

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS
FACULTAD 10 SOFTWARE LIBRE**



**Título: Propuesta de una guía para asegurar la calidad del producto
en proyectos que desarrollan Aplicaciones Web.**

Autores: Zuleydis Caridad Baralt Rivera

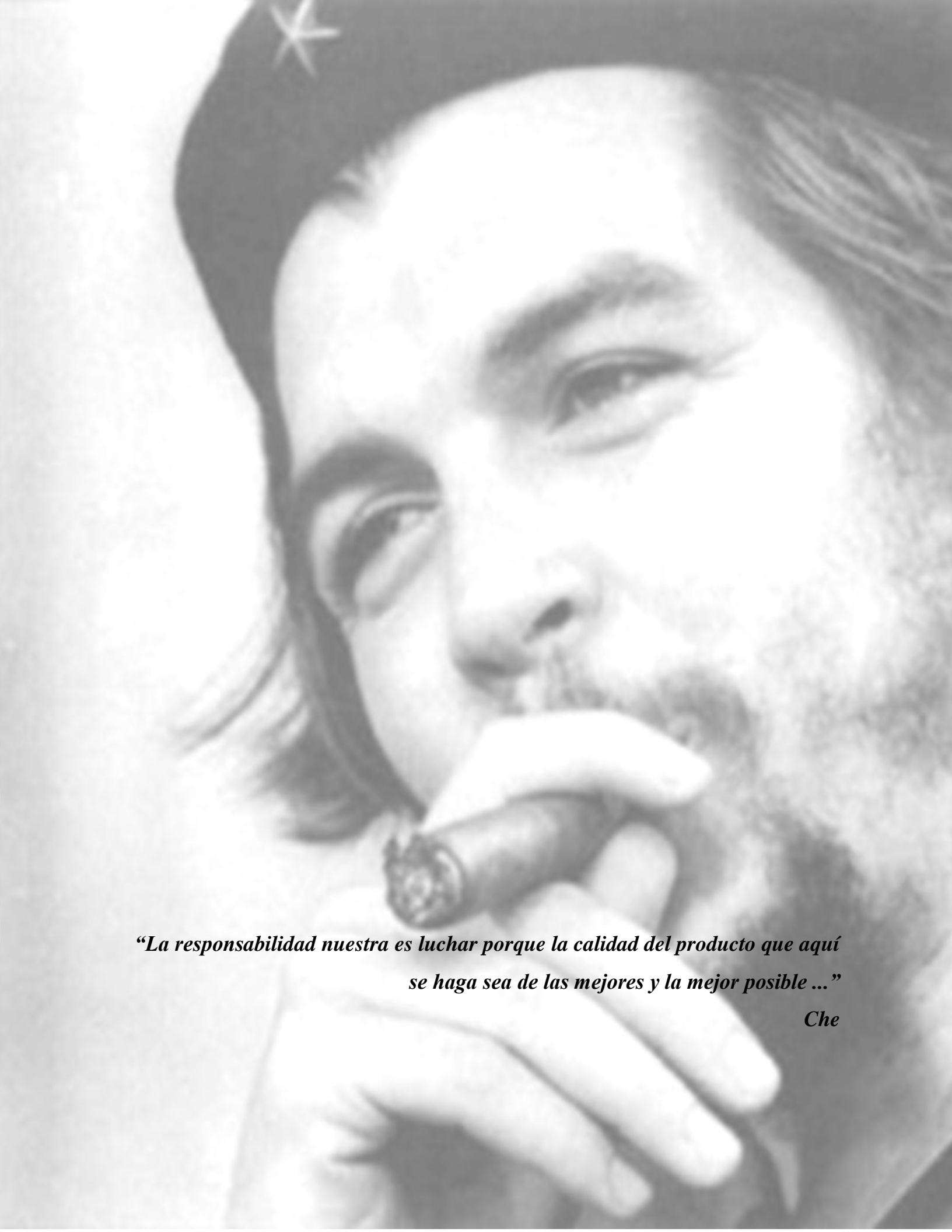
Yodelys Castro Pérez

Tutores: Ing. Yusleydi Fernández del Monte

Ing. Sonia Guerrero Lambert

Ciudad de la Habana, Junio 2009

“Año del 50 aniversario del triunfo de la Revolución”

A black and white close-up portrait of Che Guevara. He is wearing a dark beret with a light-colored star on the front. He has a beard and is looking slightly to the right of the camera. He is holding a cigar in his right hand, which is raised towards his mouth. The lighting is dramatic, highlighting his facial features.

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

Che

DECLARACIÓN DE AUTORÍA

Declaramos ser las únicas autoras de la tesis presente y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

Para hacer constancia firmamos la presente a los ----- días del mes de ----- del año 2009

Firma del Autor
Zuleydis C. Baralt Rivera

Firma del Autor
Yodelys Castro Pérez

Firma del Tutor
Ing. Yusleydi Fernández del Monte

Firma del Tutor
Ing. Sonia Guerrero Lambert

DATOS DE CONTACTO

Tutor: Yusleydi Fernández del Monte.

Email: ydelmonte@uci.cu

Curriculum:

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10

Tutor: Sonia Guerrero Lambert.

Email: sguerrero@uci.cu

Curriculum:

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10.

AGRADECIMIENTOS COMPARTIDOS.

Agradecemos a nuestros familiares en especial a nuestros padres y hermanos que han hecho posible este sueño, por ser nuestro motor impulsor y el regalo más bello que hemos recibido de la vida.

A nuestros compañeros y amigos que nos brindaron consuelo cuando el camino se tornó más difícil.

A los profesores de la Universidad de las Ciencias Informáticas, por la formación que nos han dado a lo largo de estos cinco años de la carrera.

A nuestras tutoras Zuleydis y Sonia por su asesoría constante y su preocupación por nosotras, por tanta ayuda, por tantos días, noches y madrugadas de trabajo por todo el interés, por la confianza depositada.

