



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



**Estrategia para el aseguramiento de la calidad del
“Grupo de Alternativas Libres para Multimedia”
(GALM)**

Autora: Nidia Fernández Pérez

Tutora: Ing. Anisley de la Caridad Saez Villavicencio

Ciudad de la Habana, junio 2009

Año del 50 Aniversario del Triunfo de la Revolución

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad”.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autora: Nidia Fernández Pérez

Tutora: Ing. Anisley de la Caridad Saez Villavicencio

AGRADECIMIENTOS

Si me preguntaran qué ha sido lo más difícil de mi tesis, respondería al instante que es hacer un alto y llenar esta página en blanco. Es el momento donde cierras los ojos y no sólo agradeces al que te ayudó con la tesis si no también a todos los que estuvieron desde el principio junto a ti.

A mis padres, que han sido el motor impulsor de todos estos años.

A mi hermano y mi cuñada por siempre estar ahí. A Leo y Alejandro, mis sobrinos, por ser las dos estrellas de mi cielo.

A mi abuela Patria y mi tía Nuria, por darme siempre el empujón que necesitaba.

A Fabian, por ser mi amor, mi luz y mi ejemplo. Gracias por compartir mis alegrías y por secar mis lágrimas. Gracias por ser lo más especial de mi vida.

A mis suegros, Mary y Felito, por acogerme como una hija y brindarme todo su amor.

A Mailyn, mi amiga de antes, de ahora y de siempre.

A mi tutora, por toda la ayuda y las horas que me dedicó.

A mis mejores compañeras de cuarto, Anisley, Mery y Lizy, por la amistad y la complicidad.

A Jorge, José, Eric y Humberto, por acogerme y ayudarme cuando Fabi no estuvo.

A la gente del grupo, que me enseñaron a compartir y ser junto a ellos un batallón de gente queriéndose.

A la gente del proyecto, Roberto, Yonnys y todos los demás por brindarme su apoyo.

A todos gracias.

Nidia.

DEDICATORIA

*A mi mamá,
por hacer realidad sus sueños y los míos.*

RESUMEN

La calidad del software es un tema que debe ser tratado de manera muy específica en cada proyecto teniendo en cuenta sus características, entorno de desarrollo y requerimientos o funcionalidades planteadas por el cliente. En aras de obtener un producto con la calidad requerida, se impone conocer a fondo el proceso de desarrollo del software y a partir de ahí trazar una estrategia que asegure y controle la calidad durante el ciclo de vida del proyecto.

En este sentido, con el objetivo de diseñar una Estrategia de aseguramiento de la calidad para el Grupo de Alternativas Libres para Multimedia se realizó un análisis de la situación actual de uno de los productos en desarrollo, arrojando la existencia de problemas críticos con respecto a la calidad del mismo. Con estos resultados se procedió al diseño de la Estrategia que propone actividades de aseguramiento y control de la calidad para todas las etapas del proceso de desarrollo del software que se construye en el proyecto. Dicha Estrategia se ha aplicado hasta la etapa de desarrollo actual del producto y los resultados obtenidos se registran en esta investigación.

PALABRAS CLAVES

Aseguramiento, Calidad, Estrategia

ÍNDICE

INTRODUCCIÓN	9
CAPÍTULO 1. Fundamentación Teórica.	13
1.1. Introducción.....	13
1.2. Concepto de calidad	13
1.3. Antecedentes de la calidad de software. Enfoques. Definición.	14
1.4. Factores que determinan la calidad del software	16
1.5. Gestión de la calidad del software	18
1.5.1. Planificación de la calidad del software.....	18
1.5.2. Control de la calidad del software.....	19
1.5.3. Aseguramiento de la calidad	19
1.5.4. Mejora de la calidad del software	21
1.6. Estándares para el aseguramiento de la calidad	21
1.7. Modelos para el aseguramiento de la calidad.....	29
1.8. Proceso de pruebas	32
1.9. Situación actual de la calidad del software.	37
1.10. Calidad del software en la UCI.....	39
1.11. Grupo de Alternativas Libres para Multimedia	41
1.12. Conclusiones del Capítulo.....	42
CAPÍTULO 2. Proceso de Aseguramiento de la Calidad.	44
2.1. Introducción.....	44
2.2. Descripción del Grupo de Alternativas Libres para Multimedia.....	44
2.3. Fundamentación de la metodología a utilizar	47
2.4. Proceso de evaluación del software	54
2.4.1. Identificación de las etapas del proceso de pruebas.....	54
2.4.2. Propuesta de las métricas. Instrumento de medición: Las Listas de Chequeo.....	57
2.4.3. El diseño y registro de Casos de Prueba de Aceptación:	58
2.4.4. Establecimiento de pautas para procesar los resultados.	59

2.4.5.	Responsabilidades dentro del proceso de pruebas.....	62
2.4.6.	Diseño final de la estrategia de prueba.	63
2.5.	Aseguramiento de la calidad	64
2.5.1.	Plan de Aseguramiento de la Calidad	64
2.5.2.	Verificación y validación del proceso de aseguramiento de la calidad.....	65
2.6.	Gestión de configuración del software.....	68
2.7.	Auditorías de calidad al software	69
2.8.	Herramientas y técnicas a utilizar. Propuesta de acciones correctivas.	71
2.9.	Conclusiones del capítulo	73
CAPÍTULO 3. Validación de la Estrategia de Aseguramiento.		75
3.1.	Introducción.....	75
3.2.	Resultados obtenidos de la aplicación de la Estrategia en el proyecto	76
3.2.1.	Resultado de la aplicación de auditorías	76
3.2.2.	Resultado de las pruebas.....	84
3.3.	Análisis de los resultados.....	90
3.4.	Conclusiones del capítulo	92
CONCLUSIONES.....		93
RECOMENDACIONES		94
REFERENCIAS BIBLIOGRÁFICAS.....		95
BIBLIOGRAFÍA		98
GLOSARIO DE TÉRMINOS Y DEFINICIONES		99
ANEXOS.....		104

INTRODUCCIÓN

Ante la incesante revolución tecnológica actual, donde cada vez son mayores y más ambiciosas las pretensiones de la sociedad, se hace inminente la garantía de la calidad de los productos. A medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables crece en los productores un interés constante por la búsqueda de la eficiencia.

La informática, como ciencia de avanzada del siglo XXI, no está exenta de estas exigencias, por lo que, en materia de software, se ha trazado una serie de pautas para avalar el grado con que los sistemas cumplen los requerimientos especificados y las expectativas del cliente. Para ello, como parte de las características implícitas que se espera de toda aplicación profesional, se ha establecido una concordancia entre los requisitos funcionales y de rendimientos establecidos y los estándares de desarrollo.

La calidad de los procesos de software se está convirtiendo en un elemento estratégico de las grandes organizaciones debido a su fuerte impacto en la competitividad de las empresas y, a pesar de que pueda parecer un concepto alejado de la vida diaria de la mayoría de las personas, los fallos de software afectan a todos los sectores y a todos los países.

Cuando en el ordenador aparece un mensaje de error o una pantalla azul, estamos ante un problema de calidad del software; cuando un fallo en el sistema de gestión aeroportuaria provoca retrasos, pérdidas de maletas o inutiliza las pantallas de información, estamos ante un problema de calidad del software; cuando en un banco monetario se bloquean las terminales de cobro y anotación de pedidos, estamos ante un problema de calidad del software.

La calidad de un producto no es algo que se añade al final como si se pintara de un color su exterior, es algo que se cuida a lo largo de todo el proceso de construcción del mismo. En el software esto es especialmente cierto, ya que es un resultado básicamente intelectual dependiente del trabajo de los profesionales especializados.

La Universidad de las Ciencias Informáticas (UCI), siendo una de las principales entidades productoras de software de Cuba, ha tenido que desempeñar una nueva estrategia para garantizar la fiabilidad de sus productos y así asegurar su entrada al mercado de software a nivel mundial. Uno de los pilares de dicha estrategia, es la creación de la Dirección de Calidad de la universidad como rectora de todo proceso y

que, a su vez, centraliza el trabajo de los grupos de calidad. Al mismo tiempo, el aseguramiento de la calidad de cada proyecto desarrollado en la UCI está respaldado por uno de estos grupos.

Como parte de esos proyectos a los cuales la Dirección de Calidad debe respaldar, se encuentra el perteneciente al Grupo de Alternativas Libres para Multimedia (GALM), un entorno de desarrollo integrado (IDE por sus siglas en inglés) elaborado como una solución libre para la producción de software con tecnología multimedia.

La **situación problémica** reside en que actualmente en el equipo de trabajo existe demasiado reproceso o retrabajo, las funciones dentro del proyecto no operan correctamente, la comunicación es poco fluida entre sus miembros derivado de una mala documentación en los procesos y sus resultados. Además existe un elevado número de quejas por parte del cliente luego de la entrega, y el impacto de estos conflictos recae directamente sobre la imagen del proyecto.

Sumado a esto, en la UCI se realiza el flujo de trabajo de pruebas así como la evaluación de la conformidad después de terminado el producto; esto conlleva a la detección tardía de problemas que pudieron haberse evitado con anterioridad, por lo que surge un atraso considerable en el cronograma de trabajo planificado y se obtiene como resultado un cliente insatisfecho. De allí que se defina el siguiente **problema a resolver** ¿cómo obtener periódicamente un diagnóstico que brinde la medida en que avanza el proceso de desarrollo de software?

Se pasa a considerar entonces como **objeto de estudio** el proceso de aseguramiento de la calidad en el desarrollo de aplicaciones multimedia y Software Educativo en la UCI, delimitándose como **campo de acción** el aseguramiento de la calidad del “Grupo de Alternativas Libres para Multimedia” (GALM). Para dar solución al problema planteado se define como **objetivo general** establecer para el proyecto una estrategia de aseguramiento de la calidad que permita obtener una media cuantitativa de su cumplimiento y que guíe en lo adelante la calidad en el proceso de desarrollo de software.

A raíz de lo cual se definen los siguientes **objetivos específicos**:

1. Definir el proceso de aseguramiento de la calidad.
2. Crear la propuesta de solución.
3. Generar los artefactos correspondientes al proceso de aseguramiento.

4. Obtener una primera evaluación de GALM.

Para resolver el problema planteado se propone la siguiente **idea a defender**: La elaboración de una buena Estrategia de aseguramiento de la calidad permitirá la obtención de un diagnóstico que mida el avance del proceso de desarrollo del software y garantizará la calidad del producto final.

Para el desarrollo de la investigación se cuenta con una serie de **tareas**:

- Revisión bibliográfica del tema.
- Estudio de otras soluciones similares.
- Análisis de la metodología para la gestión de configuración del software.
- Realización de un control de la documentación mediante las revisiones al software.
- Definición de las métricas a utilizar de acuerdo al tipo de proyecto.
- Definición de cada uno de los procesos de la propuesta de solución.
- Confección de un Plan de Aseguramiento de la Calidad.
- Creación de un Plan de Pruebas.
- Elaboración de las Listas de Chequeo.
- Aplicación de parte del Plan de Pruebas.
- Selección de las técnicas para el diagnóstico del software.
- Elaboración de un resumen a partir de los resultados de las pruebas.
- Elaboración de una propuesta de aseguramiento de la calidad para el proceso de desarrollo.

Fundamentación Teórica



Fundamentación Teórica

1.1. Introducción

Propiciar la calidad en el software es una actividad que ha surgido como consecuencia de la fuerte demanda de sistemas de software en todos los procesos que se desarrollan en la actualidad. Por lo que se hace necesario garantizar la fiabilidad de sus productos y satisfacer las expectativas del cliente. En este capítulo se realiza un profundo análisis referente al concepto de calidad teniendo en cuenta sus antecedentes pero enfocado a la calidad de un producto software.

Además se lleva a cabo un estudio acerca del estado actual de la calidad del software y los factores que la determinan. Adicionalmente se abordan la Gestión de la calidad del software, los modelos y estándares de evaluación, así como lo referente al proceso de pruebas.

1.2. Concepto de calidad

Un profundo conocimiento de la calidad puede conducir a una mejor comprensión de su actual importancia. Hoy en día se considera que esta es una de las variables clave en la determinación de los objetivos estratégicos de cualquier empresa que desee permanecer en el entorno competitivo. *La calidad es un factor de diferenciación, que ha de estar contenido en los objetivos estratégicos de la empresa* (Benturia, 1998).

Se considera que una estrategia basada en la calidad supone una mejora en la posición de costes, cumplimiento en las entregas, en los plazos respecto al mercado, o la capacidad de respuesta ante los cambios de éste (Fortuna, 1990). A lo largo de la historia, el éxito de cualquier empresa ha requerido la predicción de las necesidades de las personas que podían convertirse en clientes, y el esfuerzo por

abastecer las necesidades a unos precios que los clientes estuvieran dispuestos a pagar. No obstante, la competencia, actualmente, se plantea a nivel mundial y por ello la calidad es especialmente necesaria. El cliente se defiende a sí mismo, explora el planeta en busca de buena calidad y buen precio.

El concepto actual de calidad ha evolucionado hasta convertirse en una forma de gestión que introduce un concepto de mejora continua en cualquier organización y a todos los niveles de la misma, y que afecta a todas las personas y a todos los procesos involucrados en ella. En el presente documento se relacionan algunos conceptos y definiciones de personalidades e instituciones, todos considerados de relevancia en la literatura del tema que se investiga.

1.3. Antecedentes de la calidad de software. Enfoques. Definición.

Los aspectos de la calidad del software son características propias del software, aquellas que usted quiere controlar y asegurar. El software es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla. El software puede tener errores, incidencias, pero no son similares a lo que cualquier equipo de carácter físico. Es preciso destacar que la calidad del software se encuentra a la misma medida con la calidad tradicional, pero un paso atrás, debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad del software tiene de 50 a 60 años.

La calidad ha evolucionado de acuerdo a varias etapas. *La primera etapa inicia con la fabricación artesanal en la época preindustrial, la misma podía considerarse como algo similar a las obras de arte* (Gutierrez, 1989). El artesano ponía todo su empeño en hacer sus obras, dado que de la perfección dependía su prestigio artesanal. Por consiguiente, el control de la calidad era prácticamente innecesario.

Una segunda etapa puede considerarse que abarca hasta el final del siglo XIX, conocida como fabricación industrial. *El trabajador tenía la responsabilidad de la fabricación completa del producto y, por tanto, cada trabajador podría controlar totalmente su calidad de trabajo* (Feigenbaum, 1951).

La producción masiva trajo consigo cambios en la organización de la empresa. Surge formalmente la inspección constituyendo así una tercera etapa. Los operarios no se dedicaban a la elaboración de un artículo de principio a fin. Es en esta época cuando Frederick W. Taylor expuso su teoría para perfeccionar el trabajo dentro de las organizaciones. Su sistema, denominado Administración Científica, proponía planificar, supervisar y controlar el trabajo detalladamente si se quería lograr que estuviera hecho correctamente.

Hacia los años cuarenta, la producción en masa había aumentado tanto que hacía imposible la inspección al cien por cien. Es entonces cuando W. Shewhart difunde en Estados Unidos la aplicación de los métodos estadísticos al campo del control de la calidad; definiéndose concretamente una cuarta etapa. A este grupo de investigadores pertenecieron Deming y Juran. Estos autores junto con Feigenbaum son considerados como los pioneros de la Gestión de la Calidad Total.

Más tarde surge una quinta etapa que se caracteriza por la necesidad de que quedara asegurado el mejoramiento de calidad logrado. *Para ello había que formar profesionales en este terreno, lo cual suponía una partida presupuestaria dedicada a atender el programa de calidad* (Gutierrez, 1989). En estos momentos se crean normas técnicas que defienden al consumidor y ayudan tanto al cliente como al proveedor, suponiendo un gran avance en la calidad.

Durante los años 60-70 en las empresas se crean departamentos de ingeniería de calidad, de fiabilidad y de procesos. Esta sexta etapa se conoce por la creación de los llamados círculos de calidad donde a partir del aseguramiento de la calidad aparecen las auditorías de calidad, para comprobar que efectivamente se han cumplido las normas, y el manual de calidad, donde se recoge todo lo relativo a la calidad en la empresa.

Por Calidad Total (Total Quality Control), séptima etapa, se entiende exactamente lo que la expresión señala: cero defectos en los productos que salen de la fábrica y en los servicios que se ofrecen. Significa calidad en todos los aspectos de las operaciones de la empresa. La Calidad Total es simplemente la actitud permanente orientada a la mejora continua. Este concepto aparece en 1961, con Feigenbaum, en los Estados Unidos. Se franquea una etapa importante en la calidad viéndose afectados todos los departamentos de la empresa.

La calidad es responsabilidad de todos y cada uno de los que intervienen en cada etapa del proceso (Feigenbaum, 1991). Estrategia de la Calidad Total es el término más evolucionado dentro de las sucesivas transformaciones que se han venido desarrollando a lo largo del tiempo y que engloba ahora todo un sistema de gestión empresarial íntimamente relacionado con el concepto de mejora continua.

Toda empresa o equipo de desarrollo de software debe adoptar un proceso de desarrollo, siempre el que más le convenga. Hay una gran variedad de procesos de donde tomar los elementos más convenientes para alinear los desarrollos con algunas características a tener en cuenta para la calidad del software.

- *Desarrollar una coherencia desde el principio de cada proyecto. En ese momento deben definirse, cuantificarse y/o especificarse las características de calidad a cumplirse en ese producto.*
- *Poseer las herramientas necesarias que ayuden al equipo a llevar adelante todas las tareas requeridas para alcanzar los objetivos de calidad planteados.*
- *Disponer de personas preparadas técnicamente y liderados por al menos un profesional con experiencia, que formen un equipo con la capacidad de adaptarse y mejorar continuamente (Main, 2009).*

Tomar por el camino del desarrollo de software de calidad no significa disponer de grandes inversiones, sino de alinear los recursos disponibles, prepararlos y coordinarlos adecuadamente. Llegado el momento de escalar, o desear el logro de alguna certificación para ampliar mercados, o sencillamente buscar ser una empresa que logre desarrollar productos de calidad, será mucho mejor y más simple, si las empresas siguieran estos lineamientos mínimos.

1.4. Factores que determinan la calidad del software

Con el transcurso del tiempo, las empresas se han visto obligadas a atender mercados más complejos y exigentes, a hacer frente a una competencia más dura y a suministrar productos con un período de vida útil menor. La calidad del software se ha visto sometida a una serie de modificaciones conceptuales necesarias. Éstas han sido motivadas parcialmente por los cambios de cultura en la empresa, y acordes con las exigencias del mercado en cada momento.

Los factores que determinan la calidad del software pueden clasificarse en dos grandes grupos:

- Factores que pueden ser medidos directamente
- Factores que solo pueden ser medidos indirectamente.

Consecuentemente en el modelo de McCall para la calidad de la revisión, se añade que para lograr determinar la calidad es necesario centrarse en tres aspectos fundamentales en un producto software:

- Operaciones del producto: características operativas
 - ✓ Corrección (¿Hace lo que se le pide?) El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente

- ✓ Fiabilidad (¿Lo hace de forma fiable todo el tiempo?) El grado que se puede esperar que una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida
- ✓ Eficiencia (¿Qué recursos hardware y software necesita?) La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados
- ✓ Integridad (¿Puedo controlar su uso?) El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
- ✓ Facilidad de uso (¿Es fácil y cómodo de manejar?) El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.
- Revisión del producto: capacidad para soportar cambios
 - ✓ Facilidad de mantenimiento (¿Puedo localizar los fallos?) El esfuerzo requerido para localizar y reparar errores.
 - ✓ Flexibilidad (¿Puedo añadir nuevas opciones?) El esfuerzo requerido para modificar una aplicación en funcionamiento.
 - ✓ Facilidad de prueba (¿Puedo probar todas las opciones?) El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.
- Transición del producto: adaptabilidad a nuevos entornos
 - ✓ Portabilidad (¿Podré usarlo en otra máquina?) El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
 - ✓ Reusabilidad (¿Podré utilizar alguna parte del software en otra aplicación?) Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
 - ✓ Interoperabilidad (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?) El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos (Pressman, 1998).

Los anteriores tópicos resumen un pequeño conjunto de elementos básicos para orientar la producción de software, mas no garantizan con su aplicación productos de alta calidad, pues se trata de un pequeño grupo de elementos para empezar. Estos lineamientos deben ser ajustados a cada caso particular, aplicables a cualquier proyecto de desarrollo de software, permitiendo la orientación y organización del trabajo dentro y fuera del proyecto.

1.5. Gestión de la calidad del software

La Gestión de la calidad del software es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. Se basa en la determinación y aplicación de las políticas de calidad de la empresa. La Gestión o Administración de la Calidad se aplica normalmente a nivel de empresa o dentro de la gestión de cada proyecto. El propósito de la Gestión de la calidad del software es entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas.

1.5.1. Planificación de la calidad del software

La planificación es la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo, de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado. El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias.

La planificación de la calidad debería ser revisada de nuevo junto con el progreso del diseño y desarrollo, y los elementos, en cada fase, deberían ser completamente definidos al comienzo de dicha fase (IEEE, 1990) Los factores que determinan el Modelo o Estándar de calidad del software a elegir son:

1. La complejidad del proceso de diseño.
2. La madurez del diseño.
3. La complejidad del proceso de producción.
4. Las características del producto o servicio.
5. La seguridad del producto o servicio.
6. Económico.

Los aspectos a considerar en la Planificación de la calidad del software son: Modelos/Estándares de Configuración de software a utilizar, Costos de la Configuración de software, Recursos humanos y materiales necesarios.

El plan de calidad define los atributos de calidad más importantes del producto a ser desarrollado. En la Planificación de la calidad del software se debe determinar: (1) Rol de la Planificación, (2) Requerimientos

de la Configuración de software, (3) Preparación de un Plan de Configuración de software, (4) Implementación de un Plan de Configuración de software y (5) Preparar un Manual de calidad.

La planificación de la calidad facilita el modo de adaptar la planificación del sistema de gestión de la calidad a un proyecto específico, producto o contrato. La planificación de la calidad puede incluir referencias genéricas y/o proyecto / producto / contrato específico de procedimientos, como apropiados.

1.5.2. Control de la calidad del software.

El Control de la calidad del software son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en 2 objetivos fundamentales: (1) mantener bajo control un proceso y (2) eliminar las causas de los defectos en las diferentes fases del ciclo de vida (Cueva, 1999). Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la calidad del software es la “Prueba del software”.

El control de la calidad es una sucesión de inspecciones y pruebas realizadas en el transcurso del proceso del software para cerciorarse de que el producto cumple con los requerimientos establecidos. Cuenta con un bucle de realimentación conocido como el feedback del proceso que concibió al producto. *La realimentación y la medición fusionadas posibilitan la mejora del proceso cuando existe una falla de los productos de trabajos elaborados al efectuar sus especificaciones. Este enfoque ve el control de la calidad como parte del proceso de fabricación (Pressman, 1998).*

Durante el proceso de desarrollo de un producto software se deben planificar, organizar, dirigir y controlar una serie de actividades sobre las cuales se debe tener cierto control de la calidad con que se realizan las mismas, para de esta forma tratar de garantizar que se obtenga un producto con la calidad requerida y que se cumplan los objetivos previstos desde el inicio. Se debe tener en cuenta que según sea el control de la calidad durante el desarrollo del producto, mayor será la garantía que se tenga de la calidad del software al finalizar el mismo.

1.5.3. Aseguramiento de la calidad

El aseguramiento de la calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el software satisfará los requisitos de calidad ya establecidos. Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después. El aseguramiento de la calidad del software engloba: (1) un enfoque de Gestión de Calidad, (2) métodos y herramientas de Ingeniería del Software, (3) revisiones técnicas formales aplicables en el proceso de software, (4) una estrategia de prueba multiescala, (5) el control de la documentación del software y de los cambios realizados, (6) procedimientos para ajustarse a los estándares de desarrollo del software y (7) mecanismos de medición y de generación de informes.

