a cabo en el ciclo de vida del producto luego de realizada la fase de requerimiento, luego de realizadas las prueba internas y después de realizada la fase de soporte, pero para mantener un mayor control sobre estas se propone que se realicen revisiones al finalizar de las mismas.

Mediciones

Se debe definir un plan de mediciones para el proyecto, apoyándose en las métricas que sean establecidas en el Plan de Aseguramiento de la Calidad.

3. Control

Este proceso está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el mismo es la “Prueba del Software”.

Pruebas

El flujo de trabajo de prueba es el último dentro de los flujos básicos que define la metodología RUP, pero esto no significa que sea lo último que se haga durante el proceso de desarrollo de software. Se le realizan pruebas a un módulo de la aplicación y toda la aplicación, en dependencia de lo que se quiera probar se traza la estrategia, se escoge el nivel, el tipo y el método de prueba a utilizar. Para lograr que estas tengan éxito es necesario planificarlas y tener claro el objetivo de las mismas. Flujo de Pruebas [ver Anexo 2](#)

Algunos principios de las pruebas recogen lo siguiente (13):

- Las pruebas deberían planificarse mucho antes de que comiencen.
- No son posibles las pruebas exhaustivas.
- El número de permutaciones de camino para incluso programas pequeños es excepcionalmente grande. Por ese motivo es imposible ejecutar todas las combinaciones de caminos durante las pruebas.

Usabilidad: Prueba encaminada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente en la documentación de usuarios y materiales de entrenamiento.

Confiabilidad: La comprobación de la confiabilidad consiste en probar una aplicación para descubrir y eliminar errores antes de que se implemente el sistema.

Pruebas de Carga: Usadas para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

Pruebas de Estrés: Las pruebas de estrés consisten en la simulación de grandes cargas de trabajo para observar de qué forma se comporta la aplicación ante situaciones de uso intenso.

Pruebas de Configuración: Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.

Pruebas de Función: Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.

El proceso de prueba es llevado a cabo en varios niveles, cada uno se realiza en un determinado momento del ciclo de desarrollo del software. Se distinguen los siguientes niveles de pruebas (10):

- 1. Prueba de Desarrollador:** Es la prueba diseñada e implementada por el equipo de desarrollo.
- 2. Prueba independiente:** Es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva diferente y en un ambiente más rico que los desarrolladores
- 3. Prueba de unidad:** Es la prueba enfocada a los elementos testeables más pequeño del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera.
- 4. Prueba de integración:** Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Es el proceso de combinar y probar múltiples módulos juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se llama integración incremental cuando el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir.
- 5. Prueba de Sistema:** Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. En un ciclo iterativo estas pruebas ocurren más temprano, tan pronto como subconjuntos de comportamiento de caso de uso son implementados.

Se propone que para la aplicación a los productos que actualmente se desarrollan en SIG-MOVILES de las pruebas de carga y stress se utilice la herramienta automatizada JMETER.

Originalmente el Apache JMeter (23) fue diseñado para realizar pruebas de estrés sobre aplicaciones web (pruebas web clásicas). Sin embargo hoy en día su arquitectura ha evolucionado, ahora no sólo puede llevar a cabo pruebas en componentes típicos de Internet (HTTP), sino también puede realizar pruebas sobre Bases de Datos, scripts Perl, objetos java, servidores FTP y prácticamente cualquier medio de los que se pueden encontrar en la red. Para un óptimo desarrollo de pruebas, es necesario tener ciertas nociones funcionales de la aplicación que se va a evaluar. Si esto no es así, las pruebas no serán completas al no saber por ejemplo si ha devuelto la hoja apropiada a la petición hecha o si nos ha permitido acceder con un login no apropiado.

La propuesta de las técnicas para el control en el área de aseguramiento según la estrategia planteada, se considera como una visión general del proceso tratada a grandes rasgos. Por lo cual se propone que para la aplicación de las mismas se lleve a cabo un estudio más detallado enfocándose específicamente en el proceso de pruebas. Es objetivo de este trabajo el área o las acciones destinadas al aseguramiento de la calidad de los procesos y productos siguiendo un enfoque encaminado hacia las auditorías y revisiones como técnicas para realizar la evaluación objetiva según el modelo utilizado.

Plan de Aseguramiento de la Calidad

El plan define los atributos de calidad más importantes del producto a ser desarrollado. El mismo además debe contener:

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

- Recogida, mantenimiento y almacenamiento de datos sobre la documentación de actividades de aseguramiento de la calidad realizadas en el proyecto.

El Plan de Aseguramiento de la Calidad está considerado uno de los principales artefactos generados en el Área de Proceso Aseguramiento de la Calidad, por lo que representa un elemento fundamental para la validación de esta estrategia. [Ver Plan de Aseguramiento de la Calidad.](#)

Conclusiones del capítulo

Después de haber realizado un estudio de la situación actual del proyecto, en el cual se destacan las listas de chequeo y las entrevistas, como métodos fundamentales para la obtención de la información, se dieron a conocer los problemas que afectan el Aseguramiento de la Calidad en el mismo. A partir de este estudio se logró definir, los procesos y las actividades que debían estar presentes en la estrategia para lograr asegurar la calidad del producto y del proceso.

CAPÍTULO 3: Validación de la Estrategia

Introducción

En este capítulo se presentan los resultados obtenidos a partir de la aplicación de la estrategia de aseguramiento de la calidad, aplicada al proyecto SIG-MOVILES. La cual está encaminada asegurar la calidad de este proyecto, sin embargo, debido a la dimensión del mismo (actualmente más de 7 productos en desarrollo), solo generarán resultados a este trabajo el producto SIG-UCI basado en el protocolo WAP y el producto SIG-UCI basado en la herramienta GVSIG.