A esta Universidad por ser el fruto de la Batalla de Ideas, consecuencia de una coyuntura política, y resultado del pensamiento de un hombre que en las más difíciles adversidades nunca ha perdido la confianza en el futuro, por todas las enseñanzas, experiencias y recuerdos inolvidables, que quedarán para siempre guardados en nuestra memoria.

A Fidel Castro Ruz por ser un maestro excepcional, el guía impulsor de este sueño hecho realidad del cual formamos parte.

Zuleydis y Yodelys.

Zuleydis:

A la Virgen de la Caridad del Cobre, por estar ahí siempre apoyándome, en la cual he depositado mi fe y que hasta hoy no me ha defraudado nunca.

A mis padres por todo el amor, cariño y dedicación en estos 22 años, confianza, consejos y apoyo que me han brindado en los buenos y malos momentos de mi vida, por ser mis guías, por inculcarme tantos valores positivos, el amor al trabajo, el sacrificio ante las situaciones de la vida, a ser independiente, en fin por ser los padre que todos anhelan y quieren.

A mi hermana y sobrino por tanto apoyo y por la unión que reina entre nosotros.

A Alina por ser mi segunda madre y consejera.

A mi hermano Mijaíl Arístides, que aunque no se encuentre físicamente en el mundo de los vivos, se que esta siempre a mi lado apoyándome y dándome fuerzas.

A Israel por ser mi amigo en todo momento, padre, tutor y profesor. Por tantos momentos juntos, por tanto apoyo de corazón.

A Fodelys, por ser más que mi compañera de tesis, por ser amiga, hermana, a veces consejera, por estar ahí tantas pero tantas veces para escucharme en todo momento en los buenos y en los realmente malos, pero ahí siempre apoyándome.

A Ulises y Justina por tantas inyecciones de fuerza y sostén ante los problemas presentados.

A mi familia, por tanta preocupación y apoyo sentimental y fraternal.

A mis primas Fleydis y Malinabis por convertir mis vacaciones en momentos inolvidables.

A mis grandes y eternas amigas Fadhira, Xarelia y Familka por los intensos veranos y merecedores fines de año que compartimos juntas.

A Josefa por ser mi consejera desde el inicio de mi carrera, por sus respiraciones de apoyo y por cuidar de Mami en estos cinco años que estuve lejos del seno familiar.

A mis compañeros, primos y amigos complacientes por tanta dedicación y por tantos momentos de felicidad.

A todas mis amistades, pero en especial aquellas que saben que estuvieron ahí cuando las necesitaba.

A mis compañeros del aula y en especial aquellos que compartieron mis logros y decepciones.

A todos, de todo corazón, les agradezco y les pido,

Muchas Gracias.

Podelys:

*A todos mis compañeros que han estado junto a mí durante estos 5 años de carrera por haberme brindado su sincera amistad.
A mis compañeras más cercanas Deborah, Annie, Mayelin, Dania, Ailema y Fátima por haber sabido aconsejarme en
cada momento justo y haber compartido en tantas fiestas y momentos de felicidad.*

*A Baby mi más fiel amiga jobabense por haberme soportado durante mis tres años de preuniversitario, y por ser más que una
amiga una hermana.*

*A Zuleydis mi compañera de tesis, amiga, hermana y amuleto de la suerte por tantos consejos de la vida, por haberme
enseñado y ayudado en todo momento, por haber sabido entrar y dejar huellas en cada corazón de sus amistades y en el mío en
especial.*

*A Modesto que aunque no haya tenido los conocimientos suficientes sobre el tema de mi tesis siempre buscó alguna estrategia
para ayudarme y apoyarme en todo, por haber compartido tanto momentos de risotadas, fiestas y alegría como momentos de
tristeza y lágrimas, por quererme y demostrarlo tanto en cada gesto .*

*A Edel por ser una de las pocas personas que saben descubrir lo que existe en mi mirada, por tener ese concepto tan amplio de la
amistad y familia y haberlo compartido conmigo.*

*A todos mis tíos por quererme tanto en especial Lory y Eduardo por estar siempre tan cercanos y brindarme su cariño más
puro.*

A mis abuelos Adis y Amado por ser mis segundos padres y ayudar a que este sueño sea realidad.

A mi hermanito por velar siempre por mí y quererme tanto del tamaño del cielo del mundo entero y de la botellita de sangre.

*A mis padres, los mayores tesoros que tengo en la vida, por estos 22 años, por haber compartido este sueño, por ser mis inspiradores,
maestros y amigos.*

A todos muchas gracias por haber ocupado un espacio de mi vida y carrera.

DEDICATORIA

Zuleydis:

Este gran triunfo está dedicado principalmente a mi Mama, Miriam Rivera Guza, que sin ella nada de esto fuera posible, a ella que toda una vida ha sido una excelente madre, amiga, compañera, por ser lo más grande que tengo en el mundo, por dejarme ver a través de sus ojos.

A la memoria de mi hermano y abuelita que siempre estarán en mi corazón.

A mi madre La Virgen de la Paridad del Sobre con todo respeto, amor y cariño.

Podelys:

A mi papi y Berti por haberme dado la vida, a mi hermanito, abuelos Adis y Amado por su apoyo, perseverancia, confianza, amor y sabios consejos.

A los cuatro por hacer de mí lo que soy hoy y por haber hecho posible que este sueño sea una realidad.

RESUMEN

Todo proyecto en la rama de la informática generalmente, basa su desarrollo en producir software de la mejor calidad posible, que cumpla, y supere las expectativas de los usuarios.