Este aseguramiento tiene asociado 2 constitutivos diferentes: los Ingenieros de software que realizan el trabajo técnico y un grupo de SQA (Software Quality Assurance) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

Las Actividades del grupo de SQA son: (1) Establecimiento de un plan de SQA para un proyecto, (2) Participación en el desarrollo de la descripción del proceso de software del proyecto, (3) Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido, (4) Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software, (5) Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido, y (6) Registrar lo que no se ajuste a los requisitos e informar a sus superiores (Pressman, 1998).

Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y; ayuda a recopilar y analizar las métricas del software. Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable. Cuando se habla de software se refiere a la disciplina de recopilar y analizar datos basándose en mediciones reales de software, así como a las escalas de medición.

Las medidas de calidad del software deben comenzar desde la especificación y terminar con la implementación, implantación y mantenimiento o post-implantación. Debe aplicarse a lo largo de todo el proceso de Ingeniería de Software. Básicamente, la medición es una fase normal de cualquier actividad industrial. Sin mediciones es imposible perseguir objetivos comerciales normales de una manera racional.

1.5.4. Mejora de la calidad del software

La Mejora de la calidad del software es la parte de la Gestión de la Calidad que contribuye, por medio de las mediciones, a los análisis de los datos y auditorías, a efectuar mejoras en la calidad del software.

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

Los resultados de la auditoría son documentados y remitidos al director de la organización auditada, a la entidad auditora, y cualquier organización externa identificada en el plan de auditoría. El informe incluye la lista de elementos no conformes u otros aspectos para las posteriores revisiones y acciones. Cuando se realiza el plan de auditoría, las recomendaciones son informadas e incluidas en los resultados de la auditoría.

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

La calidad del software es resultado del movimiento global dentro del proceso de mejoramiento continuo de los modelos y/o estándares de producción en todos los sectores industriales, en particular, cuando éste se concentra en la producción de sistemas de información y software especializado.

1.6. Estándares para el aseguramiento de la calidad

Dado que la competencia cada día es más fuerte, es necesario que las empresas se preocupen en dar un mejor producto. Pero la calidad de un producto no solo se mide al terminarlo. La complejidad de los problemas que hoy en día buscan una solución en el software ha aumentado de manera considerable. Pero este crecimiento ha sobrepasado de sobremanera al aumento en la habilidad de desarrollar y mantener el software por parte de las organizaciones dedicadas a desarrollarlo o mantenerlo.

Las organizaciones desarrolladoras de software han estudiado y definido un conjunto de modelos y estándares que han marcado el camino para la producción de software de calidad. En los últimos años los estándares de calidad internacionales se han reorientado hacia el aseguramiento y Gestión de la calidad.

Esto ha dado paso a la aparición de importantes modelos de referencia, que tienen en común la evaluación de la capacidad de los procesos en niveles de desarrollo o madurez en la industria del software. Al compás de esto, las Normas de Calidad se han ido perfeccionando con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores que desarrollan productos de software.

Estándar Internacional (ISO/IEC 9126)

Un estándar es *un conjunto de acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados constantemente, como reglas, lineamientos o definiciones de características. Todo esto con la finalidad de asegurar que los materiales, productos, procesos y servicios son óptimos para su propósito (ISO/IEC 9126-1).*

ISO 9126 es un estándar internacional para la evaluación del software. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica 6 atributos clave de calidad:

- Funcionalidad – El grado en que el software satisface las necesidades indicadas por los siguientes subatributos:
 - ✓ Idoneidad
 - ✓ Corrección
 - ✓ Interoperabilidad
 - ✓ Conformidad
 - ✓ Seguridad

- Fiabilidad – Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos:
 - ✓ Madurez
 - ✓ Tolerancia a fallos
 - ✓ Facilidad de recuperación
- Usabilidad – Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos:
 - ✓ Facilidad de comprensión
 - ✓ Facilidad de aprendizaje
 - ✓ Operatividad
- Eficiencia – Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos:
 - ✓ Tiempo de uso
 - ✓ Recursos utilizados
- ✓ Mantenibilidad – Facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos:
 - ✓ Facilidad de análisis
 - ✓ Facilidad de cambio
 - ✓ Estabilidad
 - ✓ Facilidad de prueba
- Portabilidad – La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos:
 - ✓ Facilidad de instalación
 - ✓ Facilidad de ajuste
 - ✓ Facilidad de adaptación al cambio

El atributo Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y como resultado, la noción del usuario se amplía, tanto a operadores como a programadores, los cuales son usuarios de componentes como bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

La NC ISO/IEC 9126 define los tipos de métricas existentes para evaluar la calidad del producto software, tales como:

- *Métricas internas: son aquellas que no dependen de la ejecución del software (medidas estáticas).*
- *Métricas externas: son aquellas aplicables al software en ejecución (ISO/IEC 9126-1).*

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales. Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso. Ver la figura 1 que se muestra a continuación.

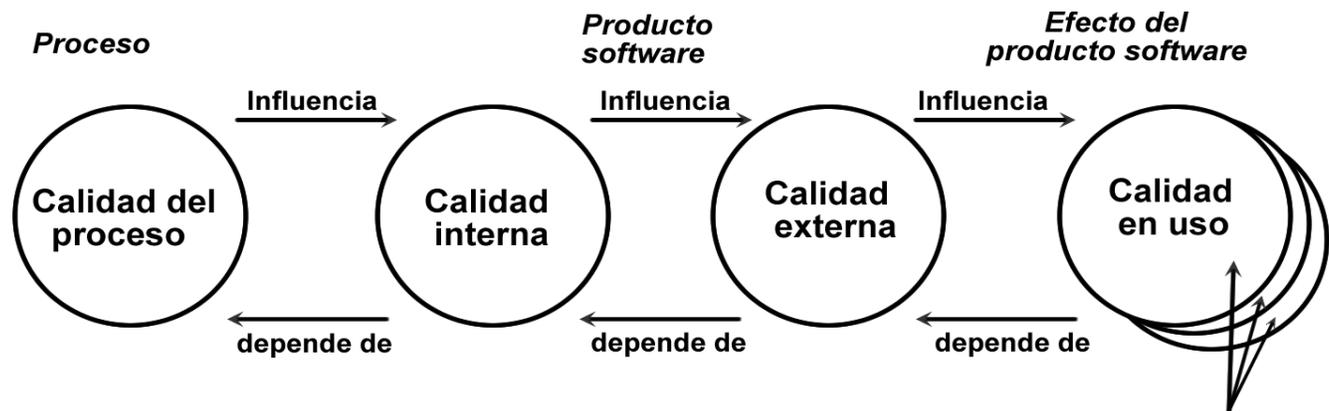


Figura 1. Métricas de Calidad.

Fuente: (ISO/IEC 9126-1).

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de características de calidad:

- Factores (especificar): Ellos describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Ellos describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Ellas son definidas y usadas para proveer una escala y método para la medida.

ISO 9126 distingue entre fallos y no conformidad, siendo un fallo el no cumplimiento de los requisitos previos, mientras que la no conformidad afecta a los requisitos especificados. Una distinción similar es hecha entre la validación y la verificación.

Este estándar está pensado para los desarrolladores, adquirentes, personal que asegure la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software. Por tanto, puede servir para validar la totalidad de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, así como criterios de aseguramiento de la calidad.

La calidad de cualquier proceso del ciclo de vida del software influye en la calidad del producto software que, a su vez, contribuye a mejorar la calidad en el uso del producto. La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta) (ISO/IEC 9126-1).

Determinación de Capacidad de Mejora para Procesos Software (SPICE)

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

Componentes de SPICE:

- Conceptos y Guía de Introducción.
- Modelo de Referencia para Procesos y Capacidad.

- Realización de una Evaluación.
- Guía de Evaluación.
- Modelo de Evaluación y Guía de uso.
- Guía de calificación de Evaluadores.
- Guía de uso para la Mejora de Procesos.
- Guía para Determinación de Capacidad de Proveedores
- Vocabulario.

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

Cliente-Proveedor (CUS)

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

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

Ingeniería (ENG)

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

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

Soporte (SUP)

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

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

Gestión (MAN)

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

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

Organización (ORG)

En esta categoría se analizan los procesos que establecen los objetivos de negocio de la empresa, estos son:

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

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

Nivel 0, Incompleto: Es fracaso general el tratar de utilizar las prácticas bases a los procesos. Ya que no es fácil identificar las salidas de los procesos o el trabajo de los productos, o sea, no hay evidencia del logro de los resultados esperados.

Nivel 1, Realizado: Las prácticas de los procesos son ejecutados generalmente. La ejecución de las prácticas dependerá del conocimiento y esfuerzo personal. Se identifica algunos procesos. El proceso es comprendido y los productos son realizados correctamente.

Nivel 2, Gestionado: La ejecución de las prácticas bases en los procesos son planificadas y seguidas. La primera distinción entre el nivel 1 y 2 es que la ejecución de los procesos está planificada y administrada y progresan hacia un proceso bien definido.

Nivel 3, Establecido: Las prácticas bases son ejecutadas de acuerdo a una versión adaptada del estándar, procesos aprobados bien definidos y documentados.

Nivel 4, Previsible: Mediciones detalladas de rendimiento o ejecución son alcanzadas y evaluadas. Es conocido el rendimiento de los procesos y es posible su predicción. La calidad de las prácticas es cuantitativamente conocida.

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

Ventajas

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

Desventajas

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

1.7. Modelos para el aseguramiento de la calidad

Modelo de evaluación ISO/IEC 14598

La serie de normas ISO/IEC 14598 indica los requisitos a tener en cuenta para la aplicación de los métodos de medición y para el proceso de evaluación, proporcionando un entorno de trabajo para la evaluación de la calidad de diferentes tipos de productos software (Rubalcaba, y otros, 2008).

Esta norma comprende las siguientes seis partes que especifican el proceso a seguir para evaluar software:

- ISO/IEC 14598-1 Visión general
- ISO/IEC 14598-2 Planificación y Gestión
- ISO/IEC 14598-3 Procedimiento para desarrolladores
- ISO/IEC 14598-4 Procedimiento para compradores
- ISO/IEC 14598-5 Procedimiento para evaluadores
- ISO/IEC 14598-6 Documentación de los módulos de evaluación.

Uno de los principales objetivos del proceso de evaluación Norma ISO/IEC 14598 es promover las siguientes características deseables en el proceso de evaluación:

- *Repetibilidad: La evaluación repetida de mismo producto con respecto a la misma especificación de evaluación por el mismo evaluador debería producir resultados que pueden ser aceptados por ser idénticos.*
- *Reproducibilidad: La evaluación repetida del mismo producto con respecto a la misma especificación de evaluación por un evaluador diferente debería producir resultados que pueden ser aceptados por ser idénticos.*
- *Imparcialidad: La evaluación no debería estar orientada hacia un resultado particular.*

- *Objetividad: Los resultados de la evaluación deberían ser ciertos, por ejemplo, no influidos por los sentimientos u opiniones del evaluador (IEEE, 1990).*

El principal propósito de la evaluación del software es apoyar directamente tanto el desarrollo como la adquisición de un software que satisfaga las necesidades del usuario y del cliente teniendo como objetivo final asegurar que el producto aporte la calidad requerida y satisfaga las necesidades declaradas e implícitas de los usuarios.

Proceso de Evaluación de Modelo de Capacidad y Madurez Integrado

Desde los albores de la disciplina de la ingeniería del software, queda patente la dificultad para que los artefactos generados alcancen un nivel de calidad óptimo dentro de unos límites de tiempo y coste. Dada la naturaleza lógica del producto, se asume que la calidad de un sistema software depende sobremanera de la calidad del proceso usado para desarrollarlo. Los modelos de evaluación y mejora de procesos y su estandarización, han tomado un papel determinante en la identificación, integración, medición y optimización de las buenas prácticas existentes en la organización y desarrollo software.

El Modelo de Capacidad y Madurez Integrado (CMMI, por sus siglas en inglés) dirige su enfoque a la mejora de procesos en una organización, estudia los procesos de desarrollo y produce una evaluación de la madurez (indicador para medir la capacidad para construir un software de calidad) de la organización según una escala de cinco niveles (inicial, repetible, definido, dirigido y optimizado). Los modelos contienen los elementos esenciales de procesos efectivos para una o más disciplinas y describen el camino para evolucionar y mejorar desde procesos inmaduros a procesos disciplinados, maduros con calidad y eficiencia mejorada y probada.

CMMI se encuentra dividido en etapas (los niveles de madurez), para la mejora organizacional basada en procesos probados, agrupados y ordenados y sus relaciones asociadas. Cada nivel de madurez tiene un conjunto de áreas de proceso que indican donde una organización debería enfocar la mejora de su proceso. Cada área de proceso se describe en términos de prácticas que contribuyen a satisfacer sus objetivos. *Las prácticas describen las actividades que más contribuyen a la implementación eficiente de un área de proceso; se aumenta el 'nivel de madurez' cuando se satisfacen los objetivos de todas las áreas de proceso de un determinado nivel de madurez (CMMI, 2006).*

1. Inicial (Proceso impredecible, control reactivo)
2. Gestionado (Gestión básica del proyecto)
3. Definido (Proceso caracterizado por la organización y proactivo)
4. Gestionado cuantitativamente (Control cuantitativo del proceso)
5. Optimizado (Mejora continua del proceso)

Modelo CMMI V1.2

En Agosto de 2006 aparece CMMI V 1.2, incluye mejoras significativas dando respuesta a asuntos que fueron surgiendo en la práctica de la versión anterior (CMMI v1.1). Los cambios se producen para la calidad de los productos CMMI y la consistencia con que son aplicados. Este modelo de calidad cuenta con 22 áreas de procesos las cuales se agrupan en 4 categorías (Sánchez, y otros, 2008):

Administración de procesos

- Innovación y despliegue de organización
- Definición de proceso de organización + Proceso de Desarrollo y Producto Integrado (IPPD, por sus siglas en inglés)
- Foco de proceso de organización
- Funcionamiento de proceso de organización
- Entrenamiento de la organización

Administración de proyecto

- Gestión del proyecto integrada + IPPD
- Monitorización y control del proyecto
- Planificación del proyecto
- Gestión cuantitativa del proyecto
- Gestión de Riesgo
- Gestión de acuerdos con los proveedores

Ingeniería

- Integración del producto
- Desarrollo de los requisitos
- Administración de los requisitos

- Solución técnica
- Validación
- Verificación

Soporte

- Análisis causal y resolución
- Administración de configuración
- Análisis y resolución de la decisión
- Medida y análisis

1.8. Proceso de pruebas

La prueba es el proceso de ejecutar un programa con intención de encontrar defectos. *La prueba exitosa es aquella que descubre defectos. El “caso de prueba bueno” es aquel que tiene alta probabilidad de detectar un defecto aún no descubierto. El “caso de prueba exitoso” es aquel que detecta un defecto aún no descubierto* (Pressman, 1998). La prueba no es: (1) demostración que no hay errores, (2) demostración que el software desempeña correctamente sus funciones y (3) establecimiento de confianza que un programa hace lo que debe hacer.

“La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación” (Pressman, 1998).

Otro criterio define la prueba como: *“... una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente”* (IEEE, 1991).

La prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la confiabilidad del software e indican la calidad del software como un todo. Pero, Glen Myers afirma que: *“La prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software”* (Myers, 2005).

Una estrategia tradicional de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente.

La prueba de unidad hace un uso intensivo de las técnicas de prueba de caja blanca, ejercitando caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores. La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software: el componente de software o módulo. Además de las estructuras de datos locales, durante la prueba de unidad se debe comprobar el impacto de los datos globales sobre el módulo.

La prueba de integración es una técnica sistemática que permite construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es juntar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se combinan todos los módulos por anticipado. Se prueba todo el programa en conjunto. Se encuentra un gran conjunto de errores. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa en lo que parece ser un ciclo sin fin.

La prueba de validación proporciona una seguridad final que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante la validación se usan exclusivamente técnicas de prueba de caja negra. El software, una vez validado, se debe combinar con otros elementos del sistema.

La prueba del sistema verifica que cada elemento se ajusta de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total. La prueba del sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se ha integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

Las pruebas de sistema están divididas en dos grandes grupos:

- **Funcionales:** *Enfocadas a los requisitos funcionales del software, a su interacción con el cliente de la forma que ha sido pactada.*

➤ **No funcionales:** Enfocadas a los requisitos no funcionales del proyecto. Desglosadas en un número de pruebas definidas como:

- ✓ Prueba de Seguridad
- ✓ Prueba de Disponibilidad y Red
- ✓ Prueba de Rendimiento o Carga
- ✓ Prueba de Compatibilidad
- ✓ Prueba de Resistencia o Stress
- ✓ Prueba de Usabilidad
- ✓ Prueba de Fiabilidad (Rubalcaba, y otros, 2008)

Existen principios planteados por Davis, Edward y Myers que guían las pruebas de software tales como (Pressman, 1998) :

1. La principal dificultad del proceso de prueba es decidir cuándo parar.
2. Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
3. Una parte necesaria de un caso de prueba es la definición del resultado esperado.
4. Los casos de pruebas tienen que ser escritos no solo para condiciones de entradas válidas y esperadas, sino también para condiciones no válidas e inesperadas.
5. Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
6. El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
7. Las pruebas deberían empezar por "lo pequeño" y progresar hacia "lo grande".
8. Con la excepción de las pruebas de unidad e integración, un programa deberá ser probado por la persona u organización que lo desarrolló.
9. Asigna el programador más creativo a la prueba.

La prueba de regresión es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse que los cambios no han propagado efectos colaterales no deseados. Este tipo de prueba es la actividad que ayuda a asegurar que los cambios no introduzcan un comportamiento no deseado o errores adicionales.

A medida que progresa la prueba de regresión, el número de la misma puede crecer demasiado. Por lo tanto, el conjunto de pruebas de regresión debe diseñarse para incluir sólo aquellas pruebas que traten una o más clases de errores en cada una de las funciones principales del programa. No es práctico ni eficiente volver a ejecutar cada prueba de cada función del problema después de un cambio.

Cuando se construye un software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo de sistema. Una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas.

El diseño de casos de prueba para el software o para otros productos de ingeniería puede requerir tanto esfuerzo como el propio diseño inicial del producto. Sin embargo, los Ingenieros de software tratan las pruebas como algo sin importancia, desarrollando casos de prueba que “parezcan adecuados”, pero que tienen poca garantía de ser completos. Se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. Cualquier producto de ingeniería puede probarse de una de estas 2 formas: (1) prueba de caja negra y (2) prueba de caja blanca.

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

Plan de pruebas

El Plan de Prueba describe las estrategias, recursos y planificación de la prueba. *La estrategia de prueba incluye la definición del tipo de pruebas a realizar para cada iteración y sus objetivos, el nivel de cobertura de prueba y de código necesario y el porcentaje de pruebas que debería ejecutarse con un porcentaje específico* (Jacobson, y otros, 2009).

El Plan de Pruebas proporciona el marco dentro del cual el equipo de prueba desarrolla las pruebas trabajando con los recursos y la planificación dada.

El Plan de Pruebas proporciona la siguiente información:

- La definición de los objetivos de las pruebas en el ámbito de la iteración.
- La definición de los elementos que se van a probar.
- Una explicación del enfoque o estrategia que se usará.
- Los recursos y planificación necesarios.
- Los resultados que se obtienen del proceso de prueba.

También se define el plan estratégico que se va a seguir en las pruebas. Es necesaria una estrategia de pruebas para convencer al gestor y a otros stakeholders de que el enfoque es sensato y alcanzable.

Una estrategia de prueba debe contener la siguiente información:

- *Una explicación del enfoque general que se usará.*
- *Especificar tipos, técnicas y estilos de prueba que se emplearán como parte de la estrategia, y para cada una:*
 - ✓ *Una indicación del ámbito y aplicabilidad de la técnica.*
 - ✓ *Un resumen de cómo se aplicará la técnica.*
 - ✓ *Un resumen de qué herramientas se necesitan para llevar a cabo la técnica.*
 - ✓ *El criterio de medida de éxito de la técnica.*
 - ✓ *Una indicación de las carencias o limitaciones de la técnica y que otra técnica las cubre (Plan de Pruebas, 2008).*

El objetivo general del plan es establecer la cronología y condiciones para la aplicación de las pruebas de manera que se pueda obtener, una Estrategia de aseguramiento de la calidad que pueda ser completado con una recepción total de los interesados.

Lista de Chequeos

Las Listas de Chequeo están basadas en la identificación de las técnicas de prueba para evaluar cada subcaracterística de las características de calidad. Una lista de chequeo es un formulario de preguntas, las cuales dependen del objetivo para el cual son usadas. Estas listas están clasificadas según las etapas del proceso de prueba. Para dar respuesta a cada pregunta se considera una escala del 1 al 5, en donde el uno siempre es la respuesta menos significativa y cinco la más significativa.

Todo lo anteriormente presentado soporta el diseño de la Estrategia de prueba. Esta consiste en un conjunto de acciones organizadas y secuenciales que se sintetizan en esta estrategia.

1.9. Situación actual de la calidad del software.

La industria del software y de las tecnologías de la información es uno de los sectores productivos más dinámicos de la economía mundial. El mercado de las empresas relacionadas con el software factura más de 500 mil millones de dólares al año y tiene un ritmo de crecimiento por arriba de los dos dígitos anuales (Seoane, y otros, 2009).

Sin embargo, su evolución depende del grado de desarrollo humano de cada país, así como de los adelantos científicos y tecnológicos. Por esto, de manera general se considera que los países del primer mundo cuentan con un nivel de tecnología y habilidades muy superiores al de los países subdesarrollados.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Cuando la innovación se mezcla con la calidad y la capacidad de producción de software, las compañías consiguen lanzar al mercado productos altamente competitivos en funcionalidad y coste. Incluso, algunos Fabricantes de Software Independiente (ISVs por sus siglas en inglés) cuentan que el reconocimiento internacional de sus certificaciones les facilita la exportación de sus productos.

Prueba de ello es que, aún cuando Estados Unidos es el mayor productor y consumidor de servicios informáticos y de software, en las últimas dos décadas tres naciones han tenido un alto nivel de desarrollo en esta industria. Se trata de las tres famosas "I": India, Irlanda e Israel. Tres países muy distintos entre sí y que, partiendo de diferentes estrategias, se han convertido en referentes de esta industria (Seoane, y otros, 2009)

El crecimiento de la industria del software en la India es un claro ejemplo de cómo una nación en desarrollo se pudo convertir en una potencia en la nueva economía de la información. *Este país se ha*

consolidado como uno de los líderes en exportación de software y servicios informáticos en todo el mundo, al grado que el llamado Bangalore Valley (ciudad india con más de mil 200 empresas tecnológicas, tanto locales como extranjeras) compite como un gran centro de desarrollo tecnológico. Actualmente, más de un centenar de firmas como Microsoft, Google, Reuters y Cisco han abierto centros en la India (Seoane, y otros, 2009).

A Irlanda se le conoce como “El Tigre Celta”, debido a que su economía crece a tasas que se parecen más a las del sudeste asiático que a las de Europa. Irlanda cuenta con uno de los mejores servicios de telecomunicaciones en Europa, con un mercado totalmente desregulado y hay cerca de 20 compañías brindando los servicios de valor agregado. Cuenta con una red de conectividad con el mundo ilimitada, que tiene capacidad para transportar terabits de información de Protocolo de Internet (IP, por sus siglas en inglés). Cabe subrayar que el ser miembro de la Unión Europea le da a Irlanda el acceso a un mercado de 400 millones de personas.