La estrategia tiene el propósito de proporcionar al equipo de trabajo y a la dirección del proyecto visibilidad objetiva de los procesos y analizar el resultado correspondiente con el fin de conocer cuan efectiva resulto ser la misma. Dentro de estos procesos se destacan la realización de actividades fundamentales como son:

- La elaboración del Plan de Aseguramiento de la Calidad.
- La reestructuración del Equipo de Calidad.
- La capacitación de los recursos humanos del proyecto.
- La evaluación objetiva de los productos utilizando el método de las revisiones.

3.1. Resultados obtenidos de la aplicación de la propuesta.

3.1.1. Definición del Equipo de Calidad.

La realización de una estrategia que asegure la calidad de los productos debe de contar con la participación de todo el recurso humano del proyecto, independientemente del personal que realizará las tareas del aseguramiento de la calidad. Por lo cual se considera como uno de los resultados significativos del trabajo la integración de los mismos.

Por lo abordado anteriormente, cabe señalar que la aplicación de esta estrategia comienza con la reestructuración de los roles vinculados a la gestión de la calidad que hasta el momento existía en el proyecto. Dado que la antigua organización no satisfacía las necesidades reales del equipo de desarrollo. Esta reestructuración dependerá en gran medida del grado en que la persona domine el contenido presentado por los cursos como parte del entrenamiento. De esta forma los estudiantes que se destaquen en el curso de auditorías y revisiones serán perfectos candidatos para ocupar el rol de revisores, de igual manera se llevará a cabo la selección del personal que desempeñará el rol de probador y diseñador de casos de prueba.

Tras haber realizado un estudio detallado de los temas relacionados con las competencias a desempeñar por cada rol, se presenta una propuesta definiendo de esta manera la nueva organización de los mismos. [Ver Plan de Aseguramiento de la Calidad sección Tareas y Responsabilidades.](#)

Luego de haber definido qué integrantes desempeñaría cada rol, se analiza lo positivo que trajo consigo esta nueva estructura:

Una de las principales ventajas de esta reestructuración consiste en que suplió o eliminó la existencia de un único rol en el equipo del proyecto correspondiente a la calidad. Esta situación contribuía a que se le asignaran excesivas actividades a un rol haciendo ineficiente su función, ya que algunas eran priorizadas y otras de igual importancia quedaban en un segundo plano. Con la nueva reestructuración se obtuvo una mayor motivación e identificación por parte del estudiante con su rol, logrando que el mismo adquiriese seguridad y experiencia en el desempeño de sus funciones.

3.1.2. Entrenamiento del equipo de desarrollo

La capacitación ha cobrado mayor importancia para el éxito de las organizaciones. Esta desempeña una función central en la alimentación y el refuerzo de estas capacidades, por lo cual se ha convertido en parte de la columna vertebral de la instrumentación de estrategias (21).

Por esta razón se adopta como uno de los resultados significativos de esta estrategia la capacitación de todo el personal del proyecto, esta actividad esta soportada por el Plan de Aseguramiento de la Calidad, [Ver Plan de Aseguramiento de la Calidad sección Entrenamiento.](#)

Debido a que el personal del proyecto poseía cierto conocimiento de los temas concernientes a la calidad de software, se decide no impartir el curso de Introducción a la Calidad, abordándose algunos temas relacionados con este en los curso de Auditoría y Revisiones, Diseño de Casos de Prueba y Prueba, considerándose estos los más prácticos dada la fase por la que transita el proyecto.

Contenido abordado en cada uno de los cursos:

- **Auditoría y Revisiones**

- Listas de chequeo.

- Criterios a tener en cuenta durante la planificación de las revisiones.

- Artefactos que se generan al efectuarse las revisiones.

- **Diseño de Casos de Prueba**

- Flujo de Trabajo de Prueba.

Casos de Uso.

Casos de Prueba.

Diseño de Pruebas a través de Casos de Uso.

- **Pruebas**

Niveles de prueba.

Tipos de prueba.

Métodos de prueba.

Técnicas de pruebas.

Estos cursos además de que contaron con la participación de todo el personal, generaron una serie de ventajas las cuales se presentan a continuación

Ventajas:

- ✓ Debido al conocimiento adquirido de las normas, procedimientos y estándares del software se redujo considerablemente el número de no conformidades detectadas.
- ✓ Se redujo el tiempo empleado en la generación de los artefactos correspondientes al flujo de trabajo de implementación y prueba.

A continuación se muestran los resultados de una encuesta realizada a 6 personas del proyecto, con el objetivo de evaluar el nivel de aceptación que tuvieron los cursos impartidos dentro del equipo, así como obtener una valoración del impacto de los mismos en vista a lograr un producto con mayor calidad.