La calidad es un logro que todos buscan convirtiéndose en un atributo primordial para el producto de software, la cual se debe garantizar en cada equipo o proyecto que se desarrolle, con el objetivo de obtener mejores resultados, dar cumplimiento de forma eficiente a los requerimientos propuestos y satisfacer las necesidades del cliente. La propuesta de la guía para asegurar la calidad del producto software en proyectos que desarrollan aplicaciones web es una posibilidad en el mejor desempeño que puede ejercer el asegurador, para la evolución y mejora de los productos. Los aseguradores dentro de los proyectos de la UCI la mayoría de las veces no realizan las actividades que requiere su rol por lo que afrontan un cúmulo de dificultades para asegurar sus productos. Esto se basa en la poca experiencia, el poco manejo y las pocas herramientas y documentación que tienen sobre el tema para realizar el trabajo. Lo que trae como consecuencia que se concluya con resultados ineficientes, que se incumpla con los requerimientos y que principalmente no se integren todas las necesidades del cliente. Atribuido a estas consecuencias se integra el hecho de que la universidad como empresa productora de software, pierda confiabilidad y prestigio, convirtiéndose a su vez, en centro de no excelencia. Este problema se alimenta debido a no poseer un procedimiento que sirva de apoyo para garantizar o asegurar la calidad de los productos. Para ello se propone elaborar una guía para asegurar la calidad del producto software en proyectos de la UCI que desarrollan aplicaciones Web. Para desarrollar la investigación se hace uso de los métodos teóricos (histórico-lógico, analítico-sintético) y empíricos (observación, entrevista, encuesta), los cuales se usaron para demostrar la necesidad de incorporar en la universidad la propuesta, como documentación y una nueva posibilidad para el aprendizaje de todos aquellos que desempeñen esta labor con el fin de garantizar la calidad del software.

Palabras Claves: Aplicaciones Web, calidad, aseguramiento, guía.

ÍNDICE

INTRODUCCION	1
CAPITULO I: Fundamento Teórico	7
1.1 Conceptos asociados al problema	7
1.1.1 Calidad de software	7
1.1.2 Calidad del producto	¡Error! Marcador no definido.
1.1.3 Aplicaciones Web.....	9
1.2 Antecedentes y actualidad de la calidad del software.....	10
1.2.1 Evolución de la calidad del software.....	10
1.2.2 Calidad de Software en el mundo	10
1.2.3 Calidad de Software en Cuba	11
1.2.4 Calidad de Software en la UCI	12
1.3 Estándares, Normas y Modelos.....	12
1.3.1 Descripciones.....	12
1.3.2 CMMI.....	13
1.3.3 Normas ISO para la Calidad de Software.....	16
1.3.4 SPICE.....	17
1.3.5 IEEE	17
1.3.6 W3C.....	18
1.3.7 Los que definen Ciclo de vida.....	18
1.4 Aseguramiento de la calidad del software.....	22
1.4.1 Descripción del aseguramiento de la Calidad.....	22
1.4.3 Aseguramiento de la calidad del producto	25
1.4.4 Métricas asociadas al aseguramiento.....	25
1.4.5 Factores de Calidad.....	26
1.4.6 Verificación y Validación	28
1.4.7 Control de la Calidad	31
1.4.8 Gestión de la configuración.....	32
1.5 Sistema de Gestión de Conocimiento.....	34
1.5.1 Descripción de la propuesta	37
1.6 Conclusiones parciales.....	38
CAPITULO II: Propuesta de Solución	39
2.1 Descripción de la propuesta para la universidad.....	39

2.2 Análisis del ciclo de vida de la propuesta.....	42
2.3 Desarrollo de las fases.....	47
2.3.1 Estructura de las Fases.....	47
2.3.2 Aspectos más importantes a desarrollar dentro de las fases.....	48
2.3.3 Fase Inicial: Planificación.....	54
2.3.4 Fase Intermedia: Desarrollo.....	58
2.3.5 Fase Final: Operación.....	62
2.4 Técnicas de uso y estudio de la Guía.....	64
2.5 Conclusiones parciales.....	66
CAPÍTULO III: Validación de la propuesta.....	67
3.1 Método de validación caso de estudio.....	67
3.2 Ventajas de utilizar la guía en un proyecto real.....	67
3.2.1 Antes del uso de la propuesta.....	70
3.2.2 Después del uso de la propuesta.....	76
3.3 Conclusiones parciales.....	83
CONCLUSIONES.....	85
RECOMENDACIONES.....	86
REFERENCIAS BIBLIOGRÁFICAS.....	87
BIBLIOGRAFÍA.....	89
ANEXOS.....	95
GLOSARIO DE TERMINOS.....	106

ÍNDICE DE FIGURAS

Figura 1: Fases de la Evolución de la Calidad del software.....	10
Figura 2: Procesos que integran a la ISO 12207.....	20
Figura 3: Procesos del ciclo de vida del modelo WQM.....	22
Figura 4: Factores de Calidad.....	28
Figura 5: Actividades de la Verificación.....	30
Figura 6: Actividades de la Validación.....	31
Figura 7: Elementos de Configuración.....	32
Figura 8: Actividades de los procesos de Gestión de Configuración.....	33
Figura 9: Metodología dML.....	35
Figura 10: Estructura de la Guía.....	40
Figura 11: Técnicas de Aseguramiento de la Calidad.....	41
Figura 12: Propuesta de Solución.....	42
Figura 13: Ciclo de Vida de Aplicaciones Web.....	43
Figura 14: Actividades asignadas a cada fase.....	44
Figura 15: Aspectos más importantes dentro de las fases.....	48
Figura 16: Diagrama de flujo de las actividades del asegurador de la calidad.....	51
Figura 17: Diagrama de flujo para evaluar el artefacto.....	52
Figura 18: Posibles resultados de una petición de cambio.....	53
Figura 19: Fase de Planificación. Actividades y Artefactos asociados.....	56
Figura 20: Fase de Desarrollo. Actividades y Artefactos asociados.....	60
Figura 21: Fase de Operación. Actividades y artefactos asociados.....	63

ÍNDICE DE TABLAS

Tabla 1: Diferencias entre VER y VAL.....	29
Tabla 2: Roles del SGC.....	36
Tabla 3: Actividades y Artefactos de las fases del ciclo de vida.....	45
Tabla 4: Roles y Actividades de las fases del ciclo de vida.....	46
Tabla 5: Evaluación de Criterios en proyectos.....	72
Tabla 6: Evaluación de Criterios en proyectos. Resultados.....	77

INTRODUCCIÓN

Como estrategia para fortalecer la industria cubana del software y como uno de los proyectos de la Batalla de Ideas, se crea la Universidad de las Ciencias Informáticas (UCI). Hoy, cuando se lleva a cabo su séptimo curso, la UCI desarrolla numerosos proyectos productivos que ya alcanzan resultados relevantes como parte de su funcionamiento, respaldado por otras instituciones del país y por el esfuerzo de toda la comunidad universitaria.