En Israel hay en la actualidad más de 35 mil especialistas en cómputo altamente calificados, con un dominio del idioma inglés, que laboran en las más de tres mil industrias de alta tecnología. Empresas como Microsoft, IBM, SAP, HP, Intel, entre otras, se han asentado en ciudades como Haifa y Tel Aviv para crear centros de investigación y desarrollo. Israel tiene la segunda mayor concentración de empresas informáticas, solo superado por el Silicon Valley en Estados Unidos (Seoane, y otros, 2009).

Cuba, aunque no con tan grandes logros tecnológicos y de comunicación, también posee un alto interés por mejorar la competitividad en el mercado internacional de las empresas del sector de tecnologías de la información. *En el gobierno cubano se han trazado una serie de proyecciones dentro de las cuales se encuentra el fomento de una Industria Cubana del Software, que permita diseñar y proveer de equipos electrónicos y sistemas informáticos que beneficien a la sociedad y también con posibilidad de exportarlos para aportar a la base material de todos los programas del país (Febles, 2003).*

El desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el estado cubano debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país. A pesar de ello, los resultados alcanzados no cubren las expectativas, ya que la productividad es baja, la cantidad real de recursos a consumir en tiempo principalmente es casi

impredicible y el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida. *Los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas, no se detectan en este medio indisciplinado y caótico de desarrollo* (Febles, 2003).

Es por ello que, para la obtención de productos con la calidad requerida y lograr el aseguramiento de la misma, la Industria Cubana del Software establece la necesidad de utilizar metodologías o procedimientos, estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo. Todo lo anterior, en aras de alcanzar una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

1.10. Calidad del software en la UCI

La Universidad de las Ciencias Informáticas se ha convertido, en muy poco tiempo, en centro de referencia para la Industria Cubana del Software. En apenas 5 años sorprendería ver el número de proyectos que han nacido y crecido en la misma.

El modelo productivo que caracteriza la producción en la UCI es otro elemento distintivo. La vinculación formación-producción-investigación debe tenerse en cuenta todo modelo, procedimiento, plantilla que se defina. Todo esto ha impuesto un reto en cuanto a la organización de la producción con la máxima de que la cantidad no puede afectar la calidad.

Ayudados en la definición de calidad de Pressman (Pressman, 1998) *concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y de las características implícitas que se espera de todo software desarrollado profesionalmente* y en aras de lograr la calidad tanto en los procesos como en los productos se ha trabajado en la estandarización de los procesos de desarrollo de software y la documentación que en estos se genera.

En cualquier caso sería evidente la necesidad de querer documentar suficientemente el producto y su proceso de producción. Por tanto, desde que un proyecto se inicia debe abrirse un expediente que recoja toda la documentación necesaria para garantizar la calidad del proceso de producción del software como del producto en sí.

Como parte de los mecanismos que permitan mejorar la calidad de los procesos de desarrollo de software, estandarizar la documentación de todos los proyectos que se realizan, facilitar la formación y adiestramiento de los equipos de proyecto en el uso de modelos y estándares propios y para elevar la calidad del producto entregado al cliente, se ha diseñado un Expediente de Proyecto que deberá implantarse en cada uno de los proyectos vigentes en la universidad. Este esquema de expediente y grupo de plantillas tipo definidas tiene como objetivo influir en la estandarización de la documentación y la creación de una cultura de calidad en la organización.

La documentación asociada a los proyectos de software y sistemas debe cumplir con algunos requisitos como son:

- Servir como medio de comunicación entre los miembros del equipo
- Servir de repositorio de información que pueda ser utilizado por los ingenieros de sistemas
- Proveer información para el control de los planes, cronogramas e hitos en el proceso de desarrollo de software
- Definir quién hace y cómo hace las actividades específicas del desarrollo.

Satisfacer estos requerimientos requiere de un grupo de documentos que no son solamente las documentaciones técnicas, asociadas al producto. No es posible organizar la documentación de los proyectos sin definir un esquema genérico para la organización que norme como usar y evaluar la documentación según el tipo de proyectos.

Con todo lo mencionado anteriormente se llegó a la siguiente definición: el Expediente de Proyecto es la herramienta que agrupa y organiza todos los artefactos que se generan durante el desarrollo de software.

La confección del expediente se dividió en varias etapas:

1. Identificar las necesidades de documentación planteadas por CMMI v1.2
2. Identificar las necesidades propias de documentación de los proyectos de la UCI
3. Revisar plantillas propuestas por normas y estándares y adaptarlas al expediente definido para la UCI
4. Adicionar las plantillas que por las características de la UCI no se encuentran presentes en los modelos seleccionados

En la actualidad no existe una homogeneidad en la documentación que se construye por parte de los proyectos, lo que dificulta la retroalimentación que pudiera obtenerse de la misma, así como su reutilización por parte de otros proyectos. Todo esto tributa a la carencia de calidad de los productos de la Universidad de la Ciencias Informáticas (UCI) contribuyendo así a la realización de software que no cumplen con las necesidades del cliente.

1.11. Grupo de Alternativas Libres para Multimedia

En la Universidad de las Ciencias Informáticas, se desarrollan una serie de actividades para garantizar la integridad de los productos desarrollados por la misma. Al analizar las solicitudes de los clientes, se tiene en cuenta la previsible evolución de sus necesidades y tendencias en cuanto a características. Además, se tiene en cuenta la evolución tecnológica del entorno de producción de las aplicaciones para suministrarlas con el nivel tecnológico adecuado; sin olvidar el nivel de calidad de los competidores, debiendo elaborar productos cuyas características y funcionalidades cumplan con los requerimientos del cliente.

El Grupo de Alternativas Libres para Multimedia está enfrascado en la creación de un Entorno de Desarrollo Integrado (IDE) multi-plataforma realizado con herramientas libres que permita desarrollar productos educativos en formato multimedia a través de una interfaz amigable para los desarrolladores de Software Educativo con un alto nivel de fiabilidad. Esta herramienta contribuye al desarrollo eficaz del plan de migración hacia software libre que está implementando el país y específicamente la UCI para extender sus productos y mejorar los servicios con fines educativos en formato multimedia. Para garantizar su integridad se le está realizando el proceso de calidad de software a este IDE.

La calidad para GALM ha dejado de ser un tópico y actualmente es necesario que forme parte del producto que elabora para el cliente. El cliente es el mejor auditor de la calidad, él exige el nivel que está dispuesto a pagar por ella, pero no más. Por tanto, se debe cuantificar cuál es el nivel de calidad que se exige para poder planificar la calidad del producto que se genera a lo largo de la confección del producto o servicio final y; además una estrategia para asegurarla.

1.12. Conclusiones del Capítulo

En el desarrollo de este capítulo se ha realizado un análisis general de la calidad del software a través del estudio de la situación actual de la misma por lo que se ha arribado a la conclusión de que la calidad es un factor muy importante en el producto software respondiendo a los requisitos, las necesidades y las expectativas del cliente o usuario final.

Se abordó sobre diferentes modelos y estándares de calidad y después de un estudio exhaustivo, teniendo en cuenta la bibliografía consultada, se decide utilizar para guiar todo el proceso de aseguramiento de la calidad el estándar ISO/IEC 9126 no así el SPICE que aunque ofrece una base para una detallada evaluación del estado actual del proceso de una organización no es práctico ni fácil de aplicar; mientras que el ISO/IEC 9126 resulta más apropiado por las métricas que proporciona y permite distinguir entre fallos y no conformidad.

De los modelos de evaluación se tuvo en cuenta CMMI ya que constituye un marco de referencia de la capacidad de las organizaciones de desarrollo de software en el desempeño de sus diferentes procesos y el ISO/IEC 14598, el cual indica los requisitos a tener en cuenta para la aplicación de los métodos de medición y para el proceso de evaluación. Teniendo en cuenta lo anteriormente expuesto no se selecciona ISO/IEC 14598 para llevar a cabo el proceso de aseguramiento de la calidad y sí CMMI v1.2 porque este último proporciona una base para la evaluación de la madurez de la organización y una guía para implementar una estrategia para la mejora continua.

Además, el modelo de evaluación CMMI v1.2 presenta la ventaja de que es utilizado por la Dirección de Calidad y la UCI actualmente paga por la valoración de este modelo, por lo que; utilizarlo constituye alinear al proyecto de acuerdo a los objetivos de la Universidad.

Otro de los temas tratados en la investigación fue el proceso de pruebas, las cuales demuestran hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento, destacándose dos artefactos importantes el Plan de Pruebas y las Listas de Chequeo. Se presentó un estudio referente a la situación actual de la calidad del software y particularmente en la UCI así como una breve descripción del Grupo de Alternativas Libres para Multimedia.

Proceso de Aseguramiento de la Calidad



Proceso de Aseguramiento de la Calidad

2.1. Introducción

Para la realización de una Estrategia de aseguramiento de la calidad es necesario efectuar un análisis de la situación actual en la que se encuentra el proyecto en cuanto al área de aseguramiento de la calidad, por lo que este capítulo estará dedicado a estudiar el estado real del Grupo de Alternativas Libres para Multimedia en este sentido.

Primeramente se realizará una descripción del proyecto donde se aborda su necesidad y objetivos, la que proporcionará información sobre qué es lo que se pretende con su desarrollo. Seguidamente se especifica la metodología a utilizar, los roles que ésta establece así como los artefactos correspondientes. Además de la realización de una descripción minuciosa de la estrategia a seguir para el proceso de aseguramiento de la calidad en el proyecto.

2.2. Descripción del Grupo de Alternativas Libres para Multimedia

La mayor parte de las grandes compañías productoras de herramientas y materiales para el sector de la información relacionado con la educación son de origen norteamericano; y las trabas que impone su gobierno para el uso libre de estos productos llevan a la necesidad de incentivar la producción nacional de Software Educativo basándose en nuevas soluciones.

Dichas soluciones deben resolver las necesidades de la nación y permitir la comercialización o instauración de productos cubanos en el exterior. Además en nuestro país se lleva a cabo un plan de migración a software libre y específicamente en la UCI este proceso de migración repercute negativamente en la producción de Software Educativo en formatos multimedia, puesto que los software

que existen actualmente para la producción de este tipo de productos no cuentan con los requisitos de diseño, programación y compilación de manera integrada en una sola aplicación que facilite el trabajo.

Por lo anteriormente expuesto se crea en la Universidad de las Ciencias Informáticas el Grupo de Alternativas Libres para Multimedia, y como parte de este proyecto se decide desarrollar un IDE para la realización de productos multimedia basados en alternativas libres. Además de que se convierta en la herramienta estándar para el desarrollo de Software Educativo basado en alternativas libres en la UCI. Debe permitir su ampliación a otros lenguajes además del ActionScript 2.0 e incluir compiladores para generar archivos en formato SWF, así como frameworks para el desarrollo de productos multimedia de forma dinámica. El cliente de este IDE es el Polo de Software Educativo de la Facultad 8.

Roles y responsabilidades

En el presente trabajo de diploma se encuentran definidos los roles y responsabilidades dentro del proyecto de acuerdo a la metodología utilizada en el mismo. Debido a que la metodología no establece un rol específico para el tema del aseguramiento de la calidad se propone la creación de un rol que responde al nombre de Administrador de Calidad, encargado de todo el proceso de aseguramiento de la calidad guiado por la estrategia que aquí se propone. Además no se considera un número mayor de desarrolladores para este proceso por la reducida cantidad de personal disponible.

Recursos humanos por roles

- Roles de profesores:
 - ✓ Líder de proyecto: 1 persona
 - ✓ Gerente: 5 personas
 - ✓ Encargado de pruebas: 2 personas
 - ✓ Analista: 5 personas
 - ✓ Arquitecto: 5 personas
 - ✓ Programador: 2 personas
- Roles de estudiantes:
 - ✓ Programador: 3 personas
 - ✓ Analista: 4 personas
 - ✓ Encargado de pruebas: 1 persona

- ✓ Arquitecto: 2 personas
- ✓ Administrador de la Calidad: 1

El Grupo de Alternativas Libres para Multimedia cuenta con un total de 14 integrantes en su composición, divididos en 5 profesores y 9 estudiantes. En la mayoría de los casos ocurre un solapamiento de roles, es decir una persona puede desempeñar en diferentes etapas, en dependencia de su responsabilidad inicial, varios roles.

Soporte tecnológico

Como soporte tecnológico, GALM cuenta con 8 estaciones de trabajo formadas por 8 computadoras, de las cuales 5 se encuentran en perfectas condiciones, que están siendo utilizadas mediante un tiempo de máquina establecido y 3 presentan dificultades técnicas que imposibilitan el desarrollo del proyecto. Se cuenta además con una computadora que hace función de servidor donde se almacena la información contenida en el repositorio del proyecto.

Módulos del proyecto

El Grupo de Alternativas Libres para Multimedia consta de tres proyectos, estos son:

Subsistema Editor Visual, Subsistema de Clases y Componentes Reutilizables, Subsistema Editor de Código. A continuación se hace una breve descripción de cada uno de estos productos:

➤ Subsistema de Clases y Componentes Reutilizables (CAFES)

El producto es un paquete de clases y componentes dinámicos desarrollados en ActionScript 2.0, orientado al desarrollo de productos Multimedia. Estos componentes y clases implementan los principales ejercicios y juegos educativos que usualmente son incluidos en muchos productos Multimedia.

➤ Subsistema Editor de Código (Davinchi)

El producto es un IDE para el desarrollo de productos Multimedia orientado a la gestión de código, basado en alternativas libres y extensible a varios lenguajes a partir de un sistema de plugins, para su construcción se utilizará Qt4 como framework para el desarrollo y C++ para la implementación

➤ Subsistema Editor Visual (Picassus)

El producto es un IDE para el desarrollo de productos Multimedia basado en alternativas libres, para su construcción se utilizará Qt4 como framework para el desarrollo y C++ para la implementación. El objetivo principal es proveer a la Universidad de una herramienta de autor sin restricciones de licencia que permita el desarrollo de aplicaciones multimedia utilizando software libre, y que amplíe las posibilidades de comercialización de dichos productos.

El proceso de aseguramiento de la calidad se lleva a cabo actualmente en el Subsistema Editor Visual, ahora como producto que recibe el nombre de Picassus, donde se integran funcionalidades propias del Subsistema de Clases y Componentes Reutilizables y el Subsistema Editor de Código.

2.3. Fundamentación de la metodología a utilizar

Para lograr una documentación con calidad y que no duplique los esfuerzos de los desarrolladores es necesario valorar las condiciones y características del proyecto. La UCI y principalmente el Grupo de Alternativas Libres para Multimedia, no están ajenos a la situación antes planteada, pues es lógico que el esquematismo al documentar trae como consecuencias, proyectos con posibilidades de fracasar, otros con un desarrollo seriamente retardado y afectado por el tiempo, y en la mayoría de los casos la imposibilidad de reutilizar el software producido, por falta de información y de referencias sobre el tema.

Actualmente la mayoría de los proyectos de la UCI realizan el proceso de desarrollo del software bajo la guía del Proceso Unificado de Desarrollo (RUP por sus siglas en inglés). Esta metodología establece un proceso dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura. Además, realiza un levantamiento exhaustivo de requerimientos y detecta los defectos en las fases iniciales. A pesar de las ventajas que ofrece esta metodología, RUP no se adapta a las necesidades y características de GALM por lo que se ha decidido utilizar otro tipo de metodologías denominadas ágiles.

Independientemente de que RUP puede ser utilizada de forma ágil, se seleccionó como metodología de desarrollo a ScrumXP ya que propone pocos artefactos, no siendo así con RUP que dedica muchas horas y esfuerzo de trabajo en la documentación. ScrumXP cuenta con pocos roles ajustándose al número de integrantes de GALM y el cliente forma parte del equipo de desarrollo, mientras que RUP establece una mayor cantidad de roles y el cliente interactúa con el grupo de trabajo mediante reuniones.

Debido a esto, el desarrollo de GALM será guiado utilizando adecuadamente la unión de las metodologías ágiles XP y Scrum para el logro de un buen desarrollo de software, propuesta en el 2007 por la ingeniera Malay Rodríguez Villar, y probada en los proyectos que trabajan con software libre en la UCI obteniendo buenos resultados. Además, para la realización del proyecto se selecciona esta metodología porque constituye un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores del producto que las metodologías convencionales.

A continuación en la Tabla 1 se muestran las diferentes fases en que está dividida la metodología y los artefactos que se generan en cada una de ellas, que son precisamente la base de la estructura del expediente de proyecto que se realizará con el propósito de asegurar la calidad de GALM.

Tabla 1. Fases de la Metodología y Artefactos.

Fuente: Elaboración propia.

Fases	Artefactos Generados	Descripción
Planificación - Definición.	Concepción del Sistema	Refleja la visión general del producto a implementar. Recoge los roles y sus responsabilidades. Es la base de los demás documentos que se generan.
	Modelo de Historias de usuario del negocio.	Se genera del juego de la planificación, luego de estar definida la concepción del sistema.
	Lista de Reserva del producto(LRP)	Primer artefacto generado en la etapa de Captura de requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto.
	Historias de Usuario	Especifican los requisitos del software, guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo.

Fases	Artefactos Generados	Descripción
	Lista de Riesgos	Quedan definidos los posibles riesgos que actuarán sobre el proceso de desarrollo de software, así como la estrategia trazada para mitigarlos, además de un plan de contingencia que describe que curso seguirán las acciones si el riesgo se materializa.
Desarrollo	Modelo de Diseño	Debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
Desarrollo	Tarea de Ingeniería	Permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación.
	Plan de releases	Define cuáles son las historias de usuario más significativas, y las ubican en las iteraciones según esta prioridad.
	Estándares de Programación	Se recoge el estándar utilizado y su explicación.
	Plan de Pruebas	Recoge los pasos y datos de cada uno de los involucrados en el proceso. Además de tener actualizadas las fechas en las que se realiza cada una de las funcionalidades a probar.
	Caso de prueba de aceptación	Valida que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.
Entrega	Manual de Usuario	Sirve de apoyo para la capacitación. En él se recogen todos los aspectos significativos que contribuyan a que los usuarios que interactúen con el sistema y lo comprendan mejor.

Fases	Artefactos Generados	Descripción
Entrega	Manual de identidad	Se trazan las líneas maestras de la imagen corporativa con las que el público identificará el producto.
	Manual de desarrollo	Recoge toda la información respecto a la implementación y el código fuente del producto
Mantenimiento	Gestión de cambios	Se recogen los cambios que se realicen en esta etapa, que pueden ser de corrección, adaptación o mejora. Además se lleva el control de la persona que realiza el cambio, así como de la que lo revisa.

Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el expediente de proyecto, estas actividades están recogidas en el guión de la metodología (ver Anexo 1).

La estructura del expediente de proyecto para el Editor Visual Picasso se encuentra en correspondencia con la metodología utilizada, la cual fue aprobada por la Dirección de Calidad para todos los proyectos que utilicen ScrumXP (ver Anexo 2).

Para el expediente de proyecto del Grupo de Alternativas Libres para Multimedia el Modelo de Historias de Usuario del Negocio no se realizará por la no existencia de un proceso de negocio, esto se explica dado que proyecto es la construcción de un software que facilita la realización de otro software.

Para la definición de los artefactos que se generan en cada una de las fases se tiene en cuenta como elemento fundamental, las características de las metodologías ágiles, las cuales tienen como premisa la no duplicación de esfuerzos, así como la integración del cliente en el equipo de desarrollo, esto garantiza que no haya necesidad de documentaciones extensas, solo se documenta lo necesario para una futura reutilización.

El desarrollo de software con metodologías ágiles exige de la creación de pequeños grupos de trabajo, donde los roles son pocos, pero están bien definidas sus actividades. *El principal aspecto antes de comenzar el proceso de documentación es distribuir las tareas por cada uno de los roles existentes, lo que*

garantiza un trabajo organizado, de ahí la necesidad de tenerlos bien definidos (Rodríguez, 2007).

A continuación en la Tabla 2 se muestran los roles para el Grupo de Alternativas Libres para Multimedia:

Tabla 2. Roles y Responsabilidades.

Fuente: Elaboración propia.

Roles	Descripción	Responsabilidades
Líder del Proyecto	Asegura que el proyecto se está llevando a cabo de acuerdo con las prácticas y que todo funciona según lo planeado.	Lista de riesgos.
Gerente	Es el responsable de tomar a las decisiones finales acerca de estándares y convenciones a seguir durante el proyecto.	Lista de riesgos. Plan de releases.
Cliente	Contribuye a definir las historias de usuario y los casos de prueba de aceptación, para validar su implementación.	Concepción del Sistema Lista de reserva del producto (LRP). Historias de usuario. Plan de releases Plan de pruebas. Caso de prueba de aceptación.
Programador	Define las tareas de ingeniería y produce el código del sistema. Además selecciona el estándar de programación a utilizar, controlando incluso la gestión de cambios.	Tarea de ingeniería. Estándares de programación. Plan de pruebas. Manual de usuario. Manual de desarrollo. Gestión de cambios.

Roles	Descripción	Responsabilidades
Analista	Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio, todo esto lo realiza junto con el cliente.	Concepción del Sistema Lista de reserva del producto (LRP). Historias de usuario. Glosario de términos.
Diseñador	Encargado del diseño del sistema, así como el de los prototipos de interfaces	Modelo de diseño.
Encargado de Pruebas	Escribe los casos de prueba de aceptación. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas. Puede ser el propio cliente o una persona ajena al equipo de desarrollo que determine el Líder del Proyecto.	Plan de pruebas. Caso de prueba de aceptación.
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema.	
Consultor	Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas, además aportan ideas y experiencias para el beneficio del sistema en desarrollo.	Manual de identidad.

Independientemente de los roles que establece la metodología utilizada en el proyecto se creará un rol con el nombre de **Administrador de la Calidad** que será un miembro del equipo de desarrollo, como máximo responsable del proceso de aseguramiento de la calidad guiado por la estrategia que en éste documento se establece.

Es importante tener en cuenta, que a pesar de aplicar este tipo de organización por roles en el Grupo de Alternativas Libres para Multimedia, no se debe de crear un esquematismo, ni pensar que un integrante del grupo de desarrollo, en todo el ciclo de vida de un software solamente se relacionará con su funcionalidad.

Esto garantiza la creación de un grupo de trabajo fuerte, donde siempre existirán personas preparadas para realizar cualquier actividad que sea necesaria, sin obviar como es lógico la necesidad de formar especialistas en temas específicos dentro del proyecto, que podrían convertirse más adelante en consultores.

El expediente de proyecto propuesto incluye un área legal, que es definida por la asesora de comercialización que atiende el Polo de Software Educativo. En ésta área se recogen todos aquellos documentos que son necesarios para la entrega de los productos, y su comercialización en el mercado.

Valoración.

Con esta propuesta se pretende dar solución a muchos de los problemas que afectan el proceso de documentación en el Grupo de Alternativas Libres para Multimedia, como son las documentaciones extensas, la duplicación de esfuerzos, así como el tiempo que se dedica a documentar, que en la mayoría de los casos es mayor que el dedicado a la producción.

Por otra parte al repartir la responsabilidad de llenado de las plantillas que conforman al expediente, teniendo en cuenta los roles definidos por la metodología utilizada en los proyectos, se garantizará que no existan plantillas cuyo rol responsable, no sea miembro del equipo de trabajo, lo que facilita las acciones del líder de proyecto, el cual no necesitará asignar a ninguna otra persona desconocedora de los aspectos que recoge la plantilla, para desarrollarla.