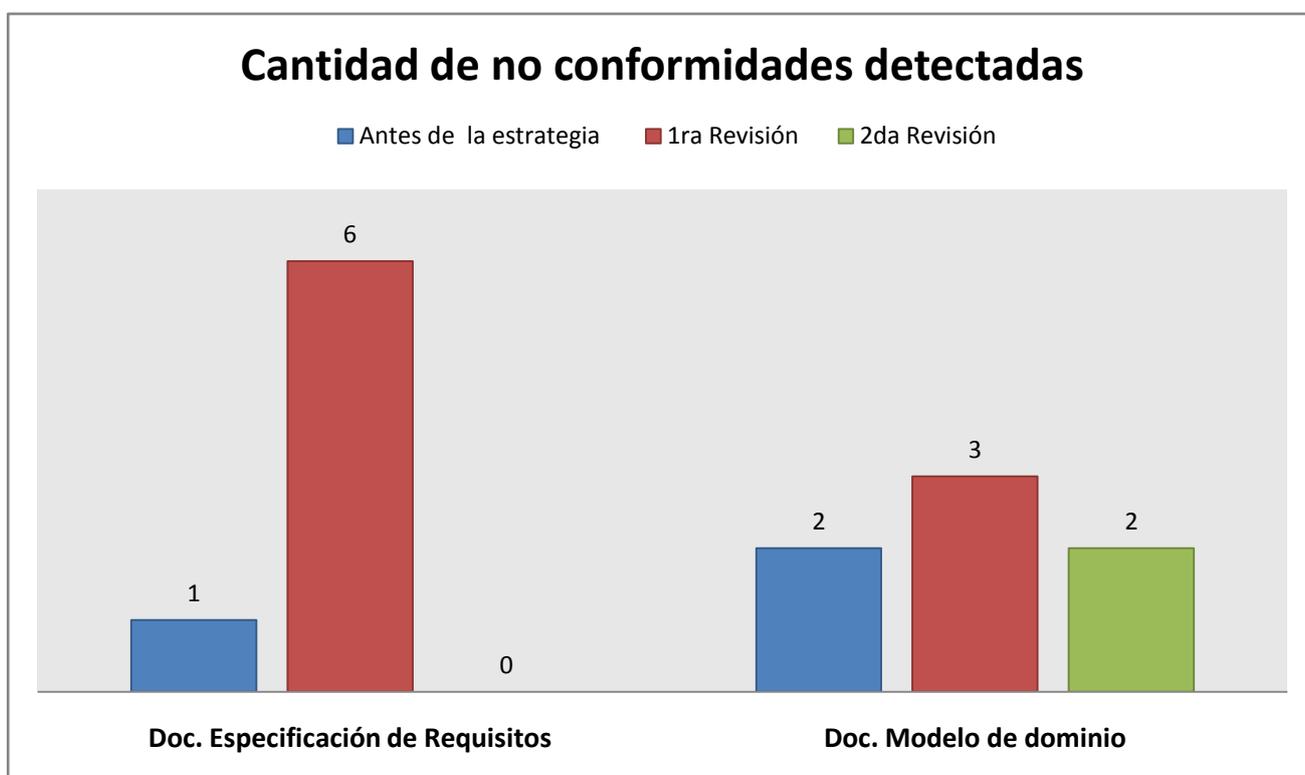
Después de realizadas las encuestas se pudo comprobar lo siguiente:

- El 100% de los encuestados le atribuyen gran importancia a estos cursos, debido a que contribuyeron a su formación como futuros ingenieros en ciencias informáticas.
- El 100% de los encuestados afirman que estos cursos fueron de gran ayuda, pues le proporcionaron en gran medida información necesaria relacionada con los artefactos correspondientes al flujo de trabajo de implementación y pruebas.

3.1.3. Resultado de la aplicación de las revisiones.

La aplicación de este método se llevó a cabo según lo previsto en el cronograma del plan de aseguramiento de la calidad. Inicialmente los artefactos o documentos revisados por parte del grupo de calidad fueron: Documento Especificación de Requisitos, Modelo del Dominio, Modelo del Sistema. En los cuales se detectaron un conjunto de no conformidades, ver ([Lista de recomendaciones SIG-UCI WAP](#) y [Lista de recomendaciones SIG-UCI GVSIG](#)).

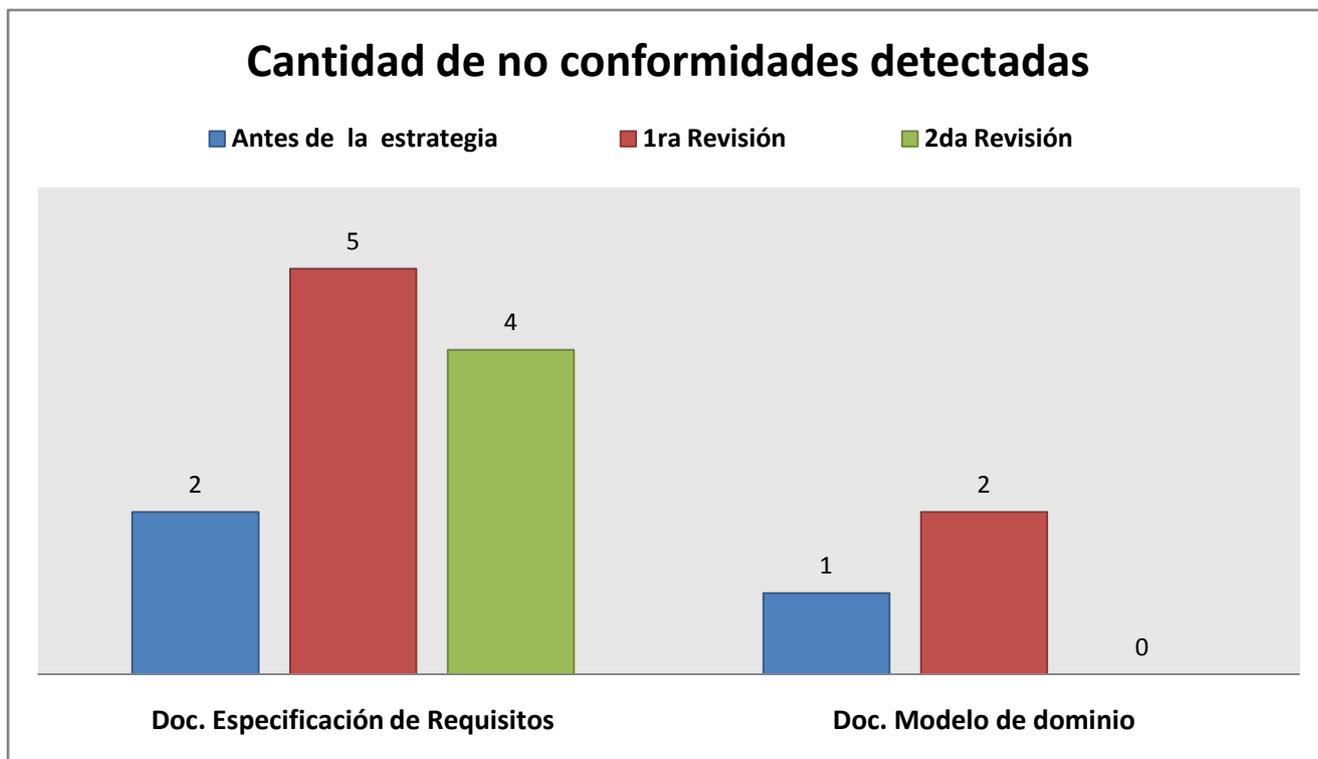
Las gráficas que a continuación se muestran, establecen una relación entre los artefactos que fueron objeto de revisión en cada uno de los proyectos y el número de no conformidades detectadas.