La UCI, como proyecto emblemático de la Revolución Cubana, tiene el compromiso de convertirse en un centro de excelencia, con objetivos fundamentales e inmediatos como: participar activamente en la informatización del país y contribuir significativamente al desarrollo de la industria del software como fuente importante de aporte de recursos económicos al mismo, mediante un modelo de formación desde la producción y la investigación, del que egresen profesionales altamente calificados y comprometidos.

Los procesos de la universidad están concebidos para formar capital humano especializado; investigando, produciendo software y servicios informáticos para la sociedad cubana y para el mundo en general. Estudiantes y profesores se integran crecientemente a equipos de proyectos productivos o de innovación y desarrollo. Ellos son los encargados de concebir y darle vida al software, cumpliendo con sus actividades, tareas y funcionalidades. Cada persona debe desempeñar un rol específico dentro de cada proyecto, en concordancia con la mitología, procesos y requerimientos del mismo. Estos proyectos y los resultados obtenidos por la mayoría de estas personas, contribuyen al desarrollo económico y profesional del país. Es por ello indispensable que cada rol dentro de los proyectos se conozca y desempeñen con eficiencia y organización sus tareas, alcanzando las competencias que lo garanticen.

Durante el desarrollo del software se concibe la necesidad de realizar un conjunto de actividades planificadas y sistemáticas dentro de los proyectos, que aseguren la calidad del producto, con el fin de elevar la exportación de software, los servicios informáticos y aportar la confianza que el producto requiere. Este aseguramiento debe garantizarse a lo largo del ciclo de vida del software evitando que se propaguen defectos y se afecte la eficacia de la producción así como la satisfacción del cliente. Uno de los retos de la universidad como institución es producir software de gran calidad, lo que entraña una determinada

responsabilidad para cada rol en los procesos, en lo que a garantizar la calidad de dicho software se refiere.

La evaluación de la calidad de software se ha convertido en una necesidad en la producción de los proyectos de la Universidad. En los últimos años este tema ha tomado un gran auge debido a que el producto final en muchas ocasiones ha tenido que ser cambiado y elaborado nuevamente. Uno de los motivos esenciales es, que no se tiene en cuenta el aseguramiento de la calidad; en muchos proyectos solo se realizan etapas de pruebas al culminar la implementación del producto. Esto trae como consecuencia que se detecten errores al final del desarrollo y se tengan que corregir desde la etapa inicial, ocasionando en la mayoría de los casos atrasos en la entrega del producto o deficiencia del mismo, provocando que el producto no satisfaga las necesidades que han sido pactadas con el cliente.

Hoy se confronta la dificultad de no disponer de un medio que contenga información y facilite el conocimiento de las acciones que se deben tomar para el mejor desempeño en el aseguramiento de los requisitos de la calidad del producto, principalmente dentro de los proyectos de la universidad que desarrollan aplicaciones Web. Es decir, no se cuenta con una medio que se ajuste a las necesidades, que sea conveniente y fácil de usar, lo que impide que la producción sea eficiente y en tiempo.

Esta dificultad trae consigo numerosos problemas en los proyectos de la universidad como centro universitario desarrollador de software.

Problemas tales como:

- No se cuenta con información ni bibliografía definida, para garantizar la calidad del software en el producto.
- Falta de personal capacitado para realizar las actividades de aseguramiento de la calidad.
- Insuficiente control de la calidad de los productos durante el ciclo de vida del software.
- Incumplimiento de entrega del producto en la fecha establecida.

Estas dificultades pueden traer como consecuencia:

1. Elaboración de un producto final insuficiente.

2. Incremento de la presión por mejorar el trabajo.
3. Exigencias de prórrogas para terminar el producto.
4. Mala calidad en lo que el cliente realmente necesita.
5. Resultados Insatisfactorios.
6. Perdida de fiabilidad.

De lo anteriormente expuesto se deriva el siguiente problema a resolver: ¿Cómo asegurar la calidad del producto de software en los proyectos de la UCI que desarrollan aplicaciones Web?

Para solucionar el problema existente se plantea el **objetivo de la investigación** siguiente: Elaborar una guía para asegurar la calidad del producto software en proyectos de la UCI que desarrollan aplicaciones Web.

Teniendo como **objeto de estudio**: Procesos de calidad del producto software en el desarrollo de aplicaciones Web.

El **campo de acción** en el que se enmarca la investigación son los procesos de aseguramiento de la calidad de software en el desarrollo de aplicaciones Web.

Para dar cumplimiento al objetivo trazado se determinan las siguientes **tareas de la investigación**:

- Realizar encuestas y entrevistas para definir los problemas existentes respecto a garantizar la calidad del producto en los proyectos que desarrollen aplicaciones Web.
- Investigar las tendencias y antecedentes existentes en el ámbito nacional e internacional relacionadas con la calidad de software para conocer cómo ha ido evolucionando el tema y su comportamiento actual.
- Estudiar todas las teorías, metodologías, métodos, estándares y normas existentes sobre formas de asegurar la calidad de software para obtener una base de conocimientos sólida que ayude a la elaboración de la guía.
- Investigar contenidos correspondientes al aseguramiento de la calidad y seleccionar las posibles informaciones que formaran parte de la guía.
- Elaborar una guía para asegurar la calidad de software en proyectos de la UCI que desarrollen aplicaciones Web como parte de la propuesta de solución.

Las **preguntas científicas** que sustentan la investigación son:

¿Qué recursos, herramientas, estándares, normas y modelos se podrán utilizar para asegurar la calidad del software en proyectos que desarrollen aplicación Web?