La estructura del nuevo expediente, facilitará el proceso de documentación, por parte de los miembros del equipo de desarrollo, pues se generan pocos documentos por cada una de las actividades que se realizan en las fases, todos de formato sencillo y fácil de llenar, evitando crear documentaciones largas y tediosas, tanto para hacerlas, como para leerlas.

2.4. Proceso de evaluación del software

El proceso de ejecución de pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad (Kruchten, 2000). Su éxito dependerá del seguimiento de una Estrategia de prueba adecuada. La Estrategia de prueba de software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba: la planificación, el diseño de casos de prueba de aceptación, la ejecución y los resultados, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software (Pressman, 2002).

La formulación de la Estrategia de prueba para el software del Grupo de Alternativas Libres para Multimedia aquí propuesta contempló seis pasos:

- Identificación de las etapas del proceso de pruebas.
- Propuesta de las Métricas. Instrumento de medición: Las Listas de Chequeo.
- El diseño y registro de Casos de Prueba de Aceptación.
- Establecimiento de pautas para procesar los resultados.
- Responsabilidades dentro del proceso de pruebas de Aceptación.
- Diseño final de la Estrategia de prueba al software orientado a objetos (EPS-OO).

2.4.1. Identificación de las etapas del proceso de pruebas

Las pruebas según lo propone la metodología de desarrollo utilizada se desagregaron en las subactividades de: Unidad, Integración y pruebas de Aceptación. Todo lo anterior debido a que un programador extremo debe siempre probar su código de forma unitaria, integrar continuamente y hacer prueba de aceptación con cierta regularidad, cada vez que una funcionalidad esta lista.

Las pruebas de Unidad centrarán el proceso de verificación en la menor unidad mientras que las de Integración serán desarrolladas con el propósito de se pruebe todo el programa en conjunto y se encuentre un gran conjunto de errores. Una vez establecidas las etapas a seguir durante el proceso de prueba, se procedió a asociar las Técnicas de pruebas para el software que garantizaran los Requisitos No Funcionales (RNF).

Para identificar los RNF se partió del Modelo de Calidad Sistémica del producto (Ortega, y otros, 2003). Este modelo permite identificar las características de calidad que deben ser evaluadas en un software. Estas características tienen a su vez subcaracterísticas asociadas. Se tomaron en cuenta las siguientes características: Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad.

En la Tabla 3 se muestra el resultado obtenido de la asociación de las técnicas de prueba con el RNF Fiabilidad a manera de ejemplo.

Tabla 3. Técnicas de prueba al RNF Fiabilidad

Fuente: Elaboración propia.

Subcaracterística	Objetivo	Técnicas que aplican
Madurez	Evaluar la capacidad que tiene el software para evitar fallas.	Prueba Negativa: Hacer que el sistema falle intencionalmente para medir su capacidad de respuesta frente a un error.
Tolerancia a Fallas	Verificar la capacidad del software para mantener un nivel de rendimiento específico ante un error, es decir, la capacidad de continuar procesando en caso de falla.	<p>Prueba de Valores Límites: Evaluar valores frontera; es decir, los valores mínimo y máximo que la unidad puede aceptar.</p> <p>Prueba Bajo Stress: Evaluar la habilidad del sistema para seguir operando apropiadamente ante bajos recursos o competencias para los recursos.</p>

Subcaracterística	Objetivo	Técnicas que aplican
Tolerancia a Fallas		<p>Prueba Negativa: Hacer que el sistema falle intencionalmente para medir su capacidad de continuar su ejecución a pesar de la falla.</p> <p>Prueba de Volumen: Someter al software a una gran cantidad de datos para determinar si los límites alcanzados hacen que este falle.</p>
Recuperabilidad	Verificar que el proceso de recuperación del sistema restaura apropiadamente la base de datos, la aplicación y el sistema a un estado conocido o deseado.	<p>Prueba de Recuperación: Exponer al software a condiciones extremas y verificar que la recuperación se realiza correctamente.</p>
Correctitud	<p>1) Evaluar la capacidad de cómputo.</p> <p>2) Comprobar la completitud de las formas estructurales y del software como un todo.</p> <p>3) Evaluar la consistencia.</p>	<p>Prueba Estructural: Verificar que las formas estructurales de las clases sean completas.</p> <p>Prueba de Ejecución: La capacidad de cómputo esperada se puede evaluar durante la ejecución del software en aquellos módulos en donde se apliquen cálculos.</p> <p>Prueba de Carga: Probar diferentes cargas para evaluar la capacidad de cómputo.</p>

Subcaracterística	Objetivo	Técnicas que aplican
Estructurado	En caso de que el software no sea desarrollado en lenguaje orientado a objetos puro se debe verificar que sea estructurado; es decir, que siga las reglas de la programación estructurada.	Prueba Estructural: Permite evaluar los valores de las variables, las constantes y los tipos de datos y si éstos son usados en el contexto en que se definieron.
Encapsulado	Si el software está desarrollado en lenguaje orientado a objetos puro no es necesario verificar que sea encapsulado, pero si no lo es, se debe aplicar las técnicas de la subcaracterística: Estructurado.	Prueba Estructural: Esta técnica permite evaluar los valores de las variables, las constantes y los tipos de datos y si éstos son usados en el contexto en que se definieron.

En el Anexo 3 se muestran esquemáticamente las etapas del proceso de prueba. Una vez establecidas, fueron asociadas técnicas a cada una de ellas. Las técnicas asociadas son aquellas que son pertinentes para ser aplicadas dentro de cada etapa identificada.

2.4.2. Propuesta de las métricas. Instrumento de medición: Las Listas de Chequeo.

Las Listas de Chequeo están basadas en la identificación de las técnicas de prueba para evaluar cada subcaracterística de las características de calidad. Una lista de chequeo es un formulario de preguntas, las cuales dependen del objetivo para el cual son usadas.

Estas listas están clasificadas según las etapas del Proceso de Prueba. Para dar respuesta a cada pregunta se considera una escala del 1 al 5, en donde el uno (1) siempre es la respuesta menos significativa y cinco (5) la más significativa. A fines ilustrativos, en la Tabla 4 se presenta la Lista de Chequeo para la Característica de Fiabilidad durante la actividad de Planificación.

Tabla 4. Lista de Chequeo para la Característica Fiabilidad.

Fuente: Elaboración propia.

Característica	Pregunta	Evaluación
Madurez	1) ¿Se realizó un proceso de levantamiento de información apropiado?	1= Inaceptable 2= Debajo del Promedio 3= Promedio 4= Bueno 5= Excelente
	2.- ¿El levantamiento de información fue establecido formalmente?	1= No 5= Si
Correctitud	1.- ¿Están siendo considerados todos los procesos necesarios para dar solución al problema?	1= No 5= Si

2.4.3. El diseño y registro de Casos de Prueba de Aceptación:

Para registrar cada uno de los Casos de Prueba realizados para dar respuesta a las Listas de Chequeo, se propone una planilla cuyos campos son los datos requeridos para estructurar un Caso de Prueba. Cada pregunta de las Listas de Chequeo puede generar uno (1) o más Casos de Prueba de Aceptación. Esta planilla, se muestra en la Tabla 5.

Tabla 5. Casos de Prueba de Aceptación.

Fuente: Documento Oficial de la metodología ScrumXP.

Caso de Prueba de Aceptación	
Código Caso de Prueba: <i>[Inicial del proyecto-número de la HU a la que pertenece la prueba-número de la prueba.]</i>	Nombre Historia de Usuario: <i>[Nombre de la HU a realizar prueba.]</i>
Nombre de la persona que realiza la prueba: <i>[Nombre y apellidos.]</i>	
Descripción de la Prueba: <i>[Descripción de la prueba realizada.]</i>	
Condiciones de Ejecución: <i>[Condiciones necesarias para poder realizar la prueba.]</i>	
Entrada / Pasos de ejecución: <i>[Serie de pasos necesarios para lograr la realización de la HU, y así realizar la prueba.]</i>	
Resultado Esperado: <i>[Que cumpla con las restricciones del producto.]</i>	
Evaluación de la Prueba: <i>[Satisfactoria o no satisfactoria.]</i>	

2.4.4. Establecimiento de pautas para procesar los resultados.

Se establecieron los pasos a seguir para obtener los resultados cuantitativos en base a las respuestas dadas en las Listas de Chequeo. Los resultados están determinados por las respuestas y por la importancia que los involucrados asignan a cada subcaracterística.

Las subcaracterísticas deben ser ponderadas de acuerdo a la evaluación; los involucrados dan un peso (en una escala del 1 al 5) a cada una de ellas a través de la Planilla de Ponderación, en donde el uno (1) siempre es el peso menos significativo y cinco (5) el más significativo. La Tabla 6 muestra la planilla de ponderación para la característica Fiabilidad.

Tabla 6. Planilla de Ponderación para la característica Fiabilidad.

Fuente: Documento Oficial de la metodología ScrumXP

Característica de Calidad: Fiabilidad					
Subcaracterística	Peso Asignado				
	1	2	3	4	5
Madurez					
Tolerancia a Fallas					
Recuperabilidad					
Correctitud					
Estructurado					
Encapsulado					

Los resultados pueden obtenerse en base a dos criterios: respuestas a las preguntas de las Listas de Chequeo y la ponderación.

Dar respuesta a las preguntas de las Listas de Chequeo. Las respuestas a las preguntas de las Listas de Chequeo se darán mediante la realización de **Casos de Prueba**. A través de los Casos de Prueba, las respuestas dependerán de los resultados obtenidos en los mismos.

Los **Casos de Prueba** serán evaluados por medio de la siguiente escala:

Aprobado = 5
 No aprobado = $\left\{ \begin{array}{l} \text{Falla Menor} = 3 \\ \text{Falla Grave} = 1 \end{array} \right.$

Una pregunta puede ser contestada por más de un **Caso de Prueba**. En este caso la respuesta a la pregunta siempre será el menor valor obtenido por los casos de prueba considerados. Esto implica que de haber un caso de prueba no aprobado, éste afectará al resto en cuanto a la calificación de la calidad.

Ponderación de resultados. A partir del procesamiento de las respuestas dadas en las Listas de Chequeo, se generan tres (3) tipos de resultados:

a) Resultados de la presencia de las subcaracterísticas en cada etapa del proceso de prueba, según la característica de calidad a la que corresponde.

1) Se calcula el promedio aritmético de las respuestas de cada pregunta de la subcaracterística que se está evaluando.

2) Sobre la base de los promedios anteriores, la presencia de las subcaracterísticas tendrán los siguientes valores:

1= La subcaracterística no está presente en esta etapa.

2= La subcaracterística se presenta de manera muy deficiente.

3= La subcaracterística se presenta medianamente.

4= La subcaracterística se encuentra presente.

5= La subcaracterística se encuentra altamente presente.

b) Resultados de la Presencia de las Características de Calidad (PCC) en cada una de las etapas, considerando la importancia dada por los involucrados. Una vez obtenidos los resultados de todas las subcaracterísticas, se procederá a realizar los cálculos para obtener la evaluación de la característica de calidad.

Este cálculo se realizará de la siguiente manera:

1) Se calcula el promedio ponderado de las subcaracterísticas tomando en cuenta los pesos que le han sido asignados a cada una de ellas.

2) Para calcular este promedio ponderado se multiplican los valores obtenidos de cada subcaracterística (SC) por su peso correspondiente (P).

3) Se suman los valores obtenidos de la multiplicación y se divide este valor entre la suma de todos los pesos. *Este cálculo se representa a través de la siguiente fórmula (Grimán, y otros, 2008):*

$$PCC = \frac{\sum SC * P}{\sum P}$$

Este resultado representa la evaluación de la presencia de la característica de calidad dentro de una etapa. Esta evaluación será representada en un rango del 1 al 5, al igual que los resultados dados para las subcaracterísticas. El valor obtenido es llevado a porcentaje con el fin de identificar si el mismo tiene el **nivel de aceptabilidad**. El nivel de aceptabilidad es **75%** y es el recomendado para decidir si una característica está presente o no. Para los casos de que no alcance este valor mínimo de presencia se recomienda que sea revisada la característica en todo el proceso de desarrollo del software.

c) Resultados de la Presencia de las Características de Calidad en todo el Sistema (PCCS).

Después de realizar las evaluaciones de las características de calidad en cada una de las etapas del proceso de prueba, se obtiene el Porcentaje de Presencia de cada una de las Características de Calidad en todo el Sistema (**PCCS**) el cual es el promedio de los Porcentajes de Presencia de cada una de las Características en cada Etapa del proceso de prueba (**PPCE**). Es importante señalar que se tomó, nuevamente, como **nivel de aceptabilidad** de la característica de calidad un valor del **75%**. Una presencia con un valor por debajo del **75%** se considera deficiente en el software que se está evaluando.

2.4.5. Responsabilidades dentro del proceso de pruebas.

La Estrategia de prueba además de indicar cómo se realizará el proceso de evaluación establece las responsabilidades dentro del equipo de trabajo para su aplicación. En dicha estrategia se establece que para cada iteración se debe desarrollar un Plan de Pruebas el cual debe contemplar los procedimientos para probar algoritmos muy complejos en el ámbito de la unidad así como todo lo referente a las pruebas de Aceptación. En el Plan no se consideran las pruebas de Integración debido a no requerirse formalmente las mismas. Este Plan de Pruebas será realizado por el programador principal y el Administrador de la Calidad.

Las pruebas de Unidad serán realizadas durante el desarrollo de cada Historia de Usuario por el programador que se encuentre desarrollándola; mientras que las pruebas de Integración serán realizadas al mismo tiempo que se estén desarrollando las Historias de Usuario por los miembros del equipo de desarrollo responsables del módulo que se esté integrando.

Antes de desarrollar cada Historia de Usuario se diseñarán los Casos de Prueba de Aceptación, esta actividad será realizada por el programador que desarrolla la Historia de Usuario en cuestión. Las pruebas de aceptación serán realizadas a lo largo de la iteración una vez terminada la Historia de Usuario correspondiente y serán llevadas a cabo por una persona ajena al equipo de desarrollo, el cliente o alguien que establezca previamente el Líder del proyecto. La ponderación de los resultados que determinará el nivel de aceptabilidad de las características de calidad será realizada por el Administrador de la Calidad.

Del proceso de pruebas se generará una serie de no conformidades que deberán ser solucionadas antes de continuar desarrollando otras Historias de Usuario, esta actividad será máxima responsabilidad del programador. Todas estas actividades deben ser ejecutadas con la mayor rigurosidad para propiciar la obtención de un producto que cumpla las necesidades y expectativas del cliente.

2.4.6. Diseño final de la estrategia de prueba.

Todo lo anteriormente presentado soporta el diseño de la Estrategia de Prueba. Esta consiste en un conjunto de acciones organizadas y secuenciales que se sintetizan en una tabla (ver Anexo 4). Esta Estrategia de Prueba, producto de la investigación, se ha nombrado, a los fines de su presentación y posterior referencia, como EPS-OO. Se presentan las actividades llevadas a cabo para el diseño de la EPS-OO con su descripción, participantes, producto obtenido y sus entregables.

Para mayor comprensión del proceso de pruebas se realizó un Diagrama de Actividades de los RNF, pues a través del diagrama se puede observar en todos los casos, las actividades, artefactos y responsables de los mismos (ver Anexo 5).

Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo que se implementó es lo que en realidad se pensaba que se había implementado. Las pruebas indican que nuestro trabajo funciona, cuando no se pueda pensar en ninguna prueba que pudiese originar un fallo en el sistema entonces se ha terminado por completo.

2.5. Aseguramiento de la calidad

2.5.1. Plan de Aseguramiento de la Calidad

Apoyándose en las planillas propuestas por la Dirección de Calidad de la UCI, se implementa el Plan de Aseguramiento de la Calidad que tiene en su contenido varias secciones, estas van a facilitar la organización y comprensión de todas las actividades de aseguramiento de la calidad en el proyecto.

El Plan de Aseguramiento 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.

En esta actividad es de vital importancia determinar los objetivos de calidad, los que deben estar en correspondencia con los objetivos del grupo de desarrollo, se definen los estándares y guías que le dan validez al documento además de proponer una métrica encaminada a estudiar el comportamiento de los defectos en el proyecto.

En el Plan de Aseguramiento de la Calidad se plasman las revisiones a realizar en el proyecto donde se señala el momento exacto en el que corresponde realizarle cada revisión precisando las herramientas que son utilizadas para su ejecución. También se mencionan las acciones correctivas y preventivas como parte de la mejora de la calidad, el Plan de Pruebas y el listado de los registros de calidad que serán utilizados en el transcurso de vida del proyecto.

Para la confección de este plan se referencia una serie de documentos que se mencionan a continuación:

- Lineamientos de Calidad UCI.
- Plan de Gestión de Riesgos.
- Plan de Gestión de Configuración.
- Plan de Desarrollo de Software.
- Plan de Pruebas.

El propósito que se busca con este Plan de Aseguramiento de la Calidad es describir y especificar cómo se asegurará la calidad del Grupo de Alternativas Libres para Multimedia, con el objetivo de proveerle seguridad al producto a obtener y que los procesos del ciclo de vida del proyecto estén acorde con los requisitos específicos y coincidan con los planes establecidos. El plan incluye las revisiones y pruebas que se efectuarán en los diferentes períodos del proceso de desarrollo de GALM.

2.5.2. Verificación y validación del proceso de aseguramiento de la calidad

La validación o verificación del proceso de desarrollo de software pueden definirse como: (IEEE, 1988)

- *Verificación: comprobar si los productos construidos en una fase del ciclo de vida satisfacen los requisitos.*
- *Validación: comprobar si el software construido satisface los requisitos del usuario.*

Dentro de las actividades de verificación y validación ligadas al control de la calidad se incluyen las pruebas y procesos de revisión y auditorías.

Las revisiones son actividades de control de calidad, que permiten detectar defectos en los proyectos de software. Las Revisiones pueden ser de dos tipos, tal y como se muestra en la Figura 2, dinámicas y estáticas. Las primeras son las que detectan los defectos ejecutando el software, fundamentalmente son las ejecutadas en la fase de prueba del proyecto. Las segundas son visuales y se realizan sin necesidad de que el software esté ejecutándose. Ambas son de suma importancia y una combinación adecuada puede detectar gran cantidad de defectos y por tanto mejorar la calidad del producto final.



Figura 2. Tipos de Revisiones. (Jacobson, y otros, 2009).

Fuente: (Jacobson, y otros, 2009).

El Grupo de Alternativas Libres para Multimedia no cuenta con suficiente personal como para asumir las revisiones con rigurosidad, pues todos los integrantes del equipo se encuentran inmersos en las tareas de desarrollo del software que les fueron asignadas.

Es por ello que para detectar a tiempo los defectos y poder eliminarlos será necesario establecer un adecuado Plan de Revisiones, donde queden planificados los momentos en que se realizarán las revisiones durante el proceso de desarrollo, considerando en esta planificación tanto las ejecutadas por los propios desarrolladores, como las realizadas por los especialistas no involucrados en el desarrollo del proyecto.

A continuación se muestran un conjunto de pasos que deben tenerse en cuenta para el desarrollo del Plan de Revisiones:

1. Realizar las revisiones de acuerdo al Plan de Proyecto.
2. Definir el producto a revisar.

3. Definir día y hora para la revisión.
4. Determinar que es necesario y quien debe hacerlo.
5. Definir el tipo de revisión a realizar.
6. Identificar a las personas que deben participar e invitarlas indicándole su responsabilidad en la revisión.
7. Si se realiza una reunión definir quien la organizará. Desarrollar una agenda de la reunión.
8. Definir qué se debe hacer durante la revisión y quien debe hacerlo.
9. Definir los criterios de éxito para la revisión (cuando se puede decir que la revisión puede finalizar).
10. Identificar y almacenar los resultados que deben conservarse de la revisión.

Además se debe tener en cuenta el Modelo de Madurez de las Capacidades (CMM) propuesto en el capítulo anterior para guiar el proceso de desarrollo del software del proyecto que, contempla la introducción de revisiones como un requisito para pertenecer al segundo nivel de CMM, que avala la calidad del proceso de desarrollo en sus diferentes estados en el perfeccionamiento de la calidad del producto final.

Se recomienda realizar al menos las siguientes revisiones y auditorías durante el desarrollo del proyecto (Delgado, 2003):

- *Inspecciones del Colectivo: ejecutadas por el propio equipo de desarrollo, en este caso se recomiendan las revisiones de los requisitos, la arquitectura, el diseño y los ciclos de desarrollo (Jacobson, 2000). Por otro lado, las revisiones de la arquitectura y el diseño son las que revisan el diseño preliminar y crítico respectivamente, que persiguen el objetivo de detectar defectos antes de pasar a la fase de codificación, y por último las revisiones al finalizar cada ciclo de desarrollo detectan defectos presentes en los componentes y artefactos del proyecto desarrollados en cada uno de los ciclos.*
- *Revisiones dinámicas, en las que se deben ejecutar las actividades de evaluación y prueba del software, considerando en cada momento el tipo de prueba que se requiere.*

Detección y seguimiento de defectos (DSD)

Las actividades del proceso varían dependiendo del tipo de revisión que se efectúe, si se trata de una inspección se comenzará por designar un miembro del equipo de desarrollo, encargado de conducir todo el proceso hasta obtener el reporte final de la inspección y entregar al Administrador de la Calidad que se

encargará de emitir las peticiones de cambio correspondientes (Febles, 2001), si fuese necesaria una reinspección. En el caso de una revisión dinámica las actividades son las siguientes:

1. Designar el equipo de prueba o evaluación del software
2. Confeccionar casos de prueba y procedimientos de prueba o en su defecto las listas de comprobación correspondientes
3. Detectar y registrar los defectos a partir de la ejecución del software
4. Notificar al Administrador de la Calidad y al Líder del proyecto

La verificación, supone por lo tanto, un proceso de mejora continua para el Grupo de Alternativas Libres para Multimedia y que gracias a la gestión de configuración permite monitorizar en cualquier momento el grado de seguimiento de los procedimientos establecidos para el desarrollo del producto que garantizan su calidad.

2.6. Gestión de configuración del software

La gestión de configuración del software es una actividad de protección que se realizará a lo largo de todo el desarrollo de software del Grupo de Alternativas Libres para Multimedia. Considerando que el cambio puede producirse en cualquier momento, las actividades que plantea la gestión de configuración permitirán: (1) identificar el cambio; (2) controlar el cambio; (3) garantizar que el cambio se implementa adecuadamente; (4) informar del cambio a todos aquéllos a los que afecte.

Las actividades que componen los procesos de gestión de configuración se pueden agrupar en cuatro grandes grupos:

- **Identificación de elementos:** Proporciona guía básicas para la identificación y etiquetado de componentes software, configuraciones de software y líneas base de software. La fecha y la asignación de un número de versión son elementos de identificación imprescindibles que pueden completarse con otros calificadores.
- **Control de cambios:** Proporciona los mecanismos de control mediante los que ante una propuesta de modificación de algún elemento, se evalúa, aprueba o rechaza y se le da seguimiento. Se basa en cuatro elementos:

- ✓ Petición de Cambio.
- ✓ Análisis de impacto del cambio. Basado en las interdependencias con el resto de elementos de la configuración.
- ✓ Modificaciones y creación de nuevos componentes.
- ✓ Creación de nuevas líneas base con las modificaciones.

Para lograr y asegurar la calidad al usuario o cliente, es necesario controlar la calidad en cada una de las etapas de desarrollo, desde el diseño del producto o sistema hasta el mantenimiento. Estas etapas se constituyen como el ciclo de la calidad, habiendo interdependencia y retroalimentación en cada una de ellas (Qualitrain, 2008).