Gráfica1 Producto SIG-UCI GVSIG MOBILES

No conformidades reincidentes en el proyecto SIG-UCI GVSIG durante la segunda y tercera revisión.

- En el documento Modelo de Dominio no se han definido los atributos de las entidades.
- En el documento Modelo de Dominio no se han indicado todos los términos que requieren explicación.



Gráfica2 Producto SIG-UCI WAP

No conformidades reincidentes en el proyecto SIG-UCI WAP durante la segunda y tercera revisión.

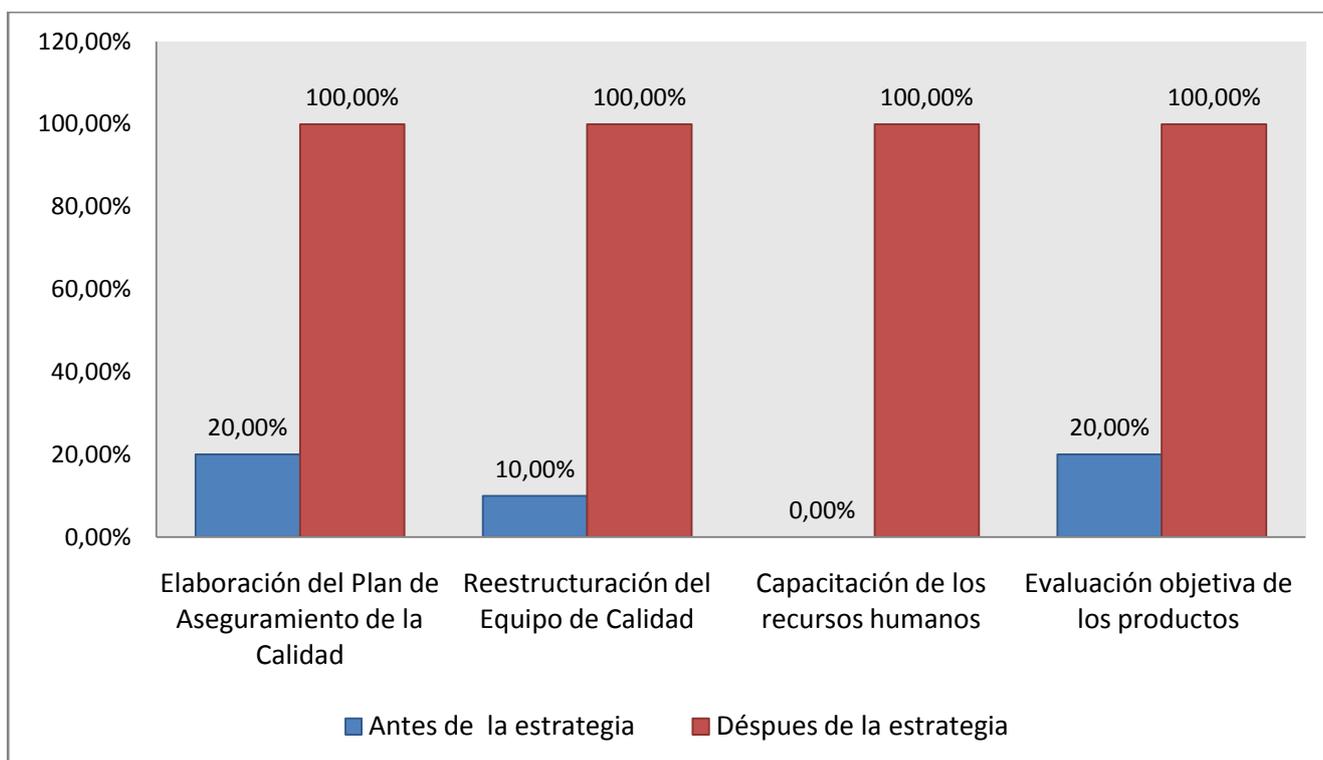
- El documento Especificación de requisitos no está acorde con la plantilla estándar del proyecto o del expediente de proyecto. Falta en el índice del documento las secciones Referencias y Eficiencia.
- En el documento Especificación de Requisitos no están bien definidas todas las interfaces que deberán ser soportadas por el sistema. (Interfaces de comunicación, Interfaces de usuarios).
- En el documento Especificación de Requisitos no existe evidencia o referencia de las normas o estándares aplicables al sistema.