- ¿Qué aspectos utilizar para desarrollar una guía para el aseguramiento de la calidad del software en los proyectos que desarrollan aplicaciones Web, enfocada al producto?

Para el desarrollo de la investigación se utilizan los siguientes **métodos científicos**:

Métodos teóricos:

- Histórico-lógico: Permite estudiar de forma analítica la trayectoria histórica real del aseguramiento de la calidad de software en los productos así como su evolución y desarrollo, de esta forma se caracteriza el objeto en sus aspectos más externos.
- Analítico-sintético: Utilizado en el análisis de los procesos, métodos y bibliografías sobre el aseguramiento de la calidad en los productos ya existentes, permitiendo el desarrollo y la extracción de elementos que sirvan de ayuda para fomentar y elaborar la información concluyente como teoría propia.

Métodos empíricos:

- Observación: Se utiliza para realizar un registro visual de la situación real existente en proyectos de la facultad 10 con respecto al aseguramiento de la calidad en aplicaciones Web, de esta forma se detectan y clasifican los hechos y acontecimientos que imposibilitan un eficaz desarrollo del aseguramiento de la calidad en productos Web y así fundamentar la situación problemática planteada. Para ello se utiliza una guía de observación que ayuda a enfocar los aspectos más importantes del tema. (Ver Anexo 1)
- Entrevista: Este método se utiliza para la recopilación de información sobre el aseguramiento de la calidad en aplicaciones Web mediante encuentros con especialistas y profesionales en el tema. Además de ser utilizado para profundizar los problemas detectados en los proyectos mediante intercambios con los encargados de la calidad. (Ver Anexo 3)

- Encuesta: Con la utilización de este método se adquiere información sobre el aseguramiento de la calidad de productos Web. Se hace uso de un cuestionario previamente elaborado, con la idea de conocer la opinión y valoración de la persona seleccionada y finalmente completar la información sobre los problemas existentes respecto a cómo asegurar la calidad del producto .(Ver anexo 2)

Población y muestra:

La población está constituida por todos los estudiantes, especialistas o profesionales, relacionados con el aseguramiento de la calidad de software en los proyectos productivos o de innovación y desarrollo de aplicaciones Web de la facultad 10. De ellos se toma como muestra el encargado de la calidad por cada proyecto.

El presente trabajo consta de tres capítulos, cada uno descrito a continuación:

Capítulo 1: “Fundamentación Teórica”: Muestra el estudio realizado en el área de la calidad, enfocado principalmente al Aseguramiento de la calidad del software. Consta de cinco epígrafes y un conjunto de sub epígrafes asociados al tema en cuestión: Conceptos asociados al problema tales como calidad de software, calidad del producto, y aplicaciones web; Antecedentes y actualidad de la calidad del software tanto en el mundo, en Cuba y en la UCI; Estándares, Normas y Modelos asociados específicamente a la calidad del software y a las aplicaciones web; Aseguramiento de la calidad del software, sus conceptos principales definidos por varios autores, métricas y factores coligados, descripción de la validación, verificación, control de calidad y gestión de la configuración; Sistema de gestión de conocimientos como parte del aprendizaje, con el fin de brindar las condiciones y ambiente necesario para que el conocimiento fluya. Para finalizar el capítulo se mencionan las conclusiones acaparadas.

Capítulo 2: “Propuesta de Solución”: Muestra la descripción de la propuesta de la guía para asegurar la calidad en los proyectos que desarrollan aplicaciones web. Consta de cuatro epígrafes y un sub epígrafes asociados a la propuesta descrita: Descripción de la propuesta para la universidad; Análisis del ciclo de vida del software de las aplicaciones web, desarrollo de cada una de las fases propuestas dentro del ciclo de vida del software, teniendo en cuenta su estructura, aspectos más importantes a desarrollar dentro de estas, y descripción de cada una de las fases (Planificación, Desarrollo, Operación), Además de las técnicas de uso y estudio de la guía. Finalmente se exhiben las conclusiones del capítulo.

Capítulo 3: “Validación de la Propuesta”: Muestras la validación realizada a la propuesta de solución desarrollada. Consta de dos epígrafes y subepígrafes asociados correspondientes a la validación desplegada: Método de validación, el cual fue realizado por caso de estudio; Ventajas de utilizar la guía en un proyecto real para lo cual se definió una situación antes y una situación después con el fin de probar la guía en grupo perteneciente algún proyecto. Posteriormente a los resultados obtenidos se describen las conclusiones del capítulo.

CAPÍTULO I: Fundamento Teórico

En este capítulo se describen los conceptos asociados al aseguramiento de la calidad del software como base a la investigación desarrollada. Se analizan la documentación investigada y se estudia el comportamiento del fenómeno a nivel mundial, en el país y en la UCI. Además se examinan normas, estándares y modelos que apoyen el aseguramiento de la calidad del software, con el fin de analizar los mismos y definir cuál puede ser viable para su uso en la posterior propuesta de solución. Luego se detallan aspectos importantes y aliados al aseguramiento de la calidad del software. Continuamente se hace un análisis de algunos aspectos de interés dentro del sistema de gestión de conocimiento, utilizando el modelo dML-UCI para gestionar conocimientos. Por último se hace una breve descripción de la propuesta de solución, dando paso a las conclusiones finales del capítulo.

1.1 Conceptos.

1.1.1 Calidad del producto software.

El concepto de calidad tiene varias definiciones formales que se han definido en el transcurso de la madurez que ha adquirido este tema. Algunas de ellas ponen mayor énfasis en el producto y otras en el cliente o el servicio. Sin embargo, todas tienen un mensaje común, que es el de obtener una mejora, ya sea en un producto software, o los procesos que llevan a la creación del mismo. Para acrecentar la calidad se establecen una serie de metodologías formales y buenas prácticas que permiten tener control sobre lo que está ocurriendo a distintos niveles y así poder evaluarlo y mejorarlo. Esto trae consigo no sólo que el producto final tenga las características que se anhelan (robustez, usabilidad, corrección, fiabilidad, etc.) sino que los procesos asociados se vean afectados positivamente, los tiempos de entrega disminuyan, los costes se reduzcan y la satisfacción del cliente aumente. La importancia que esta implica hace que se garantice la producción correcta del producto que se espera.