- **Control de versiones:** Es habitual que una misma aplicación coexista en diferentes versiones. El control de versiones proporciona mecanismos de control de los diferentes estados de evolución del software mediante etiquetas especiales que identifican a una configuración de elementos del resto.
- **Construcción de configuraciones.** Proporciona los elementos necesarios para la extracción desde el repositorio de todos los elementos que corresponden a una determinada configuración y permite de manera más o menos automatizada, o asistida por herramientas, la construcción del software desde sus archivos fuente.

De forma general, cuando se construya el software los elementos deben quedar identificados unívocamente. Una vez que esto se ha logrado, se procede a establecer los mecanismos para el control de cambio y finalmente se procede a elaborar los informes para asegurar que aquellos que necesitan conocer los cambios sean informados debidamente.

La gestión de configuración del software será llevada a cabo por un equipo de trabajo conformado por: Líder del Proyecto, Administrador de la Calidad y el Jefe de Módulo; se hará el uso de herramientas de control como el Subversión y Trac como herramienta de planificación.

2.7. Auditorías de calidad al software

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 (ISO, 2002).

Una auditoría de calidad para el proyecto tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. *Otro objetivo consiste en suministrar una evaluación objetiva del producto y los procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos* (Scalone, 2006).

Para realizar la evaluación de la calidad del software se considera la Norma ISO 9126-1:2001, estudiada en el capítulo anterior. Esta norma define las características de calidad a través de múltiples niveles de subcaracterísticas. La valoración de estas características es útil para que el usuario pueda definir los requerimientos del producto utilizando solamente las características que emplee en la práctica.

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

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

El esquema a seguir respecto al informe comprenderá varios requisitos respecto a estructura y contenido. Los puntos esenciales de un Informe de Auditoría son: (1) Identificación del Informe, (2) Identificación del Cliente, (3) Identificación de la Entidad auditada, (4) Objetivos de la Auditoría Informática, (5) Normativa aplicativa y excepciones, (6) Alcance de la Auditoría, (7) Conclusiones: Informe corto de opinión, (8) Resultado: Informe largo y otros informes, (9) Informe previo, (10) Fecha del Informe, (11) Identificación y firma del Auditor, (12) Distribución del Informe y (13) Conclusiones.

Además del Informe de Auditoría de Calidad se llevarán a cabo varios registros de calidad como documentos que guardarán información específica y relacionada a un procedimiento o instrucción de trabajo. Éstos comprueban que el proyecto cumple sus procedimientos y normas establecidas.

- Los tipos de registros que se guardarán serán:
- Actas de reuniones efectuadas.
- Listas de Chequeo aplicadas.
- Casos de Prueba de Aceptación.
- Los Informes de No Conformidades.
- Listas de errores encontrados en un artefacto.
- Órdenes de trabajo para los módulos del proyecto.
- Resúmenes de Resultados de las pruebas.
- Correos de trabajo y comunicación de intercambio de trabajo.

Todos los registros serán debidamente guardados en el repositorio del proyecto.

Este proceso de auditoría será realizado por el Administrador de la Calidad en la última semana de cada iteración con el objetivo de no pasar con no conformidades a la siguiente iteración planificada por el proyecto. Por último, se puede decir que la auditoría de la calidad del software puede traer aparejado encontrar a tiempo desviaciones en el proceso de desarrollo del software, lo cual permite mejorar el nivel de competitividad del proyecto con sus respectivas consecuencias.

Para mayor comprensión del proceso de auditorías se realizó un Diagrama de Actividades, pues a través del diagrama se puede observar en todos los casos, las actividades, artefactos y responsables de los mismos (ver Anexo 6).

2.8. Herramientas y técnicas a utilizar. Propuesta de acciones correctivas.

Las **herramientas y técnicas** que apoyan la estrategia de aseguramiento de la calidad se nombran a continuación:

- Se utiliza la unión de las metodologías Scrum y XP como guía en la definición de los roles y responsabilidades así como en la ejecución de las pruebas, el plan de prueba y el diseño de los casos de prueba de aceptación.
- Se aplican los Lineamientos de Calidad definidos por la Dirección de calidad del software de la UCI (ver Anexo 7); independientemente de que estos Lineamientos UCI que entrarán en vigor a partir del mes de mayo del presente año serán los que guíen la estrategia para propiciar la vigencia de la misma.
- Se utilizan las Plantillas del Expediente de Proyecto para registrar la documentación del proyecto.
- Se aplican las Listas de Chequeo como herramienta de medición a utilizar en el desarrollo de las pruebas.
- Se emplean las Revisiones Técnicas Formales como técnica para revisar los artefactos producidos por la Ingeniería de Software.
- Se emplea SVN para la gestión de proyectos y el control de versiones.

Propuesta de acciones correctivas asociada a la estrategia que se describe en esta investigación:

- Revisión de los procesos (políticas y procedimientos).
- Capacitación al equipo de desarrollo, y en general a todos los integrantes del proyecto mediante dos cursos para familiarizar al desarrollador con las herramientas y el sistema operativo que utiliza el proyecto.
- Reorganizar, reubicar a las personas y recursos dentro del proyecto.
- Mejorar, reforzar las actividades de auditoría (ver Anexo 8).
- Reforzar el tema del control a la asignación de responsabilidades dentro del equipo de trabajo para garantizar que no quede nada por hacer y que cada cual tenga su responsabilidad claramente definida.
- Realizar proceso de revisión y prueba más rigurosa y eficiente, con el objetivo de detectar lo antes posible, los posibles defectos antes de que pasen a etapas posteriores.
- Tomar medidas con los integrantes del equipo que no cumplieran con lo establecido por sus superiores, para evitar que estos tomaran decisiones por su cuenta.

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

2.9. Conclusiones del capítulo

El Grupo de Alternativas Libres para Multimedia de la UCI, se encuentra desarrollando un IDE y para ello se planificó un proceso de desarrollo que se ve afectado por no contar con una estrategia que garantice la calidad del proceso y el producto que en este proyecto tienen lugar. El objetivo de plantear estrategias de aseguramiento de la calidad es ayudar a los Ingenieros de software a estandarizar los procesos de desarrollo, logrando de esta manera una mayor eficiencia en la producción de software.

Durante el desarrollo de este trabajo de diploma se realizó un análisis del área de procesos aseguramiento de la calidad al Modelo de Madurez y Capacidad Integrado, lo cual permitió diseñar y aplicar una estrategia para el aseguramiento de la calidad del Grupo de Alternativas Libres para Multimedia, dando cumplimiento así, al objetivo planteado en esta investigación. Se definió, además, como parte de esta estrategia, el proceso de evaluación a aplicar al producto que se desarrolla en el proyecto para chequear que el mismo satisfaga las necesidades planteadas por el cliente.

Validación de la Estrategia de Aseguramiento



Validación de la Estrategia de Aseguramiento

3.1. Introducción

En el presente capítulo se realiza la validación de la Estrategia de aseguramiento de la calidad para el Grupo de Alternativas para Multimedia. Teniendo en cuenta el problema planteado específicamente para Picassus permitió corroborar la idea de que si se aplica apropiadamente esta estrategia se podría obtener un producto con adecuados niveles de calidad. También corregir y perfeccionar el sistema de acciones propuesto, pues todos estos criterios fueron incorporados al cuerpo del trabajo y sirvieron en la elaboración final de la estrategia, la cual quedó satisfactoriamente validada.

Esta Estrategia de aseguramiento permite garantizar la calidad del producto lo que da la oportunidad de indicarle a los desarrolladores en qué medida marcha el proyecto y las acciones que deben realizarse para corregir los errores y deficiencias encontradas y poder avanzar con más rapidez, todo lo anterior encaminado a obtener un producto que cumpla con las exigencias del cliente.

La propuesta de solución proporciona múltiples ventajas al desarrollo del software, dentro de ellas trabajar bajo la guía de estándares de codificación o documentación que apoyan a uniformizar y consolidar el proceso de desarrollo. Además verifica que los objetivos individuales vayan acordes con los objetivos del proyecto lo que incrementa la productividad y dirige al grupo de desarrollo hacia la competitividad, lo cual es de vital importancia para las actuales organizaciones.

3.2. Resultados obtenidos de la aplicación de la Estrategia en el proyecto

A continuación se muestran los resultados obtenidos de la aplicación de la Estrategia de aseguramiento de la calidad, aplicada a la primera y segunda iteración del proyecto con el propósito de auditar todos los artefactos desarrollados hasta el momento y analizar el resultado de las pruebas correspondientes. Todo esto permitirá validar la estrategia y por consiguiente conocer la efectividad de la misma.

3.2.1. Resultado de la aplicación de auditorías

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

Identificación del producto software a auditar para la primera iteración.

Módulo	Fecha de Inicio	Fecha de Fin
Picassus	2/04/2009	9/04/2009
Revisión de la documentación	2/04/2009	4/04/2009
Solución a las no conformidades detectadas	4/04/2009	9/04/2009

Elementos de la tabla	Descripción
Lineamiento	Define el lineamiento de calidad al que hace referencia
Indicador a evaluar	Es la descripción del indicador que se pretende verificar
Evidencia ¿Qué?	Es la descripción de qué se quiere verificar
Nivel de Importancia	Es el nivel de importancia de la Observación: Alta (A), Media (M), Baja (B).
Ubicación ¿Dónde?	Especifica el lugar del expediente de proyecto, la plantilla o documento donde se puede verificar el indicador a evaluar.
Procedimiento ¿Cómo?	Especifica el procedimiento mediante el cual se pueden hacer las verificaciones.
Observaciones	Hallazgo encontrado.

Tabla 7. Aplicación de Auditoría para la Iteración 1.

Fuente: Elaboración Propia

Lineamiento	Indicador	Evidencia	N.I.	Ubicación	Procedimiento	Observaciones
Iteración:1 Fase: Planificación						
Concepción del Sistema v1.2						
8.3.2	Definición del proyecto	¿Existe el documento Concepción del Sistema?	A	Documento Concepción del Sistema	Revisión de documentos	El documento Concepción del Sistema existe, específicamente en el repositorio del proyecto.
8.4.2	Establecimiento de la Gestión de Configuración	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Documento Concepción del Sistema	Revisión de documentos	El registro de control de versiones de los cambios del documento se realiza.
8.3.2	Completamiento del expediente de proyecto	¿Se define el alcance y los objetivos del documento?	M	Documento Concepción del Sistema	Revisión de documentos	El alcance y los objetivos del documento quedan bien definidos dentro del mismo.
8.3.2	Definición del proyecto	¿Se definen claramente las oportunidades de negocio del proyecto?	A	Documento Concepción del Sistema	Revisión de documentos	Las oportunidades de negocio del proyecto no quedan bien definidas. Es necesario detallar estas oportunidades y consecuentemente queden plasmadas en el documento.
8.3.2	Definición del proyecto	¿Se define el problema claramente?	A	Documento Concepción del Sistema	Revisión de documentos	El problema queda definido claramente dentro del documento.
8.3.2	Definición del proyecto	¿Se declara la posición del producto donde se detallan las áreas donde este producto pretende incidir?	A	Documento Concepción del Sistema	Revisión de documentos	No se detallan las áreas donde este producto debe incidir.
8.3.2	Definición del proyecto	¿Se realiza una descripción de todos los usuarios y los involucrados que incluya estudio de mercado, entorno de usuario y perfil de los involucrados con detalle de cada uno de ellos?	A	Documento Concepción del Sistema	Revisión de documentos	No se realiza una descripción de los involucrados debido a las características del proyecto. Si se realiza una descripción de los desarrolladores y de los roles que ocupan. No se realiza un estudio de mercado.
	Definición del proyecto	¿Se evalúan las necesidades de los involucrados?	A	Documento Concepción del Sistema	Revisión de documentos	No se evalúan las necesidades de los involucrados.

8.3.2	Definición del proyecto	¿Se identifican y evalúan las alternativas y competencias a las que pueden acceder los usuarios?	M	Documento Concepción del Sistema	Revisión de documentos	No se realiza debido a las características del proyecto.
8.3.2	Definición del proyecto	¿Se realiza una visión general del producto que incluye perspectivas, capacidades, suposiciones y dependencias?	A	Documento Concepción del Sistema	Revisión de documentos	Se realiza una visión general del producto que contiene las características detalladas.
	Definición del proyecto	¿Se describen las restricciones de costo y precio del producto? (En caso que lo requiera)	A	Documento Concepción del Sistema	Revisión de documentos	El proyecto no requiere de realizar estimaciones de costos ni restricciones.
	Definición del proyecto	¿Se describen las características básicas del producto?	A	Documento Concepción del Sistema	Revisión de documentos	Se describen detalladamente las características básicas del producto.
Historias de Usuario v1.0						
8.2.1	Gestión de los requisitos	¿Existe(n) el(los) documento de Historias de Usuario?	A	Historias de Usuario	Revisión de documentos	El documento Historias de Usuario existe, específicamente en el repositorio del proyecto.
8.4.2	Establecimiento de la Gestión de Configuración.	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Historias de Usuario	Revisión de documentos	El registro de control de versiones de los cambios del documento se realiza.
8.2.2	Completamiento del expediente de proyecto.	¿Se define el alcance del documento?	B	Historias de Usuario	Revisión de documentos	No se define el alcance del documento.
8.2.2	Desarrollo de los requisitos	¿Están descritas todas y cada una de las Historias de Usuarios?	A	Historias de Usuario	Revisión de documentos	Todas las Historias de Usuario están descritas en el documento.
8.2.2	Desarrollo de los requisitos	¿Las Historias de Usuario están descritas con suficiente detalle que permiten determinar Usuario, Nombre de Historia, Prioridad, Riesgo, Puntos estimados, Iteración Asignada, Responsable y Descripción ?	M	Historias de Usuario	Revisión de documentos	Las Historias de Usuario se encuentran suficientemente detalladas excepto el Usuario por las características del proyecto.
Lista de Riesgos v1.0						
8.3.7	Gestión de los riesgos del proyecto	¿Existe el documento "Lista de Riesgos"?	A	Lista de Riesgos	Revisión de documentos	El documento Lista de Riesgos existe, específicamente en el repositorio del proyecto.
8.4.2	Establecimiento de la Gestión de la Configuración	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Lista de Riesgos	Revisión de documentos	El registro de control de versiones de los cambios del documento se realiza.
8.4.2	Completamiento del expediente de proyecto.	¿Se define el alcance y objetivos del documento?	M	Lista de Riesgos	Revisión de documentos	No se definen ni el alcance ni los objetivos del documento.

8.3.7	Gestión de los riesgos del proyecto	¿Existe una lista de riesgos o referencia a la misma?	A	Lista de Riesgos	Revisión de documentos	Si existe una lista de riesgos en el documento.
8.3.7	Gestión de los riesgos del proyecto	¿Están especificados cada uno de los riesgos de manera que sea posible determinar indicadores, estrategia de mitigación, plan de contingencia?	A	Lista de Riesgos	Revisión de documentos	Los riesgos se encuentran bien especificados donde es posible determinar indicadores y la mitigación de cada riesgo, pero no se establece un plan de contingencia.
Modelo de Diseño v1.0						
8.2.5	Definición del Modelo de Diseño	¿Existe el documento "Modelo de Diseño"?	A	Modelo de Diseño	Revisión de documentos	El documento Modelo de Diseño existe, específicamente en el repositorio del proyecto.
8.4.2	Establecimiento de la Gestión de la Configuración	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Modelo de Diseño	Revisión de documentos	El registro de control de versiones de los cambios del documento se realiza.
8.4.2	Completamiento del expediente de proyecto.	¿Se define el alcance del documento?	M	Modelo de Diseño	Revisión de documentos	No se define el alcance del documento.
8.2.5	Definición del Modelo de Diseño	¿Se encuentra un diagrama de paquetes donde se observa el sistema organizado en pequeños paquetes y subsistemas?	M	Modelo de Diseño	Revisión de documentos	En el documento se encuentra un diagrama de paquetes donde se observa el sistema organizado.
8.2.5	Definición del Modelo de Diseño	¿Se encuentra un diagrama de clases donde se observan las interacciones entre ellas?	A	Modelo de Diseño	Revisión de documentos	Se encuentra un diagrama de clases pero debe ser mejorado para una mejor comprensión.
8.2.5	Definición del Modelo de Diseño	¿Se realiza la descripción de todas y cada una de las clases que intervienen en el diagrama de clases con suficiente detalle que permitan determinar la capa a la que pertenecen, módulo , número de la clase, nombre de la clase, propósito y descripción ?	A	Modelo de Diseño	Revisión de documentos	No se realiza la descripción pertinente para cada una de las clases.
Plan de Capacitación v1.0						
8.1.1	Establecimiento del Plan de Capacitación del Personal	¿Existe el documento "Plan de Capacitación"?	A	Plan de Capacitación	Revisión de Documentos	No existe un documento específico para el Plan de Capacitación establecido en el proyecto, sólo dos P1 de los cursos que actualmente se imparten en el proyecto para sus desarrolladores.
8.4.2	Establecimiento de la Gestión de la Configuración	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Plan de Capacitación	Revisión de Documentos	No se realiza debido a la no existencia del documento.
8.4.2	Completamiento del expediente de proyecto.	¿Se define el alcance del documento?	M	Plan de Capacitación	Revisión de Documentos	No se define debido a la no existencia del documento.
8.1.1	Establecimiento del	¿Se realiza una justificación del plan de capacitación?	B	Plan de	Revisión de	No se realiza debido a la no existencia del

	Plan de Capacitación del Personal			Capacitación	Documentos	documento.
8.1.1	Establecimiento del Plan de Capacitación del Personal	¿Se define el alcance , los objetivos , las metas , estrategias , acciones , recursos y presupuesto del plan de capacitación?	M	Plan de Capacitación	Revisión de Documentos	No se definen debido a la no existencia del documento.
8.1.1	Establecimiento del Plan de Capacitación del Personal	¿Se define un cronograma de las actividades de capacitación?	M	Plan de Capacitación	Revisión de Documentos	No se define debido a la no existencia del documento.
Minuta de Reunión v1.0						
8.3.9	Registro de acuerdos de trabajo y minutas de reunión	¿El proyecto realiza el registro de las actividades conjuntas y acuerdos de trabajo?	A	Minutas de Reuniones	Revisión de Documentos	Se realiza el registro de actividades conjuntas y acuerdos de trabajo.

Establecer las acciones correctivas respecto a los requisitos evaluados

Los errores o incongruencias que se detectaron durante las auditorías se reportaron en un documento de No Conformidades habilitado para el caso (ver Anexo 9). Este documento se le hizo llegar al equipo de desarrollo y al líder del proyecto para solucionar los problemas señalados.

Elaboración de un Informe Final

Se elaboró un Informe Final de la auditoría (ver Anexo 10) como parte del cumplimiento de la Estrategia de aseguramiento de la calidad que en esta investigación se establece.

Una vez concluida la actividad de revisión de los documentos se procedió a solucionar todas las no conformidades encontradas con el objetivo de iniciar la siguiente iteración con la máxima garantía de un software con altos niveles de calidad.

Como parte de las no conformidades se encontró que no se había realizado un estudio de mercado lo que fue solucionado con la orientación a un integrante del equipo de desarrollo realizar la investigación correspondiente. Se detectó además la no existencia de un plan de contingencia ante la ocurrencia de los riesgos previstos, esta no conformidad fue erradicada con la creación del plan antes mencionado.

Otro de los problemas detectados fue la poca comprensión de los diagramas de clases realizados para el Modelo de Diseño lo que dificultaba el entendimiento del mismo para los desarrolladores como para una persona ajena al proyecto; para ello se realizaron nuevamente los diagramas siendo de ayuda considerable en el proceso de desarrollo del software.

Identificación del producto software a auditar para la segunda iteración.

Módulo	Fecha de Inicio	Fecha de Fin
Picassus	4/05/2009	9/05/2009
Revisión de la documentación	4/05/2009	6/05/2009
Solución a las no conformidades detectadas	6/05/2009	9/05/2009

Tabla 8. Aplicación de Auditoría para la Iteración 2.

Fuente: Elaboración Propia

Lineamiento	Indicador	Evidencia	N.I.	Ubicación	Procedimiento	Observaciones
Iteración:2 Fase: Desarrollo						
Glosario de Términos v1.0						
8.2.9	Establecimiento del glosario de términos del proyecto	¿El proyecto posee un glosario con todos los términos de difícil comprensión así como con las abreviaturas utilizadas en el resto de los documentos?	A	Glosario de Términos	Revisión de Documentos	El proyecto posee un documento Glosario de Términos.
Tareas de Ingeniería v1.0						
8.3.4	Desarrollo del Plan de Proyecto	¿El proyecto mantiene los registros de las tareas de Ingeniería ejecutadas durante el desarrollo del proyecto?	A	Tareas de Ingeniería	Revisión de Documentos	El proyecto mantiene los registros de las tareas de ingeniería ejecutadas pero no se encuentran lo suficientemente desglosadas por cada una de las Historias de Usuario.
Plan de Aseguramiento v1.0						
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Existe el documento "Plan de Aseguramiento de la Calidad"?	A	Plan de Aseguramiento de la Calidad	Revisión de Documentos	El documento Plan de Aseguramiento existe, específicamente en el repositorio del proyecto.
8.4.2	Establecimiento de la Gestión de la Configuración	¿Se realiza el registro de control de versiones de los cambios del documento?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	El registro de control de versiones de los cambios del documento se realiza.
8.4.2	Completamiento del expediente de proyecto.	¿Se define el alcance y el propósito del documento?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se encuentra bien definido el propósito y el alcance del documento.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Están claramente definidos los objetivos de calidad?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se encuentran claramente definidos los objetivos de calidad.
8.1.3	Definición de Roles y Responsabilidades	¿Se describe la estructura organizacional de todos los responsables de la calidad así como las tareas y los responsables?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se describe la organización así como los responsables de la calidad y sus tareas.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Se describen las métricas utilizadas como parte del monitoreo y control del trabajo?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se describen las métricas utilizadas.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Se referencian cada uno de los estándares y guías utilizados por el Plan de Calidad?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se referencian los estándares y guías utilizados por el Plan.

8.2.8	Seguimiento de elementos no conformes	¿Se describen los tipos de revisiones y auditorías planificadas en el proyecto así como un cronograma con la planificación de las mismas?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se realiza la descripción los tipos de revisiones y auditorías, así como el cronograma con la planificación de las mismas.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Se identifican los involucrados o los grupos específicos que intervienen en las revisiones y auditorías programadas?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se identifican los involucrados que intervienen en las revisiones y auditorías.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Existe el procedimiento o referencia al mismo de las tareas a realizar para la corrección de los defectos detectados?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Existe un acápite para las acciones correctivas.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Existe un plan de pruebas o referencias al mismo?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	No existe un plan de pruebas pero si referencia al mismo.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Se describen las herramientas, técnicas y metodologías a utilizar durante las revisiones y auditorías?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	Se describen las herramientas, técnicas y metodología a utilizar.
8.4.1	Establecimiento del Plan de Aseguramiento de la Calidad	¿Se describen los registros que se mantendrán durante el proyecto, incluyendo su ubicación?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	No se describen los registros.
8.1.1	Establecimiento del Plan de Capacitación del Personal	¿Se describen las actividades de formación o capacitación necesarias para que el personal responsable ejecute las tareas correspondientes del Plan de Aseguramiento de la Calidad?	M	Plan de Aseguramiento de la Calidad	Revisión de Documentos	No se describen estas actividades.

Actualmente el Líder del proyecto así como el equipo de desarrollo se encuentran inmersos en la tarea de darle solución a los problemas encontrados en la realización de la auditoría correspondiente a la segunda iteración. Una de las dificultades detectadas fue que no se desglosaron lo suficiente varias tareas de ingeniería por cada Historia de Usuario lo que provocó la no culminación de la Historia de Usuario en la iteración correspondiente debido a su extensión y complejidad.