Como se puede apreciar en las gráficas anteriores, el comportamiento de las no conformidades para ambos productos y en ambos artefactos es similar. Entre la 1era y 2da revisión el número de no conformidades aumenta, esto se debe a un aumento del nivel de criticidad por parte de los revisores. La realización de una segunda revisión le permitió al equipo de calidad involucrado en las revisiones

mostrar una mayor habilidad para detectar no conformidades, en correspondencia con el rol que desempeñaban.

Entre la 2da y 3ra revisión el número de no conformidades disminuye, esto se le atribuye también a la capacitación adquirida del personal desarrollador, lo cual le permitió poner en práctica todo el conocimiento adquirido en favor de lograr desarrollar los artefactos con mayor calidad. Sin embargo la existencia de no conformidades reincidentes en la 2da y 3ra revisión evidencia que no se realizó el seguimiento de las no conformidades y puesta en práctica de las acciones correctivas por parte del personal desarrollador.

En la gráfica se muestran algunas de las principales tareas que fueron definidas en la Estrategia y los resultados obtenidos al ser aplicada al producto SIG-UCI WAP y SIG-UCI GVSIG MOBILE del proyecto SIG-MOVILES.



Gráfica 3 Realización de las actividades

Conclusiones del capítulo

En este capítulo se puso en práctica la estrategia de aseguramiento de la calidad para el proyecto SIG-MOVILES la cual se llevó a cabo directamente en los productos SIG-UCI. Se expusieron los resultados arrojados por las actividades realizadas teniendo en cuenta las características de calidad. Se realizó un análisis y presentación de las principales deficiencias detectadas, dándole solución a las mismas y planificando las acciones correctivas a tener en cuenta para próximos controles.

CONCLUSIONES GENERALES

En el presente trabajo se realizó un estudio acerca de los temas necesarios para el entendimiento de la estrategia el cual permitió:

- Seleccionar el modelo de referencia y las técnicas a utilizar para llevar a cabo las evaluaciones objetivas
- Determinar la situación actual y el soporte tecnológico con que contaba el proyecto.
- Detectar los elementos que afectaban la calidad en el proyecto, a partir de los cuales se sentarían las bases para la validación de la estrategia.
- Definir una estrategia la cual a la vez permitió:
 - ✓ Especificar las actividades a desarrollar para cumplir con lo propuesto por el modelo CMMI.
 - ✓ Lograr establecer correctamente la asignación de los roles dentro del proyecto.
 - ✓ Aumentar el conocimiento del personal a partir de los cursos de capacitación impartidos.
 - ✓ Elaborar un plan de aseguramiento de la calidad, en el que se concibieron las actividades necesarias para guiar el aseguramiento de la calidad en el proyecto

RECOMENDACIONES

De acuerdo a la importancia que tiene la estrategia propuesta para lograr una calidad adecuada en el proyecto se recomienda que:

- Se realice un estudio en aras de poner en práctica en el proyecto el proceso de control definido por la estrategia.
- La estrategia propuesta, sea aplicada en todos los productos desarrollados por el proyecto SIG-MOVILES durante todo su proceso de desarrollo.

Bibliografía

1. **Ministerio de Relaciones Exteriores de la República.** Cuba MinRex. *Cuba MinRex*. [En línea] [Citado el: 21 de Noviembre de 2010.] http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm.
2. **Zulueta, Yeleny y Despaigne, Eder.** La gestión de riesgos en la producción de software y la formación de profesionales de la informática: experiencias de una universidad cubana. [En línea] 2009.
3. **Cruzata Santos, R.** *Un enfoque para la gestión del conocimiento en la universidad de las ciencias informáticas*. . CUBA : s.n.
4. **Beth Chrissis, Mary, Konrad, Mike y Shrum, Sandy.** CMMI Guía para la integración de procesos y la mejora de productos. [aut. libro] S.A. Pearson Educación. *CMMI, Guía para la integración de procesos y la mejora de productos*. España : s.n., 2009.
5. **Cueva Lovelle, Juan Manuel.** *Calidad del Software*. España : s.n., 1999.
6. **Amendola, Luis.** mailxmail.com. *mailxmail.com*. [En línea] [Citado el: 21 de 03 de 2011.] <http://www.mailxmail.com/>.
7. Mi Oruro.com. *Mi Oruro.com*. [En línea] [Citado el: 22 de Noviembre de 2010.] <http://www.mioruro.com/>.
8. AulaFacil.com. *AulaFacil.com*. [En línea] [Citado el: 21 de Noviembre de 2010.] <http://www.aulafacil.com/cursocalidad/Lecc-13.htm>.
9. Ar Consultores. [En línea] [Citado el: 21 de Noviembre de 2010.] <http://www.arconsultores.com.ar/2010/04/19/planificacion-de-la-calidad-en-un-proyecto/>.
10. **Pressman, R. S.** *Ingeniería de Software. Un enfoque práctico*. Mc Graw Hill. 2005.
11. **PERALTA, MARIO LUIS.** Calidad del Software.com. [En línea] [Citado el: 22 de Noviembre de 2010.] <http://www.calidaddelsoftware.com>.
12. ciclodevidassoftware.wikispaces.com. *ciclodevidassoftware.wikispaces.com*. [En línea] [Citado el: 10 de 03 de 2011.] <http://ciclodevidassoftware.wikispaces.com>.
13. **Neuland Agüero, D.** *Áreas del aseguramiento de la calidad*. La Habana : s.n.
14. **SCALONE, LIC. FERNANDA.** *"Estudio comparativo de los modelos y estándares de calidad del software"*. BUENOS AIRES : s.n., 2006.