Distintos autores la han descrito de las siguientes maneras:

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” R. S. Pressman (1992).

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” ISO 8402 (UNE 66-001-92).

“La Calidad del Software es el conjunto de cualidades que la caracterizan y que determinan su utilidad y existencia, plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad ” (Scalone, 2006).

La calidad del producto software es una preocupación cada vez mayor en el ámbito de las tecnologías de la información. En los proyectos, grupos u organizaciones no es una actividad nueva la inquietud por ofrecer productos acompañados de altos niveles de calidad. Actualmente la satisfacción hacia el uso de un producto puede marcar la diferencia en el mercado de productos similares por lo que es primordial garantizar las características y cualidades necesarias para satisfacer las expectativas del cliente.

El concepto de calidad en los productos software debe formularse de forma particular. Hay que destacar que el producto software tiene características diferenciadoras de otros productos. El software se desarrolla, no se fabrica en el sentido clásico, es inmaterial y no se deteriora con el uso o el tiempo (aunque tiene un ciclo de vida). Su fiabilidad es difícil de comprobar. La mayoría del software se desarrolla a medida y necesita actualizaciones permanentes, además es dependiente del entorno donde se ejecuta. Todas estas características hacen del producto software algo particular. Además de destacar que para poder ofrecer un producto software de calidad, hay que observarlo en todo el ciclo de vida.

La calidad de un producto es vista generalmente desde dos enfoques tradicionales:

- ❖ Perceptiva: Satisfacción de las necesidades del cliente.
- ❖ Funcional: Cumplir con las especificaciones requeridas.

Las autoras la han definido como: Características o factores de un producto de software que contribuyen a la satisfacción total de las necesidades explícitas o implícitas de un usuario u organización, además de complacer sus expectativas.

1.1.2 Aplicaciones Web.

El auge que han alcanzado hoy en día las aplicaciones Web es un hecho, que igual a la introducción de Internet, hace que se abran infinidad de posibilidades en cuanto al acceso a la información desde cualquier parte que se encuentre una computadora conectada a red. Es por ello que aplicaciones Web se denomina a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o una intranet mediante un navegador.

Aunque existen muchas variaciones posibles, una aplicación Web está normalmente estructurada en su arquitectura como una aplicación de tres-capas. En su forma más común, el navegador Web ofrece la primera capa y un motor capaz de usar alguna tecnología Web dinámica constituye la capa intermedia dejando por último, una base de datos que constituye la tercera y última capa. El navegador Web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

La gran presencia de tecnología Web y la gran información asociada a esta tecnología hace imprescindible que los diseños se realicen bajo unos criterios mínimos de calidad, hasta ahora prácticamente inexistentes. La importancia que las tecnologías Web ha cobrado en los últimos años hace que se preste una atención especial a su calidad. Una forma de asegurar esta calidad es mediante la utilización de modelos específicamente definidos para el entorno Web, permitiendo controlar la calidad de los diseños realizados. Es por ello que, aquellas aplicaciones desarrolladas sin criterios de calidad tienen un pobre rendimiento y causan fallos, a lo que se ha hecho necesario que los sistemas Web sean gestionados y dirigidos de forma rigurosa y cualitativa ya que su ciclo de vida suele ser relativamente corto. Este proceso del ciclo de vida está dividido en un conjunto de actividades y cada actividad está dividida en un conjunto de tareas [Salanova 2006].

El hecho de conseguir que una aplicación Web sea de calidad es fundamental ya que es el único mecanismo para conseguir que los usuarios accedan nuevamente a esta.

1.2 Antecedentes y actualidad de la calidad del software.

1.2.1 Evolución de la calidad del software.

La calidad de un producto informático (software) se ha venido abordando desde hace varios años, y comenzó a profundizarse en ello desde que McCall y Boehm, según [Dávila 2006] desarrollaron los primeros modelos con el objetivo de mejorar la calidad del software producido.

El concepto calidad ha sido derivado debido al afán y búsqueda de la perfección por parte del hombre, de su interés por el trabajo bien hecho y la necesidad de asumir responsabilidades sobre la labor efectuada [ROMERO Arturo Luis y MIRANDA Sandor Luis] .El proceso evolutivo de la calidad de software viene enmarcado en cinco etapas principales que son mencionadas a continuación:

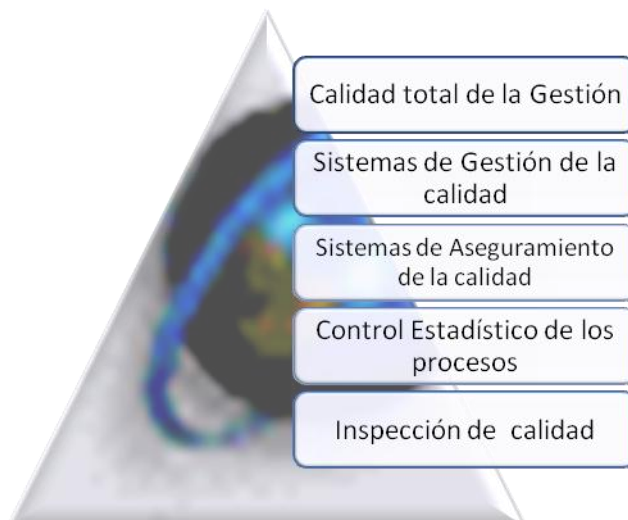


Figura 1: Fases de la Evolución de la Calidad del software

1.2.2 Calidad de Software en el mundo.

Es necesario tener en cuenta cómo evoluciona el mundo con la tecnología en un entorno cada día más global, más tele-comunicado, con grandes niveles educativos y experiencia profesional específicamente en la rama de la informática. Las empresas son directamente proporcionales a la tecnología por lo que trabajan día a día, con el objetivo de hacer grandes

software que puedan competir en el mercado internacional, lo que depende en grandes rasgos de la calidad total.