Es importante destacar que aunque la cantidad de documentos revisados en la segunda iteración es considerablemente menor que la anterior se obtuvieron mejores resultados ya que los documentos fueron elaborados con mayor calidad y los errores detectados disminuyeron considerablemente. Esto demuestra en gran medida que la estrategia planteada por esta investigación es factible.

3.2.2. Resultado de las pruebas

La realización de las pruebas fue guiada por la Estrategia de Prueba Orientada a Objetos detallada en el capítulo anterior, la cual integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba: planificación, el diseño de casos de prueba de aceptación, la ejecución y los resultados, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software.

Resultado de las pruebas para la primera iteración

En esta iteración fueron desarrolladas las siguientes Historias de Usuario:

- Exportar Biblioteca con Imágenes
- Observar Propiedades de los elementos de la biblioteca

Las Historias de Usuario presentadas anteriormente fueron evaluadas durante todo el proceso de pruebas pero en el presente trabajo de diploma solo se detallará la Historia de Usuario Exportar Biblioteca con Imágenes debido a la extensión de la tesis y las otras serán registradas en un documento en el Expediente de Proyecto.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Exportar Biblioteca con Imágenes.
Modificación de Historia de Usuario Número:	
Usuario: Yonnys Pablo Martin Olivera	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	Puntos Reales: 2
Descripción: Se realiza un recorrido por todos los elementos de la biblioteca obteniendo sus datos, además se recorren las escena verificando cuales están instanciados y obteniendo sus respectivos datos, con los cuales se conformara un XML que se salvará en disco. Después de obtenido este fichero se ejecuta el programa "swfmill" con los parámetros correspondientes para obtener un .SWF con los elementos imágenes de la biblioteca	
Observaciones: La Historia de Usuario no presenta ningún señalamiento.	
Prototipo de interfaces: Por motivos de espacio de esta investigación (ver Anexo 11).	

Se realizó la asociación de las técnicas de prueba con todos los requisitos no funcionales para esta Historia de Usuario. Sin embargo, a manera de ejemplo para este trabajo solo fue seleccionado uno de ellos. A continuación se evalúan explícitamente las características de calidad para cada una de las pruebas correspondientes al RNF de fiabilidad:

De los resultados obtenidos se procedió a utilizar las Listas de Chequeo definido como instrumento de medición en el capítulo anterior.

Característica	Pregunta	Evaluación
Madurez	1) ¿Se realizó un proceso de levantamiento de información apropiado?	4= Bueno
	2.- ¿El levantamiento de información fue establecido formalmente?	5= Si
Correctitud	1.- ¿Están siendo considerados todos los procesos necesarios para dar solución al problema?	5= Si

Registro de los Casos de Prueba de Aceptación realizados para dar respuesta a las Listas de Chequeo

Caso de Prueba de Aceptación	
Código Caso de Prueba: P-01-01	Nombre Historia de Usuario: Exportar Biblioteca con Imágenes
Nombre de la persona que realiza la prueba: Roberto Ferrer Obregón	
Descripción de la Prueba: Se procederá a agregar imágenes a la biblioteca del proyecto y se generará un SWF con estos elementos en la biblioteca.	
Condiciones de Ejecución: Deben agregarse elementos a la biblioteca	
Entrada / Pasos de ejecución: Se adicionan varios elementos a la biblioteca. Se selecciona la opción Exportar SWF y se genera un archivo de este tipo con los elementos en la biblioteca.	
Resultado Esperado: Se obtiene un SWF que tiene en la biblioteca todos los elementos agregados.	
Evaluación de la Prueba: Satisfactorio	

Los resultados cuantitativos se encuentran en base a las respuestas dadas en las Listas de Chequeo y por la importancia que los involucrados asignan a cada subcaracterística. Lo anterior se expresa en la ponderación de los resultados mediante una planilla. A continuación se muestra la planilla de ponderación de resultados para la característica de Fiabilidad.

Característica de Calidad: Fiabilidad					
Subcaracterística	Peso Asignado				
	1	2	3	4	5
Madurez				X	
Tolerancia a Fallas		X	X	X	X
Recuperabilidad				X	
Correctitud			X	X	X
Estructurado					X
Encapsulado			X		

Teniendo en cuenta la Estrategia de Prueba para el procesamiento de las respuestas dadas en las Listas de Chequeo, se generan tres (3) tipos de resultados:

1) Resultados de la presencia de las subcaracterísticas en cada etapa del proceso de prueba, según la característica de calidad a la que corresponde.

- La subcaracterística de Madurez se presenta medianamente.
- La subcaracterística de Tolerancia a Fallas se presenta medianamente.
- La subcaracterística de Recuperabilidad se encuentra presente.
- La subcaracterística de Correctitud se encuentra presente.
- La subcaracterística de Estructurado se encuentra altamente presente.
- La subcaracterística de Encapsulado se presenta medianamente.

2) Resultados de la Presencia de las Características de Calidad (PCC) en cada una de las etapas, considerando la importancia dada por los involucrados.

De acuerdo a la fórmula utilizada se obtuvo un resultado de 3.9 en la escala del 1 al 5 establecida. Este resultado fue llevado a porcentaje con el fin de identificar si el mismo tiene nivel de aceptabilidad obteniendo un 78 % por lo que la característica de Fiabilidad se encuentra presente.

3) Después de realizar la evaluación de cada una de las características de calidad se obtuvo el Porcentaje de Presencia de cada una de las Características de Calidad en todo el Sistema (PPCS), se obtuvo un 75.5 % de nivel de aceptabilidad por lo que se considera eficiente la evaluación que se está realizando.

Resultado de las pruebas para la segunda iteración

En esta iteración fueron desarrolladas las siguientes Historias de Usuario:

- Funcionalidades de Edición de elementos.
- Menú Recientes
- Dialogo de Inicio
- Exportar figuras y trazos de un proyecto general

Las Historias de Usuario presentadas anteriormente fueron evaluadas durante todo el proceso de pruebas perteneciente a la segunda iteración del proyecto. Para cada una de ellas se definió la plantilla de Historia de Usuario así como sus Casos de Prueba de Aceptación correspondiente. A continuación a manera de ejemplo se muestra la Historia de Usuario Funcionalidades de Edición de elementos:

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Funcionalidades de Edición de elementos.
Modificación de Historia de Usuario Número:	
Usuario: Yonnys Pablo Martin Olivera	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	Puntos Reales: 2
Descripción: Al seleccionar un elemento del escenario se permitirán acceder a las opciones de copiar, cortar, enviar al fondo y traer al frente tanto desde el menú principal como a través de el menú contextual, así como solo se activara la función de Pegar cuando el porta papeles se encuentre ocupado.	
Observaciones: La Historia de Usuario no presenta ningún señalamiento.	
Prototipo de interfaces: Por motivos de espacio de esta investigación (ver Anexo 12).	

Para esta Historia de Usuario se desarrollaron varios Casos de Prueba de Aceptación como por ejemplo:

Caso de Prueba de Aceptación	
Código Caso de Prueba: P-03-01	Nombre Historia de Usuario: Funcionalidades de Edición de elementos.
Nombre de la persona que realiza la prueba: Roberto Ferrer Obregón	
Descripción de la Prueba: Se procederá a aplicar combinaciones de teclas de las acciones de pegar, y eliminar sin que antes se haya seleccionado ningún elemento de la escena ni se hayan guardado en el clipboard ningún elemento.	
Condiciones de Ejecución: Que existan elementos en la escena en al menos un fotograma.	

Entrada / Pasos de ejecución: Se adicionan elementos a la escena. Se presiona la combinación de teclas Ctrl+V en una parte de la escena vacía. Se presiona también en la escena la combinación de teclas Del en una parte de la escena vacía.
Resultado Esperado: El programa siga funcionando sin que se altere el número de elementos en la escena.
Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación	
Código Caso de Prueba: P-03-02	Nombre Historia de Usuario: Funcionalidades de Edición de elementos.
Nombre de la persona que realiza la prueba: Roberto Ferrer Obregón	
Descripción de la Prueba: Se procederá a pegar elementos en todos los fotogramas de la aplicación.	
Condiciones de Ejecución: Que existan elementos en la escena en la menos un fotograma. Y estén creados al mínimo 200 fotogramas.	
Entrada / Pasos de ejecución: Se seleccionara un elemento de la escena. Se copiara o cortara de manera aleatoria. Se procederá a pegarlo en el fotograma siguiente del cual se volverá a seleccionar el elemento pegado, se copiara o cortara de manera aleatoria y se realizará la acción anterior	
Resultado Esperado: Se pegue el elemento guardado en el clipboard cada vez que se presione la combinación Ctrl+V y que al cortarlo se proceda correctamente.	
Evaluación de la Prueba: No satisfactoria. A los 200 fotogramas la aplicación se cierra. El error obtenido es: ASSERT failure in QList<T>::at: "index out of range", file /usr/include/qt4/QtCore/qlist.h, line 395	

Al igual que en la iteración anterior se utilizaron las Listas de Chequeo como instrumento de medición y luego a partir del resultado de los Casos de Prueba de Aceptación se realizó la ponderación de los mismos para evaluar la presencia de las características de calidad en el software donde se obtuvo el Porcentaje de Presencia de cada una de las Características de Calidad en todo el Sistema (PPCS), se obtuvo un 78 % de nivel de aceptabilidad por lo que se considera eficiente la evaluación que se realizó.

Los errores de las pruebas fueron corregidos al mismo tiempo que fueron encontrados para dar cumplimiento a lo que establece la metodología que todos los defectos deben ser corregidos en el momento en que son detectados para evitar que se extienda el tiempo de desarrollo y por consiguiente la iteración en que se encuentra el proyecto.

3.3. Análisis de los resultados

Es importante resaltar que a pesar de que el nuevo expediente de proyecto propuesto en esta investigación, no ha sido aplicado en su totalidad y actualmente se dan los primeros pasos con la metodología utilizada, se han obtenido resultados con la aplicación de muchas de las ideas que se tuvieron en cuenta para su realización.

Con la utilización de la metodología ScrumXP en el proyecto, nació la necesidad de buscar una nueva forma de documentar el producto que se desarrolla, de manera que respondieran al proceso de desarrollo de las metodologías ágiles. Es así como se comenzó a utilizar muchas de las plantillas que han sido incluidas en este expediente, para que el proceso de documentación de software fluyera en el proyecto, tales como:

- Concepción del sistema, permitió resumir los aspectos más relevantes del sistema, así como realizar una concepción inicial.
- Historia de usuario, permitió ratificar que en ella se resumía todo lo referente a las especificaciones de los requerimientos, así como lo relacionado con los casos de uso, conocidos del proceso unificado.

Con la utilización de estas plantillas se alcanzaron resultados significativos, como son:

- La entrega en tiempo y forma de la documentación mínima y necesaria.
- Un proceso de documentación menos tedioso.

Precisamente estos resultados se convirtieron en el incentivo para el desarrollo de esta investigación, en busca de un enriquecimiento del proceso de recopilación de información en el proyecto, teniendo como premisa que con una mejor organización de la documentación, así como su adaptabilidad a la metodología utilizada, se obtendrán resultados superiores a los antes expuestos.

La realización de la primera auditoría realizada al proyecto que se refleja en este trabajo de diploma ayudó al equipo de desarrollo a tomar decisiones que optimicen sus activos de software. Tomando como precedente que la mayoría de los documentos auditados se encontraban incompletos y algunos no existían en el repositorio, es por eso que se decidió por parte del Líder de Proyecto reforzar las actividades para el control de la calidad de los artefactos correspondientes a cada desarrollador. Además de que es importante señalar que la segunda auditoría arrojó mejores resultados que la primera demostrando la viabilidad y la importancia de la Estrategia de aseguramiento para el proyecto.

El proceso de pruebas desarrollado en ambas iteraciones obtuvo mejores resultados que las auditorías realizadas, porque aunque se detectaron defectos mediante la puesta en práctica de los diferentes tipos de prueba al analizar la evaluación de todas las características de calidad se obtuvo más de un 75 % nivel de aceptabilidad lo que demuestra la presencia de la calidad en el desarrollo del software hasta el momento del software.

Teniendo en cuenta la utilización del modelo de evaluación CMMI v1.2, que permite identificar problemas potenciales antes de que ocurran y las actividades para manejar los riesgos, se realizó la evaluación del software. Se considera que el nivel de madurez del proyecto se encuentra orientado hacia un nivel 1 o Inicial de CMMI porque aún no se tienen bien estructurados los procesos dentro del equipo de trabajo y el desarrollo es desordenado debido a la inconsistencia de la documentación, además de la reanudación de los riesgos. Esta evaluación permitió conocer el estado en que se encuentra el proyecto y así determinar una serie de aspectos para lograr mejores resultados y un producto con la calidad máxima requerida.

Para lograr avanzar hasta el siguiente nivel de madurez se establecen una serie de pautas:

- Perfeccionar la dirección del proceso de desarrollo del software a la Administración básica del proyecto teniendo en cuenta las diferentes áreas de procesos:
 - ✓ Administración de requerimientos: Identificar las inconsistencias entre estos requisitos y los planes del proyecto y productos de trabajo.
 - ✓ Planificación del proyecto: Mantenimiento de los planes que definan las actividades del proyecto.
 - ✓ Monitorización y Control del proyecto: Proporcionar una mejor comprensión de la evolución del proyecto a fin de que las medidas correctivas adecuadas puedan tomarse cuando el proyecto se desvíe significativamente de los resultados del plan.

- ✓ Medición y Análisis: Sostener una capacidad de medición que sea usada para soportar las necesidades de información de la administración.
- ✓ Aseguramiento de la calidad de los procesos y productos: Proporcionar la capacitación del personal sobre el aseguramiento de la calidad con visión objetiva dentro de los procesos y productos de trabajo asociados.
- ✓ Gestión de Configuración: Mantener la integridad de productos de trabajos con mayor rigurosidad usando:
 - Identificación de configuración.
 - Control de configuración.
 - Conteo del estado de configuración y auditoría de la configuración.

La evaluación del proyecto mediante CMMI v1.2 proporcionó al equipo de desarrollo y la administración un entendimiento de los objetivos de los procesos de trabajo y el correspondiente producto.

3.4. Conclusiones del capítulo

En este capítulo se puso en práctica la Estrategia de aseguramiento de la calidad para el producto software Picasso. Se mostraron los resultados arrojados por las auditorías realizadas y las pruebas teniendo en cuenta las características de calidad. Además del análisis posterior de los mismos, el cual indicó el nivel de madurez del Grupo de Alternativas Libres para Multimedia según el modelo de evaluación CMMI v1.2. Además de que se expusieron las principales deficiencias detectadas dándole solución a las mismas y las posibles acciones correctivas a tener en cuenta para etapas posteriores.

Se concluyó que una estrategia que tenga en cuenta las características y necesidades del proyecto, que contenga además un detallado Plan de Aseguramiento donde se describan correctamente los pasos a seguir para realizar una auditoría y un eficaz proceso de pruebas, mostrará el nivel de calidad que puede tener el producto final y con esto dar respuesta a la interrogante de cómo saber cuándo se puede determinar un nivel adecuado de calidad.

CONCLUSIONES

Culminando el desarrollo de la estrategia se dio cumplimiento, de forma satisfactoria, a los objetivos que se perseguían inicialmente:

- Se definió el proceso de aseguramiento de la calidad mediante el estudio de otras soluciones existentes y sus características correspondientes.
- Se creó una propuesta de solución teniendo en cuenta las características del proyecto, la metodología de desarrollo, la definición de las métricas y cada uno de los pasos a seguir por la estrategia planteada.
- Se logró crear un Plan de Aseguramiento con todas las características de la estrategia así como una Estrategia de Pruebas que permitiera la evaluación del proyecto.
- Se logró la aplicación y validación de la Estrategia de aseguramiento mediante la realización de auditorías y pruebas a la primera y segunda iteración del proyecto.
- Se logró confeccionar un Informe de No Conformidades que dieron paso al establecimiento de acciones correctivas con el fin de guiar al proyecto hacia la construcción de un producto con altos niveles de calidad.

Se concluye que la Estrategia de aseguramiento de la calidad brinda una solución factible a la problemática inicial y su desarrollo significará una mejora considerable en el desarrollo del producto Picasso del Grupo de Alternativas Libres para Multimedia; dándose origen a una estrategia base para el desarrollo de otras soluciones para proyectos de la UCI y un paso de avance para la búsqueda de la calidad total.

RECOMENDACIONES

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

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

REFERENCIAS BIBLIOGRÁFICAS

- Antonio, A. 2007.** “La gestión de la configuración de la software”. [En línea] 2007. [Citado el: 1 de Marzo de 2009.] http://www.ls.fi.upm.es/udis/docencia/plani/G_Configuracion.pdf.
- Benturia, R. 1998.** La calidad total y estrategia. Un camino a recorrer. *Boletín de Estudios Económicos*. 1998. Vol. XLVI, 143.
- CMMI. 2006.** Software Engineering Institute, CMMI® for Development, Version 1.2. [En línea] Software Engineering Institute Pittsburgh, 2006. [Citado el: 2 de Marzo de 2009.] <http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html>.
- Cueva, Juan M. 1999.** Calidad del Software. Oviedo : s.n., 1999.
- Delgado, M. 2003.** “*Inspecciones a Proyectos de Software, Garantía de Calidad*”, *Memorias de Informática 2003*. La Habana, Cuba. : s.n., 2003.
- Díaz, H. y Delgado, L. 2007.** Propuesta de Gestión de Calidad Interna para el proyecto Registro y Notarías . Ciudad de La Habana : s.n., 2007.
- Febles, A. 2001.** “CASE Corporativo para el proceso de control de cambios”. Tesis presentada en opción al título de Master en Informática Aplicada. Instituto Superior Politécnico “José Antonio Echeverría”. : s.n., 2001.
- . **2003.** Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. 2003.
- Feigenbaum, A. 1991.** *Control Total de la calidad*. 1991.
- . **1951.** *Total Quality Control*. 1951.
- Fortuna, R. 1990.** *El imperativo de la calidad*. 1990.
- Grimán, A., Pérez, M. y Mendoza, L. 2008.** *Estrategia de Pruebas para Software Orientado a Objetos*. Caracas, Venezuela : s.n., 2008.
- Gutierrez, M. 1989.** *Administrar para la calidad. Conceptos administrativos del control de la calidad*. 1989.
- HUMPHREY, W. 1997.** “*Managing technical people: innovation, teamwork, and the software process*”, Addison Wesley Longman. 1997.
- IEEE. 1991.** 1991.

—. **1990**. “IEEE Standard Glossary of Software Engineering Terminology,” . [En línea] Septiembre de 1990. [Citado el: 27 de Febrero de 2009.] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp.

—. **1988**. ANSI/IEEE Std 1028-1988 IEEE standard for software reviews and audits. 1988.

ISO. 2002. “ISO 19011:2002 Auditorias de gestión de calidad y/o ambiental,”. [En línea] Octubre de 2002. [Citado el: 25 de Febrero de 2009.] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?.

—. International Organization for Standardization.

ISO/IEC 9126-1. NC.

Jacobson, I. 2000. “*El Proceso Unificado de Desarrollo de Software*”, Addison Wesley Longman Inc. 2000.

Jacobson, I., Booch, G. y Rumbaugh, J. 2009. El Proceso Unificado de Desarrollo de Software. [En línea] 2009. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

Kruchten, P. 2000. The Rational Unified Process as Introduction. 2 nd Edition. Addison – Wesley. 2000.

kynetia, K. 2007. kynetia. [En línea] 2007. [Citado el: 17 de Marzo de 2009.] <http://www.kynetia.es/especializacion/calidad.html>.

Main. 2009. Main. [En línea] 2009. [Citado el: 30 de 01 de 2009.] <http://www.mitecnologico.com/Main/DefinicionCalidadDeSoftware>.

Myers, G. 2005. *Fase de Elaboración. FT Prueba (Procedimientos genéricos y aplicación de algunos tipos de pruebas simples)*. 2005.

Napal, I. 2007. “Las métricas de software en la actualidad. Su necesidad de aplicación.”, III Conferencia Científica de la Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2007.

Ortega, M. y Pérez, M. 2003. Construction of a Systemic Quality Model for Evaluating a Software Product. Software Quality Journal, 11, 219-242. 2003.

Plan de Pruebas. 2008. [En línea] 2008. http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/prod_des/documento/plan_pruebas_d.php.

Pressman, R. 2002. Ingeniería del Software: Un enfoque Práctico. McGraw Hill. 2002.

Pressman, R.S. 1998. *Ingeniería del software. Un enfoque práctico*. s.l. : McGrawHill, 1998.

Presuman, R. 2005. *Ingeniería del Software. Un enfoque práctico, Félix Varela*. 2005.

Qualitrain, B. 2008. Blog de Qualitrain. [En línea] 2008. [Citado el: 16 de Marzo de 2009.]
<http://www.qualitrain.com.mx/index.php/Procesos-de-software/Aseguramiento-de-la-calidad-de-software/Page-3.html>.

Rodríguez, M. 2007. Propuesta de Metodologías ágiles. 2007.

Rubalcaba, M. y Zambrana, Y. 2008. Medición de la calidad de Software durante el Proceso de Pruebas en el Proyecto Modernización del CICPC. 2008.

Sánchez, G. y Medel, Y. 2008. Propuesta de un procedimiento para el Aseguramiento de la Calidad en los Proyectos Productivos de la Facultad 3. Ciudad de La Habana : s.n., 2008.

Scalone, F. 2006. Software Quality Management . [En línea] 2 de Octubre de 2006. [Citado el: 16 de Marzo de 2009.]
<http://softqm.blogspot.com/2006/10/auditora-de-la-calidad-del-software.html>.

Seoane, G. y Cano, R. 2009. Desarrollo de software en India, Irlanda e Israel. Tres casos de éxito. [En línea] 2009.
<http://janiel85.spaces.live.com/blog/cns!7CC3B1BDE3E6FAD!2475.entry>.

BIBLIOGRAFÍA

1. ISO/IEC Standard 9126, Software Product Evaluation – Quality Characteristics and.
2. **Jimenez, Sasha.** *Documento Especificaciones Suplementarias.* 2006.
3. **Pressman, Roger.** *Ingeniería del software. Un enfoque práctico.* . s.l. : McGraw Hill, 1998.
4. ISO/IEC 14598-1: 1999 Information technology - Software product evaluation -Part 1: General overview. . 1999.
5. **Angeleri, M.P.M.** Programa Calidad de software del proyecto BID-FOMIN ATN-ME-8532-RG.
6. ISO/IEC 14598-5 International Standard. *Information technology - Software product evaluation - Part 5: Process for evaluators.* 1998.
7. **González., R.A.H.L.y.S.C.** El paradigma cuantitativo de la investigación científica. Ciudad de la Habana : Editorial Universitaria, 2002.
8. ISO/IEC Standard 9126. *Software Product Evaluation – Quality Characteristics and.* 1992.
9. **Lovelle, Juan Manuel Cueva.** Calidad del Software. *www.uniovi.es.* [En línea] Grupo GIDIS, Universidad Nacional de la Pampa, 1999.
10. **Z., Dr. Marcello Visconti.** Ingeniería de Software Avanzada.
11. **Muñoz, Dra. Coral Calero.** MODELOS DE CALIDAD. Calidad de Sistemas de Información. Universidad de Castilla-La Mancha: s.n., 2005.
12. **Society, Software Engineering Standards Committee of the IEEE Computer.** IEEE Std 1044-1993 Standard Classification for Software Anomalies. United States of America : s.n., 1993. ISBN 1-55937-383-0.
13. **Fernández-Pello, J.** Software Quality Assurance Network. [En línea] 27 de Junio de 2007. [Citado el: 15 de Enero de 2009.] <http://www.softqanetwork.com/>.
15. **Santos, Profesor Adjunto José Manuel.** Mediciones. Enfoques de las métricas.
16. **9126-3, ISO/IEC TR.** ISO/IEC TR 9126-3:2003 Software engineering -- Product quality - Part 3: Internal metrics. 2003. Switzerland: s.n., 2003.
17. Software Quality Management. [En línea] 2009. <http://softqm.blogspot.com/2006/10/auditora-de-la-calidad-del-software.html>.