15. **9000:2005, ISO.** *"Sistema de Gestión de Calida. Fundamentos y Vocabularios"*.
16. **Duque Oliva, Edison Jair.** UniVirtual. *UniVirtual*. [En línea] [Citado el: 22 de Noviembre de 2010.]
17. **PRESSMAN, R. S.** *Ingeniería de Software. Un enfoque Práctico*. 2005.
18. **Departamento de Matematicas Aplicadas y Sistemas.** Universidad Autónoma Metropolitana . *Universidad Autónoma Metropolitana* . [En línea] [Citado el: 23 de Noviembre de 2010.] <http://scma.cua.uam.mx/Documents/book/ZamudioEstandarIEEE.pdf>.
19. **Oficina Nacional de Gobierno Electrónico e Informática.** APRUEBAN DOCUMENTO "GUÍA TÉCNICA SOBRE EVALUACIÓN. *Diario Oficial "El Peruano"*. 2004.
20. **SOFTWARE.** Rational Software. [En línea] 2007 . www.rational.com.
21. **Castro, Gloria.** *PROGAMA DE CAPACITACIÓN* .
22. mitecnologico. [En línea] [Citado el: 18 de 2 de 2011.] <http://www.mitecnologico.com/Main/ConceptoDeRoles>.
23. **EJIE, S.A.** [ejie.com](http://www.ejie.com). *ejie.com*. [En línea] [Citado el: 2 de 6 de 2011.] <http://www.ejie.com>.
24. **Segoviano, Zamira y Pardo, Jose Manuel.** *InfoCali.Revista Mensual de Calisoft*. Havana : s.n., 2010.
25. **Dapena, Delgado.** *Una propuesta de métricas para un sistema de Gestión de Revisiones*. . Havana.Cuba : s.n.
26. **Sifuentes, Jesús James.** *Arquitectura para el software de aseguramiento de calidad de los proyectos de software bajo el marco CMMI: Subsistema Evaluador de Calidad de Software - Planeación y Preparación de la Evaluación*. Lima-Perú : s.n., 2008.
27. **Proyecto SIG-MOVILES.** *Plan de Desarrollo de Software*. Havana : s.n.
28. **Hernández Aguilar, Violena.** *Pruebas de Caja Negra*. [Digital] La Havana : s.n., 2009.

Trabajos citados

1. **Ministerio de Relaciones Exteriores de la República.** Cuba MinRex. *Cuba MinRex*. [En línea] [Citado el: 21 de Noviembre de 2010.] http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm.
2. **Zulueta, Yeleny y Despaigne, Eder.** La gestión de riesgos en la producción de software y la formación de profesionales de la informática: experiencias de una universidad cubana. [En línea] 2009.
3. **Cruzata Santos, R.** *Un enfoque para la gestión del conocimiento en la universidad de las ciencias informáticas*. . CUBA : s.n.
4. **Beth Chrissis, Mary, Konrad, Mike y Shrum, Sandy.** CMMI Guía para la integración de procesos y la mejora de productos. [aut. libro] S.A. Pearson Educación. *CMMI, Guía para la integración de procesos y la mejora de productos*. España : s.n., 2009.
5. **Cueva Lovelle, Juan Manuel.** *Calidad del Software*. España : s.n., 1999.
6. **Amendola, Luis.** mailxmail.com. *mailxmail.com*. [En línea] [Citado el: 21 de 03 de 2011.] <http://www.mailxmail.com/>.
7. Mi Oruro.com. *Mi Oruro.com*. [En línea] [Citado el: 22 de Noviembre de 2010.] <http://www.mioruro.com/>.
8. AulaFacil.com. *AulaFacil.com*. [En línea] [Citado el: 21 de Noviembre de 2010.] <http://www.aulafacil.com/cursocalidad/Lecc-13.htm>.
9. Ar Consultores. [En línea] [Citado el: 21 de Noviembre de 2010.] <http://www.arconsultores.com.ar/2010/04/19/planificacion-de-la-calidad-en-un-proyecto/>.
10. **Pressman, R. S.** *Ingeniería de Software. Un enfoque práctico*. Mc Graw Hill. 2005.
11. **PERALTA, MARIO LUIS.** Calidad del Software.com. [En línea] [Citado el: 22 de Noviembre de 2010.] <http://www.calidaddelsoftware.com>.
12. ciclodevidassoftware.wikispaces.com. *ciclodevidassoftware.wikispaces.com*. [En línea] [Citado el: 10 de 03 de 2011.] <http://ciclodevidassoftware.wikispaces.com>.
13. **Neuland Agüero, D.** *Áreas del aseguramiento de la calidad*. La Habana : s.n.
14. **SCALONE, LIC. FERNANDA.** "Estudio comparativo de los modelos y estándares de calidad del software". BUENOS AIRES : s.n., 2006.