1.2.3 Calidad de Software en Cuba.

Para Cuba, el desarrollo de las tecnologías de información (TI) y la industria del software en general es uno de los objetivos necesarios e inmediatos que debe garantizar para incrementar su desempeño económico y lograr la informatización de la sociedad. El país cuenta con diversos sectores u organismos que aportan e incrementan el desarrollo tecnológico, convirtiéndose en substanciales fuentes de ingresos monetarias. En su mayoría estos sectores poseen una alta perspectiva económica, contribuyen en gran medida al desarrollo de la industria de software del país incrementando su competitividad en el mercado internacional, prestan servicios, imparten cursos, se realizan asesorías, entre otras funciones. Se citan ejemplos de estos sectores u organismos:

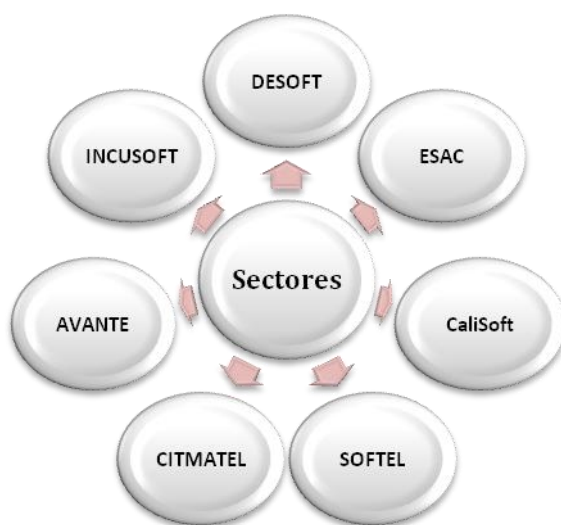


Figura 2: Empresas Cubanas desarrolladoras de Software

CITMATEL: Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados.

AVANTE (Ministerio de Informática y Comunicaciones): Agencia de Negocios para la promoción de Exportaciones de software, Productos y Servicios, perteneciente al Ministerio de la Informática y las contrataciones.

INCUSOFT: Industria Cubana del Software

DESOFT: Empresa Nacional de Software, entidad legal con personalidad jurídica.

ESAC: Empresa de Servicios de Aseguramiento de la Calidad.

SOFTEL: Empresa del Ministerio de la Informática y las Comunicaciones.

CaliSoft: Centro de calidad del software en Cuba.

A pesar de ello en algunos casos los resultados alcanzados no cubren las expectativas por la baja productividad, el trabajo que se realiza no presenta en su mayoría la calidad requerida, los proyectos no terminan en el tiempo establecido, los procesos son inestables e inmaduros en aquellos con poca experiencia en la producción del software, entre otros aspectos.

1.2.4 Calidad de Software en la UCI.

En la UCI se logró formar y consolidar una estructura productiva formada por los Proyectos Productivos de las diez Facultades. En aras a lograr una producción de Software con la calidad necesaria y poder insertar los productos en el mercado se crea la Dirección Central de Calidad (DCS) que dirige y controla la calidad en la universidad, y un Grupo de Calidad por facultad dirigidos por un Asesor de Calidad, que controla la calidad de los proyectos dentro de las facultades correspondiente. La DCS garantiza el crecimiento continuo de una producción de software con calidad en la organización, a través de la definición de procesos siguiendo las especificaciones de metodologías, estándares y modelos de desarrollo de software, brindando asesorías, entrenamiento, métodos de medición y servicios de verificación-validación a los diferentes entidades.

En la actualidad el centro está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (Capability Maturity Model Integration) bajo los servicios del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey, conteniendo a su vez 5 Niveles de madurez. El proceso de mejora está encaminado a que la universidad alcance en el 2010 una certificación internacional del nivel 2, lo que convertiría a la universidad en la primera empresa cubana certificada con este modelo y una de las pocas en el área del Caribe.

1.3 Estándares, Normas y Modelos de Calidad.

1.3.1 Descripciones.

Los estándares de calidad son normas y protocolos que deben cumplir determinados productos para su distribución y consumo por el cliente final. Suministran los medios para que los

procesos se realicen de la misma forma, convirtiéndose en guía para lograr la productividad y la calidad del software. En ellos la calidad se convierte en algo concreto, que se puede definir, medir y planificar.

La norma, es un documento que establece las condiciones mínimas que debe reunir un producto o servicio para que sirva al uso al que está destinado. La norma IRAM 50-1:1992 basada en la Guía ISO/IEC 2:1991, la define de esta forma “*es un documento establecido por consenso y aprobado por un organismo reconocido que establece, para usos comunes y repetidos, reglas, criterios o características para las actividades o sus resultados, que procura la obtención de un nivel óptimo de ordenamiento en un contexto determinado*”.

Los modelos de calidad son herramientas que guían a los grupos o proyectos a la mejora continua y a la competitividad. Sin embargo para aplicar este modelo se requiere de mucho esfuerzo y compromiso de toda la organización, primero se diseña y documentan los procesos, luego se despliegan y por último se ponen en práctica [Quiñones 2007]. Son utilizados con el fin de construir mejores productos y asegurar su calidad en todo su ciclo de vida. “La calidad de un producto software debe evaluarse usando un modelo de calidad que tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquirentes y usuarios finales” (ISO, 2001).

Los modelos de calidad son los que te dicen QUE hacer en la obtención de un software, por lo que se han desarrollado varios para diferentes productos y procesos. La mayor parte están basados en la norma ISO9126. Esta norma define un conjunto de características de calidad que son después refinadas en sub-características que están descompuestas en atributos. Los valores de estos atributos se calculan mediante la utilización de métricas.

En este epígrafe se hace referencia a algunos estándares, normas y modelos de calidad que se utilizan en los proyectos que desarrollan aplicaciones Web en la universidad y otros de suma importancia dentro de los procesos para lo obtención de un producto específico.