GLOSARIO DE TÉRMINOS Y DEFINICIONES

Aplicación Multimedia: solución informática en los que se combinan audio, video, imágenes, textos.

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.

Auditoría: proceso de recoger, agrupar y evaluar evidencias para determinar si un Sistema de Información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos.

Calidad: la palabra calidad tiene múltiples significados. La calidad de un producto o servicio es la percepción que el cliente tiene del mismo. Es una fijación mental del consumidor que asume conformidad con un producto o servicio determinado, que solo permanece hasta el punto de necesitar nuevas especificaciones. La calidad es un conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.

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.

CMM: Por sus siglas en Ingles, en español, Modelo de Capacidad y Madurez. Es un modelo de evaluación de los procesos de una organización.

Configuración de software: conjunto de datos que determina el valor de algunas variables de un programa o sistema de software, éstas opciones generalmente son cargadas en su inicio y en algunos casos se deberá reiniciar para poder ver los cambios, ya que el programa no podrá cargarlos mientras se

esté ejecutando, si la configuración aún no ha sido definida por el usuario (personalizada), el programa o sistema cargara la configuración por defecto (predeterminada).

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

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

Estándares: es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

Estandarización: es la redacción y aprobación de normas que se establecen para garantizar el acoplamiento de elementos construidos independientemente, así como garantizar el repuesto en caso de ser necesario, garantizar la calidad de los elementos fabricados y la seguridad de funcionamiento.

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

Feedback: en una organización, el proceso de compartir observaciones, preocupaciones y sugerencias, con la intención de recabar información, a nivel individual o colectivo, para mejorar o modificar diversos aspectos del funcionamiento de una organización. La realimentación tiene que ser bidireccional de modo que la mejora continua sea posible, en el escalafón jerárquico, de arriba para abajo y de abajo para arriba.

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

HP: Hewlett-Packard es la mayor empresa de tecnologías de la información del mundo, superando en el 2006 a IBM, fábrica con sede en Palo Alto, California. Fabrica y comercializa hardware y software además de brindar servicios de asistencia relacionados con la informática. La compañía fue fundada en 1939 y se dedicaba a la fabricación de instrumentos de medida electrónica y de laboratorio. Hoy en día es la empresa líder en venta de impresoras.

IBM: International Business Machines (conocida coloquialmente como **el Gigante Azul**) es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911, pero lleva operando desde 1888.

IDE: Entorno de Desarrollo Integrado.

Ingeniería de Software: Disciplina tecnológica y administrativa orientada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo dentro de un presupuesto definido

ISVs: Fabricantes de Software Independientes.

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 ágil: Nuevo enfoque metodológico orientado a la gente y los resultados.

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.

Métricas: es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

Modelo: el resultado del proceso de generar una representación abstracta, conceptual, gráfica o visual.

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).

Niveles de Madurez: procesos de desarrollo de software en una escala de cierta cantidad de niveles en donde se tiene en cuenta aspectos muy variados de los procesos de desarrollo, como el grado de ambigüedad de las especificaciones, la verificación independiente de la fiabilidad de los programas, etc.

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

Plataforma Flash: aplicación en forma de estudio de animación que trabaja sobre "*Fotogramas*" destinado a la producción y entrega de contenido interactivo para diferentes audiencias alrededor del mundo.

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.

Prueba: Procedimientos para verificar que no existan errores.

Pruebas de aceptación: son las pruebas realizadas por el cliente para validar el software.

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.

SAP: (Sistemas, Aplicaciones y Productos), con sede en Walldorf (Alemania), es el segundo proveedor de software empresarial en el mundo, después de Oracle. Como empresa, comercializa un conjunto de

aplicaciones de software para soluciones integradas de negocios, entre ellas mySAP Business Suite, que provee soluciones escalables, es decir posibles de futura modificación, con más de 1.000 procesos de negocio, que la empresa clama se encuentran entre las mejores prácticas empresariales.

Servicios Informáticos: Un servicio IT o informático es una metodología de tercerización laboral muy común en países en vías de desarrollo. Funciona cuando una empresa provee de forma fija o por tiempo determinado recursos humanos especializados en Informática a otra empresa, por lo general más grande y con más recursos económicos.

Solución Libre: denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

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

Terabits: Un terabit es una unidad de información o de almacenamiento informático normalmente abreviada como Tbit o a veces Tb. 1 terabit = 10^{12} bits = 1.000.000.000.000 bits (que equivalen a 125 gigabytes decimales). El terabit está estrechamente relacionado con el tebibit, que es igual a 240 o 1, 099, 511, 627,776 bits. Nótese que la diferencia entre un trillón de bits y un tebibit es un 10%.

Total Quality Control: Control de la Calidad Total.

Usuario: Persona encargada de utilizar el sistema, obteniendo algún beneficio

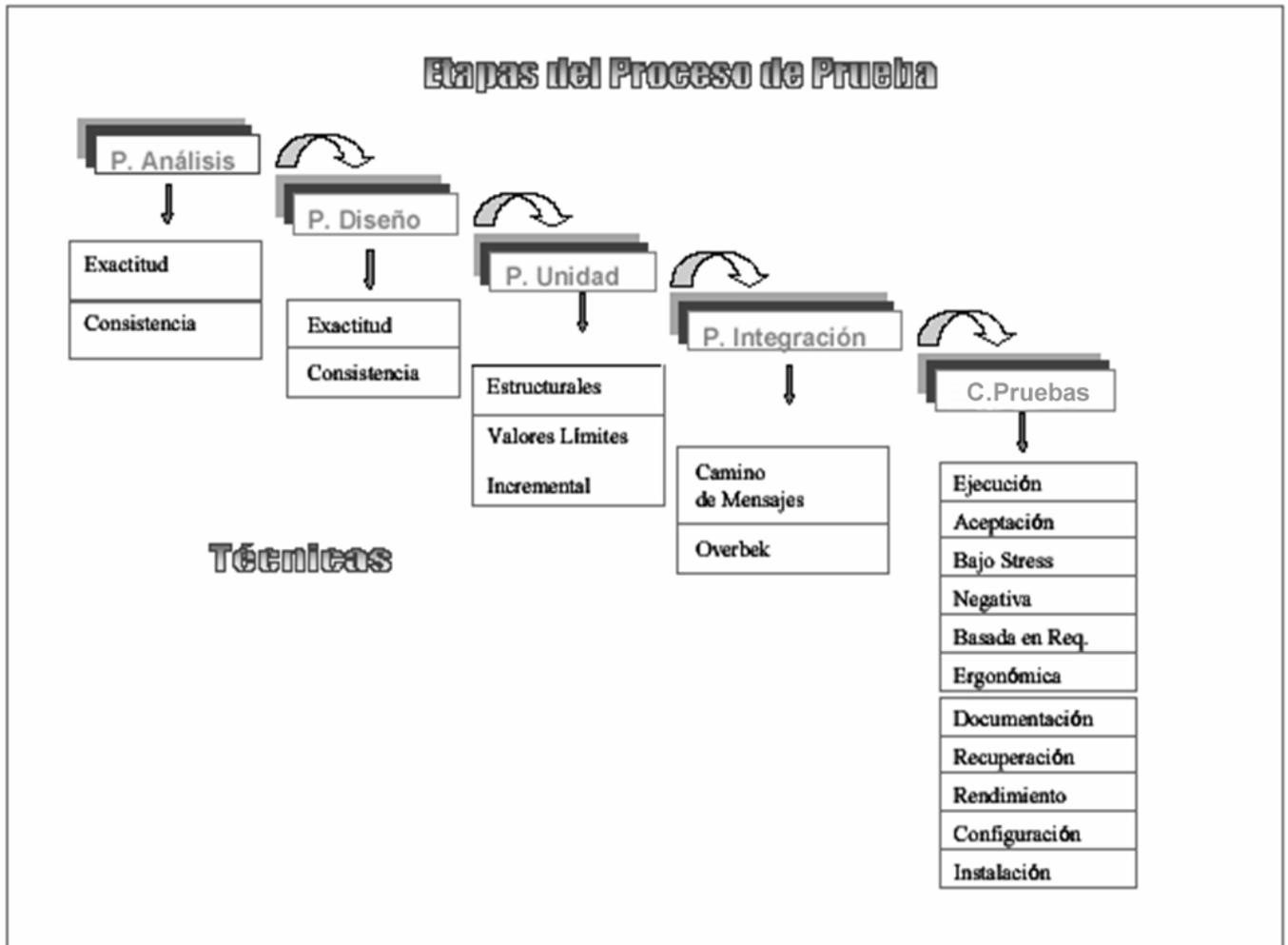
Anexo 1. Guión de la Metodología.

<p style="text-align: center;">GUIÓN DE LA METODOLOGÍA</p> <p>Planificación ↔ Definición</p> <ul style="list-style-type: none">✓ Entrevista con el cliente (concepción inicial).✓ Juego de la planificación.✓ Captura de requisitos:<ul style="list-style-type: none">➔ Creación de la LRP.➔ Definir las historias de usuario.➔ Asignar las historias de usuario.✓ Valoración del esfuerzo.✓ Valoración de riesgos.✓ Diseño con las metáforas.✓ Reunión de revisión del diseño. <p>Desarrollo</p> <ul style="list-style-type: none">✓ Junta de planificación:<ul style="list-style-type: none">➔ Definir las historias de usuario a implementar.➔ Tareas para lograr dicha implementación.✓ Implementación.✓ Junta de seguimiento.✓ Taller técnico.✓ Junta de revisión.✓ Pruebas:<ul style="list-style-type: none">➔ Unitarias.➔ Autorizadas. <p>Entrega</p> <ul style="list-style-type: none">✓ Entrega de la documentación.✓ Entrenamiento:<ul style="list-style-type: none">➔ Capacitación➔ Instalación. <p>Mantenimiento</p> <ul style="list-style-type: none">✓ Soporte. <p>Observación: Desde el diseño en la fase de Planificación ↔ Definición y toda la fase de Desarrollo es un ciclo.</p>

Anexo 2. Expediente de Proyecto.

- [-]  Picasso
 - [+]  Desarrollo
 - [-]  Expediente de proyecto
 - [-]  1- Planificación
 - [-]  1.1- Concepcion inicial
 -  01-Concepcion del sistema.odt
 - [-]  1.2- Requisitos
 -  02-Modelo de Historia de Usuario del negocio.odt
 -  03-Lista de reserva del producto (LRP).odt
 -  04-Historia de usuario.odt
 - [-]  1.3- Planificación
 -  05- Lista de riesgos.odt
 - [-]  1.4- Disenno
 -  06- Modelo de disenno.odt
 - [-]  2- Desarrollo
 - [-]  2.1- Planificación
 -  07- Glosario de términos.odt
 -  08-Tareas de ingeniería.odt
 -  09- Cronograma de producción.odt
 -  10- Plan de releases.odt
 - [-]  2.2- Implementacion
 -  11- Código fuente.odt
 -  12- Estándar de código.odt
 -  Informe de No Conformidades.odt
 - [-]  2.3- Prueba
 -  13- Plan de prueba.odt
 -  14- Caso de prueba de aceptación.odt
 - [-]  3- Entrega
 - [-]  3.1- Entrenamiento
 -  15- Manual de usuario.odt
 -  16- Manual de identidad.odt
 -  17- Manual de desarrollo.odt
 - [-]  4- Mantenimiento
 - [-]  4.1- Soporte
 -  18- Gestión de cambio.odt
 -  Plan de Aseguramiento de la Calidad.odt
 - [-]  5- Legal
 -  Acta de Aceptación.odt
 -  Acta de entrega.odt
 -  Acta de inicio de proyecto.odt
 -  Acta de terminación del proyecto.odt
 -  Carta.odt
 -  Indefiniciones.odt
 -  Informe Técnico.odt
 -  Minuta de reuniones.odt
 -  Proyectos técnicos.odt
 - [+]  Capacitación de Calidad
 - [-]  Gestión de Proyecto
 -  Guía de Auditorías.odt
 -  Informe Final de Auditoría.odt
 -  Plan de Auditorías.odt
 -  Estado_del_Arte_Picassus.odt
 - [+]  Información de la Tecnología

Anexo 3. Técnicas asociadas a las Etapas del Proceso de Pruebas.

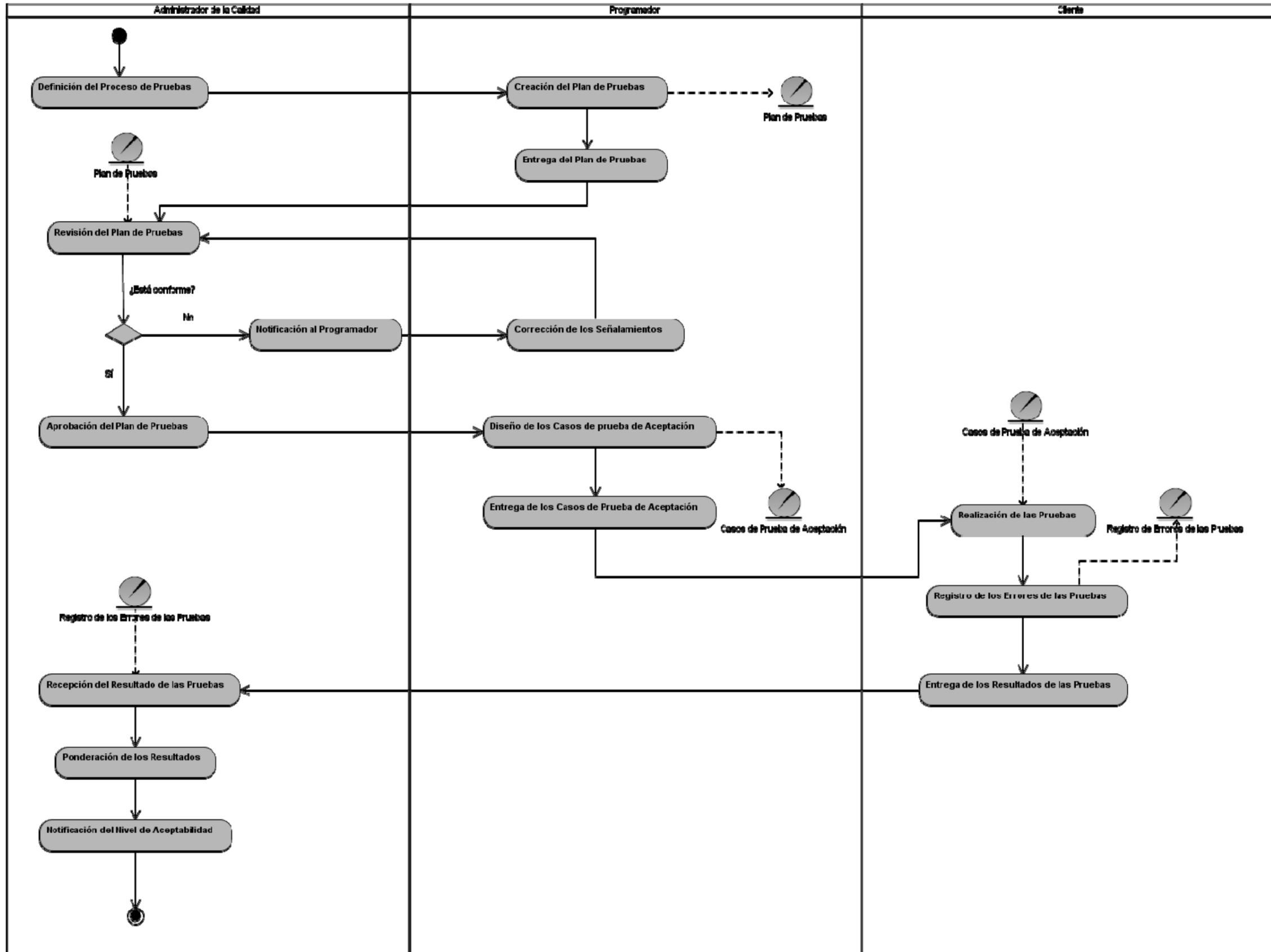


Anexo 4. Diseño Final de la Estrategia de Prueba.

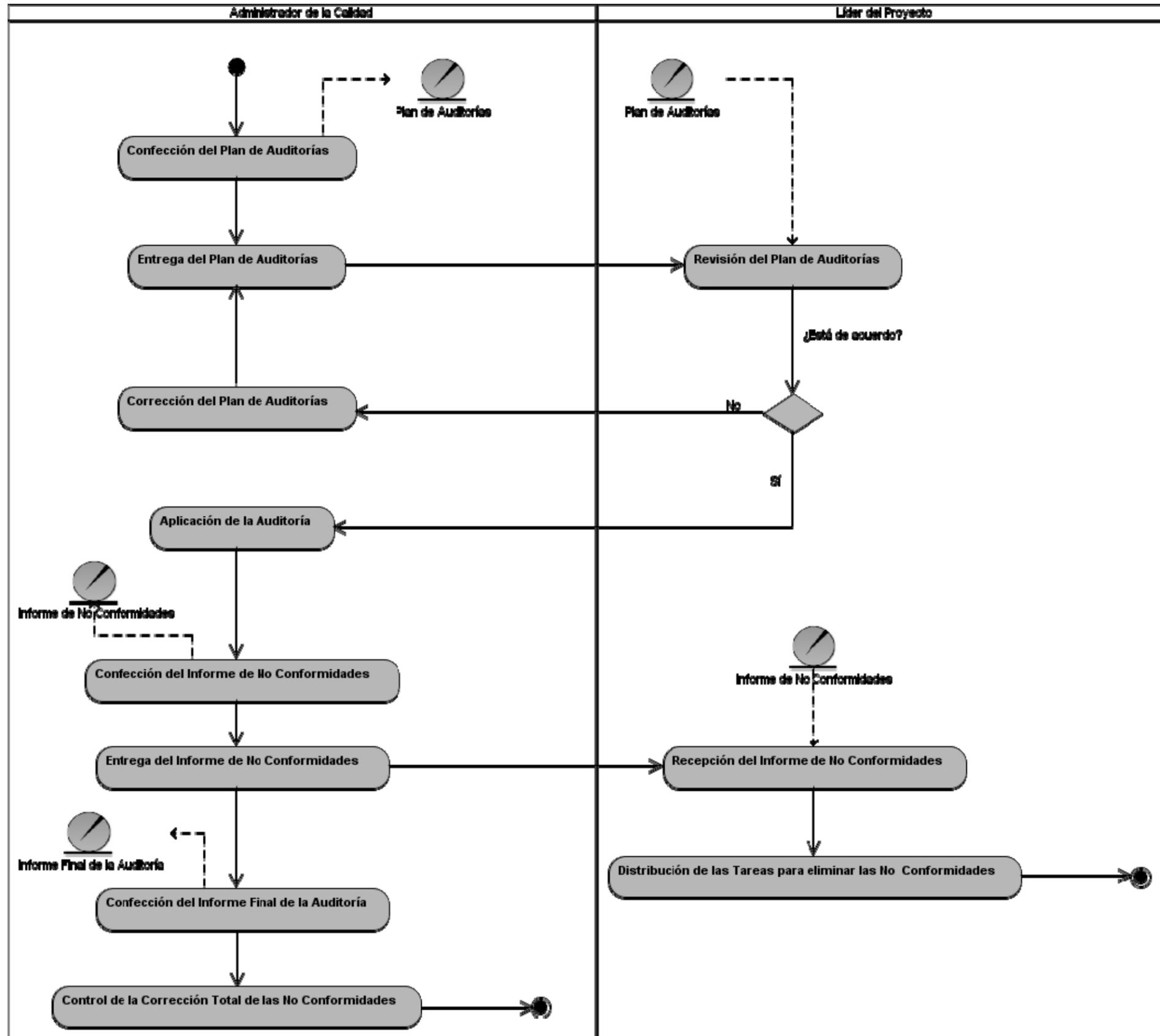
Actividad	Descripción	Participantes	Producto Obtenido	Documento Entregable
1) Solicitud de Prueba	Se debe procesar la planilla de solicitud de prueba para dar comienzo al análisis de la prueba solicitada	- Programador Principal. - Administrador de la Calidad.	- Datos del Solicitante - Tiempo estimado. - Información del equipo - Definición de riesgos - Plan de prueba	Plan de Pruebas
2) Recepción de los requisitos de la prueba	El evaluador debe verificar toda la información suministrada por el solicitante sobre el módulo o sistema a probar. Esta información servirá de apoyo a la ejecución de las pruebas.	- Líder del Proyecto - Encargado de pruebas	- Requerimientos del sistema (funcionales, no funcionales, HW, SW) - Manuales - Documentos de Análisis y Diseño	No aplica
3) Presentación del Sistema o Módulo	El evaluador recibe una presentación del módulo o sistema. Debe revisar con el solicitante cómo opera el sistema, validaciones, cálculos, etc.	- Encargado de pruebas.	- Certificación del ambiente de prueba. - Certificación de la operabilidad del sistema o modulo - Certificación de los cálculos y procesos	No aplica
4) Identificar los recursos necesarios para ejecutar las pruebas	Se identifican, en base a los recursos utilizados por el sistema, los recursos necesarios para ejecutar las pruebas, considerando de antemano el programa EPS.	- Encargado de pruebas - Líder de Proyecto	No aplica	Plan de Pruebas
5) Identificar riesgos y limitaciones	En esta actividad se deben detallar los riesgos y limitaciones que se pueden presentar durante la ejecución de las pruebas.	- Líder de Proyecto	Riesgo y Limitaciones Certificadas	Plan de Pruebas
6) Definir el alcance de la prueba	Una vez analizados los requerimientos se debe definir cuál será el alcance real de la prueba, esto depende de los requerimientos iniciales, tiempo y limitaciones	- Programador principal.	Propuesta del alcance de la prueba	Plan de Pruebas
7) Definir el Plan de Pruebas	En esta actividad se definen y planifican todas las actividades que se deben realizar, estimando los tiempos y prioridad de la ejecución de las mismas, a través de un diagrama de Gantt.	- Programador principal. - Administrador de la Calidad	Plan detallado de la prueba	Diagrama de Gantt del Cronograma de actividades

Actividad	Descripción	Participantes	Producto Obtenido	Documento Entregable
8) Ponderación de las características de calidad	Se realiza una ponderación a nivel de subcaracterísticas de calidad, para considerar la importancia dada por el solicitante a las mismas, y así establecer prioridad de una sobre otras al momento de evaluarlas	Administrador de la Calidad	Ponderación de las subcaracterísticas	Planilla de ponderación
9) Establecimiento del ambiente de prueba	Se establece el ambiente de prueba para que el evaluador pueda dar inicio al proceso.	- Líder de Proyecto	No aplica	No aplica
10) Ejecución de las pruebas	Las pruebas se realizan por etapas a través de la aplicación de Listas de Chequeo. Para dar respuesta a algunas preguntas se deben procesar y registrar casos de prueba.	- Encargado de pruebas.	No aplica	- Planillas de Casos de Prueba -Listado de fallas encontradas -Reporte de resultados del EPSOO. -Análisis de resultados
11-Culminación de las pruebas	El criterio de culminación es completar las Listas de Chequeo para cada etapa del proceso de prueba, tomando en cuenta el tiempo previsto para las pruebas.	-Administrador de la Calidad. -Encargado de pruebas.	No aplica	No aplica

Anexo 5. Diagrama de Actividades del Proceso de Pruebas de los RNF.