15. **9000:2005, ISO.** *"Sistema de Gestión de Calida. Fundamentos y Vocabularios"*.
16. **Duque Oliva, Edison Jair.** UniVirtual. *UniVirtual*. [En línea] [Citado el: 22 de Noviembre de 2010.]
17. **PRESSMAN, R. S.** *Ingeniería de Software. Un enfoque Práctico*. 2005.
18. **Departamento de Matematicas Aplicadas y Sistemas.** Universidad Autónoma Metropolitana . *Universidad Autónoma Metropolitana* . [En línea] [Citado el: 23 de Noviembre de 2010.] <http://scma.cua.uam.mx/Documents/book/ZamudioEstandarIEEE.pdf>.
19. **Oficina Nacional de Gobierno Electrónico e Informática.** APRUEBAN DOCUMENTO "GUÍA TÉCNICA SOBRE EVALUACIÓN. *Diario Oficial "El Peruano"*. 2004.
20. **SOFTWARE.** Rational Software. [En línea] 2007 . www.rational.com.
21. **Castro, Gloria.** *PROGAMA DE CAPACITACIÓN* .
22. mitecnologico. [En línea] [Citado el: 18 de 2 de 2011.] <http://www.mitecnologico.com/Main/ConceptoDeRoles>.
23. **EJIE, S.A.** [ejie.com](http://www.ejie.com). *ejie.com*. [En línea] [Citado el: 2 de 6 de 2011.] <http://www.ejie.com>.
24. **Segoviano, Zamira y Pardo, Jose Manuel.** *InfoCali.Revista Mensual de Calisoft*. Havana : s.n., 2010.
25. **Dapena, Delgado.** *Una propuesta de métricas para un sistema de Gestión de Revisiones*. . Havana.Cuba : s.n.
26. **Sifuentes, Jesús James.** *Arquitectura para el software de aseguramiento de calidad de los proyectos de software bajo el marco CMMI: Subsistema Evaluador de Calidad de Software - Planeación y Preparación de la Evaluación*. Lima-Perú : s.n., 2008.
27. **Proyecto SIG-MOVILES.** *Plan de Desarrollo de Software*. Havana : s.n.
28. **Hernández Aguilar, Violena.** *Pruebas de Caja Negra*. [Digital] La Havana : s.n., 2009.

ANEXOS

Anexo1

Roles y Responsabilidades

Rol	Responsabilidad	Competencias
Líder de Proyecto	Guía y líder del proyecto, responsable de definir, coordinar, facilitar y comunicar las tareas dentro del equipo de proyecto.	<ul style="list-style-type: none"> • RUP. • Tecnologías de desarrollo de software. • Gestión de proyectos. • Ingeniería de Software. • Facilidades para la comunicación.
Jefe de Módulo	Dirige y Coordina el trabajo de su módulo. Define u orienta las tareas a cumplir por el resto de su equipo.	<ul style="list-style-type: none"> • RUP. • Tecnologías de desarrollo de software. • Gestión de proyectos. • Ingeniería de Software.
Analista	Encargado de elaborar el análisis del modelado del software, es quien define los artefactos que se deben realizar para el análisis del software y los construye.	<ul style="list-style-type: none"> • Análisis de sistemas. • RUP y UML. • Herramienta de modelado visual (Visual Paradigm). • Técnicas de recopilación de información.
Planificador	Encargado de elaborar el plan de proyecto y definir el conjunto de tareas que se deben realizar en cada fase del desarrollo del software, se encarga también	<ul style="list-style-type: none"> • Herramientas de planificación de proyectos. • Métricas de estimación. • Ingeniería de Software.

	del cumplimiento de las mismas, y hace un estimado del tiempo de desarrollo.	
Arquitecto de Información	Encargado de definir la arquitectura de la información, qué contenidos y la estructura en que se visualizará la información.	<ul style="list-style-type: none"> • Ingeniería de Software. • Facilidades para la comunicación.
Implementador Aplicaciones web	Encargado de la programación de los servicios y funcionalidades del software construido para la web.	<ul style="list-style-type: none"> • Herramienta de desarrollo: Netbeans 6.7). • Lenguaje de programación PHP o java. • Gestor de BD: PostgreSQL. • Framework Cartoweb. • CCS. • Javascript.
Programador de Bases de Datos	Usa el diagrama entidad-relación para generar el diseño físico de la base de datos. Participa y colabora en la codificación de la capa de acceso a datos.	<ul style="list-style-type: none"> • Herramienta de modelado visual (ErStudio). • Modelo relacional. • SQL.
Calidad	Se encarga de revisar la documentación y el software como tal, además, vela que se cumple en todo momento las normas de calidad establecidas.	<ul style="list-style-type: none"> • RUP y UML. • Calidad de Software. • Ingeniería de Software.
Gestor de	Maneja, gestiona y asegura la	<ul style="list-style-type: none"> • Subversión • Configuración de servidores.

Configuración	infraestructura, la gestión documental y la información del Repositorio del Proyecto. Realiza salvadas periódicas de la información del Repositorio.	<ul style="list-style-type: none"> • Linux.
Diseñador	Encargado de diseñar la interfaz.	<ul style="list-style-type: none"> • Photoshop. • GIMP. • CCS. • Javascript.
Arquitecto	Define y establece la línea base de la arquitectura. Establece los patrones de diseño y los estándares y restricciones de la implementación.	<ul style="list-style-type: none"> • Herramientas de desarrollo de software. Netbeans). • Lenguajes de programación. (Javascript y PHP). • Patrones de diseño. • Estilos arquitectónicos. • RUP y UML. • Bases de datos. • Protocolos de red. • Facilidades para la comunicación.