Anexo 6. Diagrama de Actividades para el Proceso de Auditorías.



Anexo 7. Lineamientos de Calidad UCI.

INFRAESTRUCTURA PRODUCTIVA
IPL-3100:2008
Lineamientos de Calidad de
Software
Sección: 02 – Infraestructura Productiva
Capítulo: 02.18
Instrucción 02.18.00

Reglas de Confidencialidad

Clasificación: LIMITADO

Forma de distribución: PDF Digital

Este documento contiene información propietaria de la **Universidad de las Ciencias Informáticas** y fue elaborado confidencialmente para un propósito específico. El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimiento público su contenido, excepto para cumplir el propósito por el cual se ha generado. Estas reglas son aplicables a las 15 páginas de este documento.

Resumen

El aseguramiento de la calidad de las producciones software es una de las prioridades de la Universidad. Para ello se hace necesario contar con Lineamientos, procedimientos y estándares que normen las producciones. En este contexto se suscriben los Lineamientos de Calidad que constituyen una guía a seguir por los proyectos de desarrollo de software atendiendo a cuáles son los artefactos que deben generar y procesos que deben realizar para aspirar a un producto final con calidad.

La Vicerrectoría Primera, la Dirección General de la IP, la Dirección de Calidad de Software de la IP, así como el jefe y especialistas que integran el Grupo de aseguramiento de la calidad de la IP, son los máximos responsables del cumplimiento y aplicación de estos lineamientos.

1 Nombre del procedimiento

Lineamientos de Calidad de Software (IPP-3100:2008).

2 Objetivo

Establecer los lineamientos a seguir para la producción de software en la UCI.

3 Alcance

Proyectos Productivos de la UCI.

4 Referencia

Manual de procedimientos IPP-1000:2008

5 Responsable

Ejecuta: Proyectos de desarrollo de software de la UCI

Responsable de su ejecución: Directores Infraestructura Productiva

Revisa y actualiza este procedimiento: Director de Calidad de Software, Jefe y Especialistas del Grupo de aseguramiento de la calidad.

Fiscaliza su cumplimiento: Grupo de Trabajo para la implementación de la Resolución 297/2003 del Ministerio de Economía y Planificación, Grupo de Estrategia, Grupo de Toma de Decisiones y Director General de la Infraestructura Productiva

6 Términos y Definiciones

IP: Infraestructura Productiva.

UCI: Universidad de las Ciencias Informáticas.

EP: Expediente de proyecto.

(P): Existe plantilla en el Expediente de proyecto definida para el artefacto.

(A): Artefacto que evidencia el cumplimiento del lineamiento.

(G): Guía de referencia para la ejecución del lineamiento.

7 Normas generales

- El cumplimiento de los presentes lineamientos será chequeado en las auditorías y revisiones que se le practiquen al proyecto.
- El cumplimiento de estos lineamientos es obligatorio para todos los proyectos de desarrollo de software de la UCI.
- En caso de que algunos de los lineamientos expuestos en el presente documento no sea aplicable a algún proyecto debido a la naturaleza del mismo, deberá constar en la documentación los motivos de la decisión firmada por el Director General de Producción.

- Para la construcción de los artefactos del proyecto es obligatorio hacer uso del Expediente de proyecto con las siguientes consideraciones.
- Si una plantilla para un determinado artefacto no se encuentra presente, el proyecto envía una propuesta al asesor de calidad de su facultad que es el encargado de aprobarla.
- El formato de las plantillas no es estricto, estas pueden ser adaptadas al ámbito del proyecto siempre y cuando no se viole el contenido.

8 Lineamientos

8.1 Generales

8.1.1 Planificar capacitación para el personal del proyecto.

Definir estrategia de capacitación para el personal del proyecto que incluyan cursos de postgrado, cursos optativos y cualquier otra modalidad definida por el proyecto, así como fomentar las maestrías, doctorados, tesis de pregrado, publicación de artículos y participación en eventos asociados todos a un proyecto de Investigación + Desarrollo aprobado por el Consejo Científico de la Universidad.

(A)(P) Plan de capacitación.

(G) Revisar documento del proceso “Entrenamiento Organizacional” (OT de CMMI).

8.1.2 Establecer registro de resultados de investigación del proyecto

Registrar los resultados de las investigaciones realizadas en el marco del proyecto. Estos resultados deben quedar almacenados en el expediente de proyecto.

8.1.3 Definir roles y responsabilidades

Definir los roles presentes en el proyecto y las responsabilidades del mismo.

(A)(P) Roles y responsabilidades.

(G) Revisar documento “Roles del Equipo de Proyecto”.

8.1.4 Definir las competencias

Definir las competencias necesarias para el desempeño de cada uno de los roles en el proyecto.

(A)(P) Roles y responsabilidades.

8.1.5 Definir equipos de proyecto

Definir equipos de trabajo, documentar su composición incluida la dirección del proyecto y asignar roles a los integrantes.

(A)(P) Roles y responsabilidades.

(G) Revisar documento “Roles del Equipo de Proyecto”.

8.1.6 Cuidar los bienes del cliente.

Identificar, verificar, proteger y salvaguardar los bienes que son propiedad del cliente suministrados para su utilización o incorporación al producto.

(A)(P) Plan de Desarrollo de Software.

8.2 Ingeniería

8.2.1 Gestionar requisitos

Gestionar los requisitos de los productos del proyecto y componentes de producto e identificar inconsistencias entre los requisitos, el plan de proyecto, visión del proyecto y productos de trabajo. Planificar la gestión de requisitos de software.

(A)(P) Plan de gestión de requisitos.

(G) Revisar documento del proceso “Gestión de Requisitos” (REQM de CMMI).

8.2.2 Desarrollar requisitos

Obtener, analizar, describir y validar los requisitos del producto y sus componentes.

(A)(P) Especificación de Requisitos de Software

(A)(P) Modelo del sistema

(G) Revisar documento del proceso “Desarrollo de Requisitos” (RD de CMMI).

8.2.3 Definir arquitectura de software

Definir la arquitectura de software a utilizar en la construcción de la solución.

Esta arquitectura debe encontrarse aprobada por la Dirección Técnica de Producción.

(A)(P) Documento de arquitectura de software.

8.2.4 Definir arquitectura de información

Tener definida la arquitectura de información.

(A)(P)Arquitectura de información

(A)(P)Informe del levantamiento de información para la arquitectura. (Documento).

8.2.5 Definir modelo de diseño

Definir clases de diseño y diagrama de clases de diseño.

(A)(P)Modelo de diseño.

8.2.6 Definir estándares para el desarrollo del proyecto

Especificar estándares para:

- Nombre de documentos
- Codificación
- Base de datos

(A)(P) Plan de Aseguramiento de la Calidad.

8.2.7 Planificar y ejecutar pruebas del proyecto

Planificar pruebas internas y de liberación donde se verifiquen los requisitos funcionales y no funcionales del proyecto. Cada prueba debe estar documentada en su correspondiente plan de pruebas.

(A)(P) Diseño de Casos de Prueba.

(A)(P)Plan de Pruebas.

8.2.8 Efectuar seguimiento de las No Conformidades

Efectuar seguimiento a las No Conformidades detectadas. Este seguimiento implica mantener registro de las No Conformidades detectadas y el estado de solución de las mismas.

(A)(P) No Conformidades.

8.2.9 Diseñar artefactos de prueba

Diseñar casos de pruebas y listas de chequeo para la ejecución de las pruebas internas del proyecto.

(A)(P)Diseño de casos de prueba.

(A)(P)Listas de chequeo.

8.3 Gestión de proyectos

8.3.1 Definir proyecto técnico para el proyecto

Definir proyecto técnico para el proyecto donde se identifique el marco de referencia del proyecto, la organización del proyecto y se describan las características generales del mismo.

(A)(P)Proyecto Técnico.

8.3.2 Estimar costo y esfuerzo del proyecto

Realizar estudio que refleje el costo del proyecto teniendo en cuenta los conceptos de licencias de software, compra de equipamiento, compra de bibliografía y otros que estime el proyecto. También debe aparecer reflejado el esfuerzo que conlleva la ejecución del proyecto reflejado en la cantidad de personas necesarias, el número de horas/hombres y el tiempo previsto para el desarrollo. Especificar método de estimación utilizado.

(A)(P) Plan de Desarrollo de Software.

8.3.3 Desarrollar plan del proyecto

Definir cronograma para el proyecto, los planes para la ejecución de los procesos de desarrollo y cómo chequear este plan estableciendo puntos de chequeo. Este plan debe estar en concordancia con los costos y esfuerzos estimados.

(A)(P)Plan de Desarrollo de Software.

(A)(P)Presupuesto.

8.3.4 Definir plan de resultados del proyecto

Definir plan de resultados a obtener por el proyecto especificado con fechas y artefactos.

(A)(P)Plan de Desarrollo de Software.

8.3.5 Establecer horarios de trabajo de los miembros del proyecto

Establecer horario de trabajo para todos los miembros del equipo de proyecto.

8.3.6 Identificar riesgos del proyecto y establecer plan de mitigación de los mismos.

Identificar los riesgos que presenta la ejecución del proyecto así como un plan para la mitigación de los mismos.

(A)(P) Plan Mitigación de Riesgos.

(G) Revisar documento del proceso “Gestión de Riesgos” (RSKM CMMI v1.2).

8.3.7 Gestión de recursos

Gestionar los recursos materiales del proyecto.

- Laptops
- Computadoras
- Sillas
- Periféricos
- Otros de importancia para el proyecto.

Definiendo las etapas del proyecto en la que son necesarios.

Gestionar recursos humanos del proyecto, definiendo las etapas del proyecto en la que son necesarios.

8.3.8 Registrar los acuerdos de trabajo y las minutas de las reuniones

Registrar los aspectos tratados y acuerdos tomados con responsables y fecha de cumplimiento de las reuniones del proyecto.

8.4 Soporte

8.4.1 Definir Plan de Aseguramiento de la Calidad

Contar con un Plan de Aseguramiento de la Calidad donde queden definidos los estándares y procedimientos a utilizar en el proyecto.

(A)(P) Plan de Aseguramiento de la Calidad.

(G) Revisar documento del proceso “Aseguramiento de la calidad de procesos y productos” (PPQA de CMMI).

8.4.2 Definir Plan de Gestión de la Configuración de Software

Definir Plan de Gestión de Configuración donde se especifique el esquema de identificación de los Elementos de Configuración de Software (ECS), se define la estrategia a seguir para la creación de la línea base del proyecto, se especifiquen las métricas asociadas a la configuración de software y las auditorías a la configuración.

(A)(P)Plan de Gestión de Configuración de Software.

8.4.3 Definir glosario de términos

Definir glosario donde se reflejen los términos utilizados en el negocio y las abreviaturas. Así como los conceptos que se estimen pertinentes para lograr una mejor comprensión del proyecto.

(A)(P)Glosario de términos.

8.4.4 Definir configuración de la metodología a utilizar

Documentar la configuración de la metodología a utilizar: fases, actividades, roles y artefactos.

8.4.5 Utilizar herramienta CASE

Declarar herramientas CASE utilizadas en el proyecto para las actividades de:

- Modelado
- Implementación
- Control de versiones
- Gestión de Proyecto
- Gestión documental

(A)(P)Documento de arquitectura de software.

Estas deben estar aprobadas por la Dirección Técnica de la IP.

Anexo 8. Plan de Auditorías.

Tareas generales de Revisiones y Auditorías

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

- Inspecciones del Colectivo: Estas inspecciones se realizarán semanalmente siempre en dependencia de la duración de la iteración.
- Revisiones dinámicas: Estas revisiones se realizarán al finalizar cada iteración.

Grupo de revisiones y auditorías sugeridas a usar como base para la planificación son:

- Revisión de los Requerimientos (se corresponde con la tradicional Revisión de las Especificaciones del Software)
- Revisión del Diseño (se corresponde con la tradicional Revisión del Diseño Crítico)
- Auditoría de la configuración funcional (para verificar que todos los requerimientos han sido cumplidos)
- Auditoría de la configuración física (para verificar que el software y su documentación están completos y listos para entregar)
- Auditoría del Proceso
- Revisión del Proceso
- Revisión Administrativa (Revisión de Aprobación del Proyecto, Revisión de la Planificación del Proyecto, Revisión del Plan de Iteración)
- Revisiones Post-mortem (Revisión de Aceptación de la Iteración, Revisión de Aceptación del Proyecto).

Se realizará una auditoría en la última semana de cada iteración planificada por el proyecto, ejecutada por el Administrador de la Calidad. Además de que los artefactos auditados estarán en correspondencia con los Lineamientos UCI definidos para la fase de desarrollo en la que se encuentre el proyecto.

Cronograma de Auditorías

Módulo	Fecha de Inicio	Fecha de Fin
Picassus	2/04/2009	9/04/2009
Picassus	4/05/2009	11/05/2009
Picassus	9/06/2009	15/06/2009

Organización y Responsabilidades

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

Resolución de problemas y actividades de corrección

Las actividades del proceso varían dependiendo del tipo de revisión que se efectúe, si se trata de una inspección se comenzará por designar un miembro del equipo de desarrollo, encargado de conducir todo el proceso hasta obtener el reporte final de la inspección y entregar al administrador del aseguramiento de calidad que se encargará de emitir las peticiones de cambio correspondientes, si fuese necesaria una reinspección. En el caso de una revisión dinámica las actividades son las siguientes:

1. Designar el equipo de prueba o evaluación del software
2. Confeccionar casos de prueba y procedimientos de prueba o en su defecto las listas de comprobación correspondientes.
3. Detectar y registrar los defectos a partir de la ejecución del software.
4. Notificar al Administrador del aseguramiento de la calidad y Líder del proyecto.

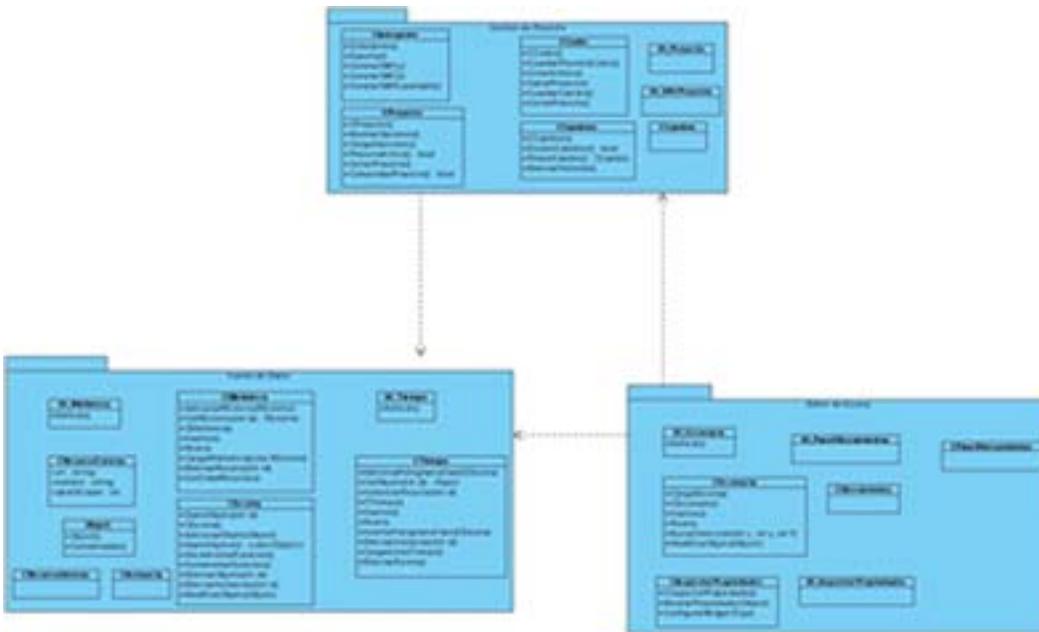
Anexo 9. Documento de No Conformidades.

CCV
Informe de No Conformidades
Versión 1.0

No.	Elemento Evaluado	Aspecto Correspondiente	Descripción de la no conformidad	Tipo	Responsable de Solución	Versión
1	Concepción del Sistema	Contenido	1-Las oportunidades de negocio no se encuentran bien definidas. 2-No se detallan las áreas donde el producto debe incidir. 3-No se realiza un estudio de mercado. 4-No se evalúan las necesidades de los involucrados.	Importante	Roberto Ferrer Michel Miranda	1.0
2	Historia de Usuario	Contenido	1-No se define el alcance del proyecto.	Importante	Jonny P. Martin	1.0
3	Lista de Riesgos	Contenido	1- No se define ni alcance ni los objetivos. 2-No se estable un plan de contingencia.	Importante	Nidia Fernández	1.0
4	Modelo de Diseño	Contenido	1-No se define el alcance. 2-Diagrama de clases de poca comprensión. 3-No se realiza la descripción de cada una de las clases.	Importante	Nidia Fernández	1.0

5	Plan de Capacitación	Contenido	1-No existe este documento en el repositorio del proyecto.	Importante	Roberto Ferrer	1.0
9	Plan de Aseguramiento	Contenido	1-No se describen los registros que se mantendrán durante el proyecto. 2-No se describen las actividades de capacitación.	Importante	Nidia Fernández	1.0

Anexos



No Conformidad 4.2

Anexo 10. Informe Final de Auditoría.

Identificación del Cliente

Polo de Software Educativo de la Facultad 8.

Identificación de la Entidad auditada

Producto: Picassus **Proyecto:** Grupo de Alternativas Libres para Multimedia

Objetivos de la Auditoria Informática

La Auditoria de Calidad para el proyecto tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva del producto y los procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos establecidos.

Normativa aplicativa y excepciones

La auditoría fue realizada teniendo como norma la guía del documento establecido por la Dirección de Calidad de la UCI que recibe el nombre Bases para las Auditorías y Revisiones y los Lineamientos de Calidad UCI.

Alcance de la Auditoria

El alcance de la auditoria es realizar una revisión exhaustiva de toda la documentación de GALM para detectar todos los errores cometidos y poder establecer acciones correctivas.

Conclusiones

La aplicación de esta auditoría propició conocer el estado con que avanza el proyecto en cuanto a la documentación y si el trabajo es eficiente y si se ajusta plenamente a lo establecido por la metodología utilizada.

Se encontraron numerosos defectos en cuanto al llenado de los documentos pues muchos de ellos estaban incompletos o no daban la información necesaria para cualquier persona que los leyera. Debido

esto se realizó un Informe de No Conformidades para que el Líder del proyecto junto al equipo de desarrollo pudiera establecer un plan de acciones correctivas.

Como acciones correctivas que pudieran ser consideradas por el equipo de trabajo para el mejoramiento del proceso de desarrollo del software se encuentran las siguientes:

- Revisión de los procesos (políticas y procedimientos).
- Capacitación al equipo de desarrollo, y en general a todos los integrantes del proyecto mediante dos cursos para familiarizar al desarrollador con las herramientas y el sistema operativo que utiliza el proyecto.
- Establecer un documento de Plan de Capacitación donde queden bien establecidos los cursos que imparten en el proyecto y el propósito de los mismos.
- Reforzar el tema del control a la asignación de responsabilidades dentro del equipo de trabajo para garantizar que no quede nada por hacer y que cada cual tenga su responsabilidad claramente definida.
- Tomar medidas con los integrantes del equipo que no cumplieran con lo establecido por sus superiores, para evitar que estos tomaran decisiones por su cuenta.

Con el desarrollo de esta auditoría se aseguro la construcción hasta el presente de un producto con calidad que satisfaga las necesidades y expectativas del cliente.

Fecha del Informe

22 de Abril del 2009

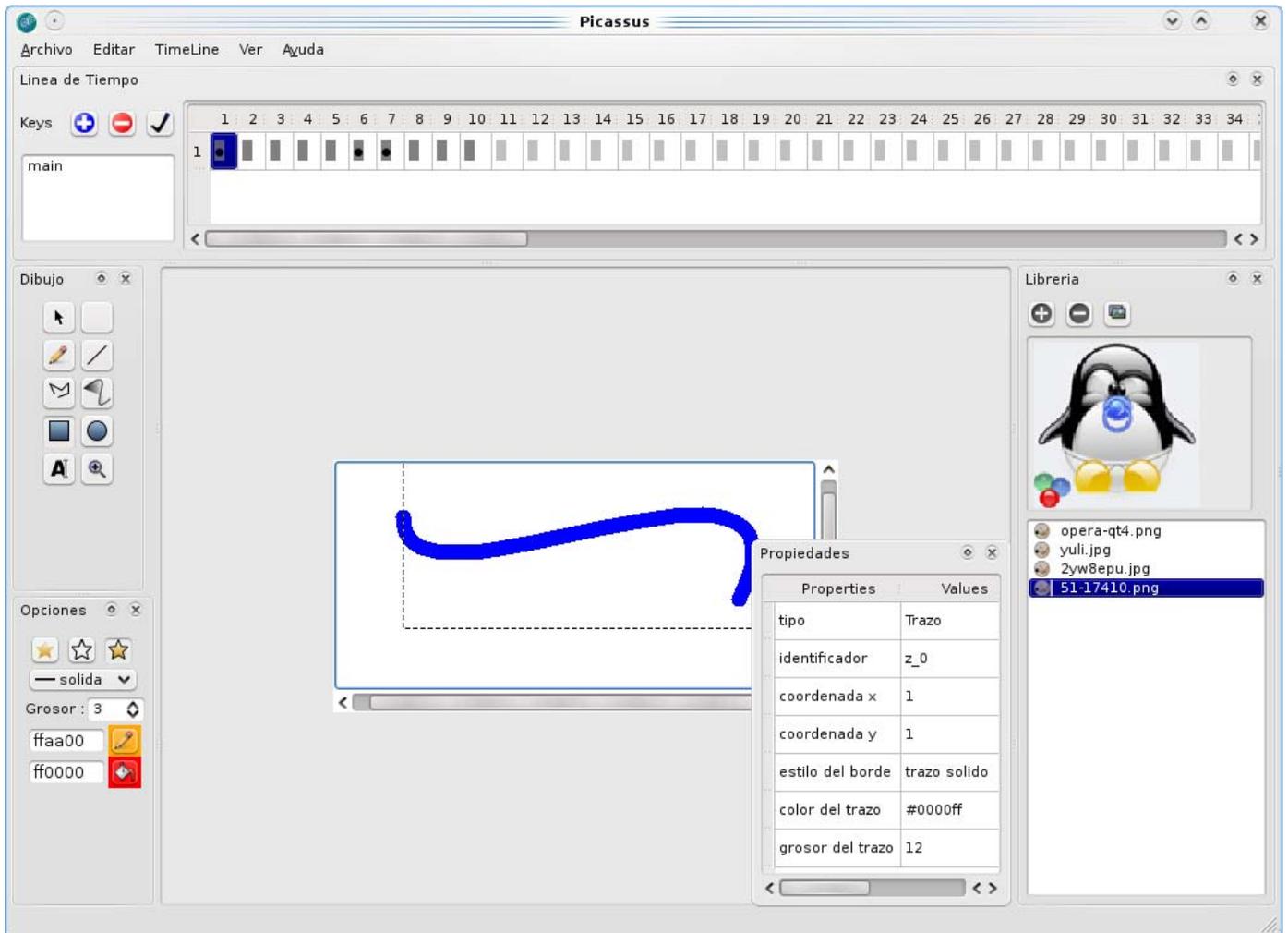
Auditor

Nidia Fernández Pérez

Líder del Proyecto

Roberto Ferrer Obregón

Anexo 11. Prototipo de Interfaces para la Historia de Usuario Exportar Biblioteca con Imágenes.



Anexo 12. Prototipo de Interfaces para la Historia de Usuario Funcionalidades de Edición de elementos.