Anexo 2

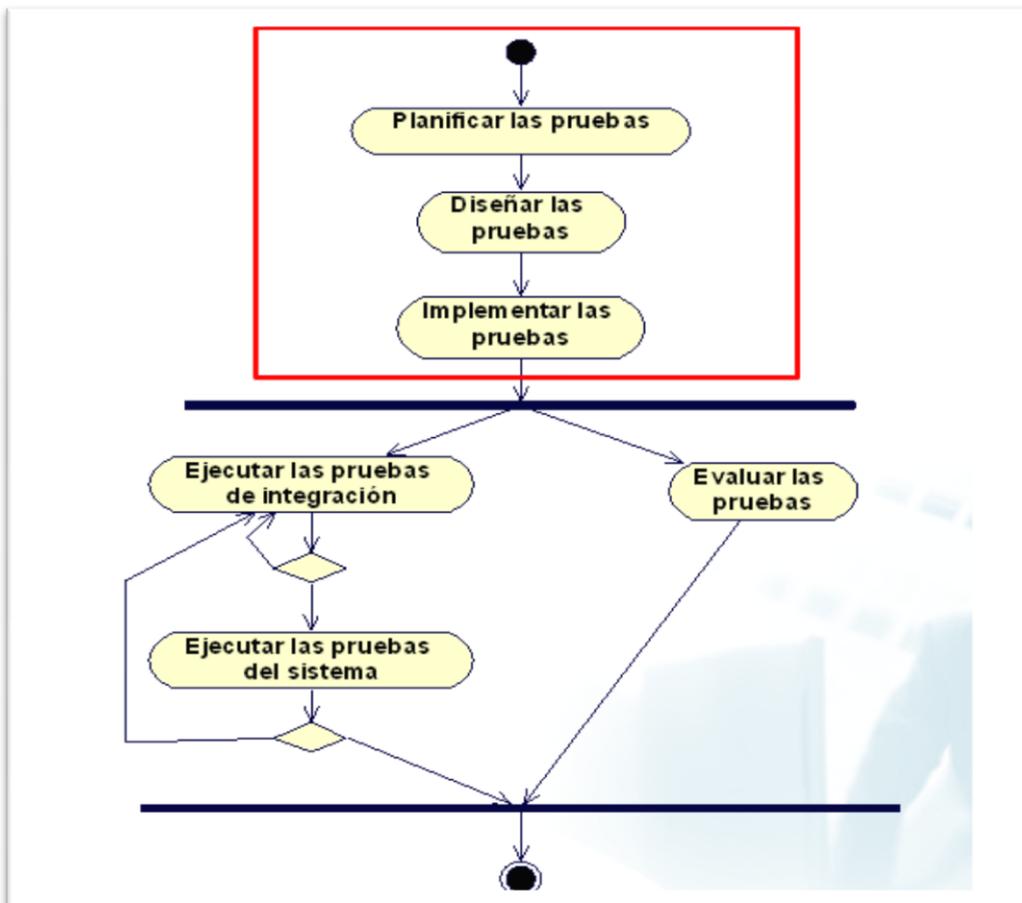
Encuesta realizada al personal del proyecto acerca de los cursos de capacitación impartidos.

1. ¿Considera que los cursos impartidos le tributan a su Trabajo de Diploma?
2. ¿Considera que los cursos impartidos contaban con la calidad requerida?
3. ¿Considera que los cursos fueron impartidos de forma tal que se entendiera perfectamente su contenido?

4. ¿Anteriormente habían recibido cursos referentes a la calidad del software?
5. Plantee brevemente su opinión respecto los cursos impartidos.
6. ¿Qué importancia le atribuye usted a estos cursos?

Anexo 3

Flujo de actividades del proceso de prueba



GLOSARIO DE TÉRMINOS

UCI: Universidad de las Ciencias Informáticas.

RTF: Revisiones Técnicas Formales.

RUP: Rational Unified Process. Proceso unificado de desarrollo.

SQA: Software Quality Assurance. Aseguramiento de la Calidad de Software.

PPQA: Garantía de la Calidad de Procesos y de Productos. Área de procesos de CMMI.

CMMI: Capability Maturity Model Integration. Es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces.

UML: Unified Modeling Language. Lenguaje Unificado de Modelado. Lenguaje de modelado de sistemas de software. **DNSP:** Dirección Nacional de Servicios Penitenciarios.

Estrategia: Plan que permite la mejor distribución de los recursos y medios disponibles a efectos de poder obtener aquellos objetivos deseados mediante una secuencia de pasos establecidos.

Artefacto: en tecnología, es un dispositivo concebido y fabricado, sea de modo artesanal o industrial, por una o más personas.

Aseguramiento de la calidad: Determina si las necesidades de los usuarios están siendo satisfechas adecuadamente.

Casos de Prueba: son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Ciclo de vida: Es un proceso por el cual los analistas de sistemas, los Ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas.

Cliente: Persona, organización o grupo de personas que encargan la construcción de un producto software.

Eficiencia: Cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

IDE: Entorno de Desarrollo Integrado.

Entorno de Desarrollo Integrado: es un programa compuesto por un conjunto de herramientas para un programador.

Evaluación de la Conformidad: demostración de que se cumplen los requisitos especificados relativos a un producto, proceso, sistema, persona u organismo.

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

Mejora continua: es una herramienta de incremento de la productividad que favorece un crecimiento estable y consistente en todos los segmentos de un proceso. La mejora continua asegura la estabilización del proceso y la posibilidad de mejora.

Metodología de desarrollo: Es una versión amplia y detallada de un ciclo de vida COMPLETO de desarrollo de sistemas que incluye: Reglas, procedimientos, métodos, herramientas, funciones individuales y en grupo por cada tarea, productos resultantes, normas de Calidad.

Módulo de software: un módulo es una parte de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará una de dichas tareas (o quizá varias en algún caso).

Procedimiento: Son los mecanismos de gestión que soportan a los métodos: El control de los proyectos, el control de la calidad.

Proceso: secuencia de actividades que tienen un marcado inicio y fin.

Producto: resultado de un proceso.

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

Requisito Funcional: define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

Requisitos: Capacidades, condiciones o cualidades que el sistema debe cumplir y tener.