1.3.2 CMMI.

CMMI (Modelo Integrado de Capacidad de Madurez) se dio a conocer en diciembre del 2000 por el SEI. Es tratado a través de dos representaciones: continua y escalonada. La representación escalonada que es la que se implementa en la universidad clasifica las

empresas en niveles de madurez que sirven para conocer el grado de experiencia de los procesos desarrollados para producir software.

CMMI se puede utilizar para guiar el proceso de mejora a través de un proyecto, una división, o de toda una organización. Además ayuda a integrar funciones tradicionalmente separadas de la organización, establecer los objetivos y las prioridades de la mejora del proceso, facilitar la orientación de procesos de calidad, y proporcionar un punto de referencia para la evaluación de los procesos actuales. [SEI, 2008].

CMMI propone 5 distintos modelos de madurez para medir la capacidad de los procesos:

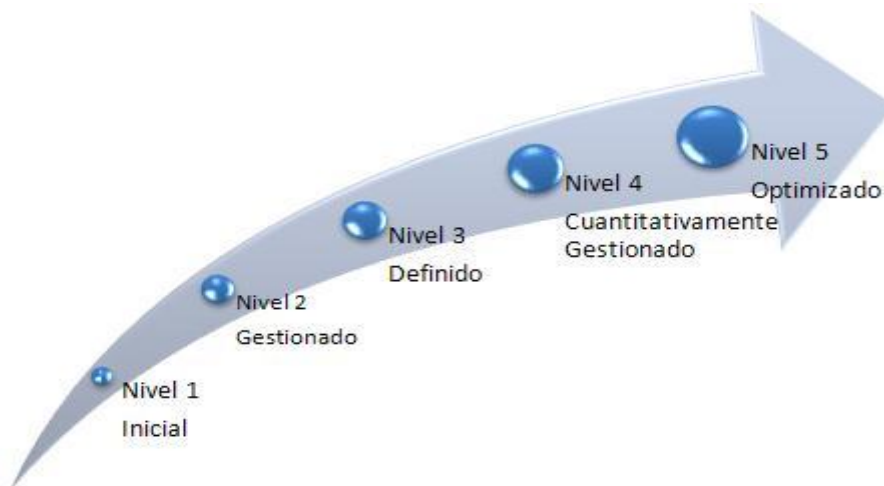


Figura 3: Niveles del modelo CMMI.

Nivel 1 Inicial: Estado inicial donde el desarrollo se basa en la heroicidad y responsabilidad de los individuos.

- Los procedimientos son inexistentes o localizados a áreas concretas.
- No existen plantillas definidas a nivel corporativo.

Nivel 2 Gestionado: Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

- En este nivel consolidado, las buenas prácticas se mantienen en los momentos de estrés.
- Están definidos los productos a realizar.

- Se definen hitos para la revisión de los productos.

Nivel 3 Definido: La organización entera participa en el proceso eficiente de proyecto software.

- Se conoce de antemano los procesos de construcción de software. Existen métodos y plantillas bien definidas y documentados.
- Los procesos no solo afectan a los equipos de desarrollo sino a toda la organización relacionada.
- Los proyectos se pueden definir cualitativamente.

Nivel 4 Cuantitativamente Gestionado:

- Se puede seguir con indicadores numéricos (estadísticos) la evolución de los proyectos.
- Las estadísticas son almacenadas para aprovechar su aportación en siguientes proyectos.
- Los proyectos se pueden pedir cuantitativamente.

Nivel 5 Optimizado:

- En base a criterios cuantitativos se pueden determinar las desviaciones más comunes y optimizar procesos.
- En los siguientes proyectos se produce una reducción de costes gracias a la anticipación de problemas y la continua revisión de procesos conflictivos.

En el apartado anterior se explica que la UCI está trazando una estrategia de mejora de sus procesos basado en el modelo CMMI. Este proceso está encaminado a que se alcance una certificación internacional del nivel 2. Lo que se pretende con este nivel es conseguir que en los proyectos haya una gestión de los requisitos organizada y que los procesos estén planeados, ejecutados, medidos y controlados.

1.3.3 Normas ISO para la Calidad de Software.

Las normas ISO (International Organization for Standardization) serie 9000, han tenido una gran difusión y aplicación en todo el mundo. La importancia de la aplicación de estas para el desarrollo e implementación de sistemas de aseguramiento de la calidad radica en que son normas prácticas, no normas académicas. Por su sencillez han permitido su aplicación generalizada sobre todo en pequeñas y medianas empresas. Siendo la calidad hoy uno de los factores esenciales de la competencia en cualquier actividad, se ha generado la necesidad de implementar sistemas normalizados de aseguramiento de la calidad. Las normas ISO Serie 9000 brindan el marco para documentar en forma efectiva los distintos elementos de un sistema de calidad y mantener la eficiencia del mismo dentro de la organización. La serie ISO 9000 es un conjunto de cinco normas relacionadas entre sí, son normas genéricas, no específicas que permiten ser usadas en cualquier actividad ya sea industrial o de servicios. Las series de ISO 9000 son un grupo de 5, distintas pero relacionadas entre sí, normas internacionales de administración de la calidad y aseguramiento de calidad, las mismas son:

➤ **ISO 9000**

Cumple el papel de eje distribuidor y distribuidor del sistema. Expone el alcance real de la serie. Define la filosofía general de las normas los distintos tipos, niveles y pautas para la aplicación de las distintas normas.

➤ **ISO 9001**

Se aplica cuando la empresa debe responsabilizarse por todas las etapas del ciclo, es decir: diseño, desarrollo y elaboración.

➤ **ISO 9002**

Se aplica cuando las características del bien o servicio son definidas por el cliente.

➤ **ISO 9003**

Cubre las obligaciones de aseguramiento de calidad en las áreas de control final y pruebas. Es de limitada aplicación por lo que existen planes para su eliminación.

En los casos de exigencia contractual las normas aplicables son las normas ISO 9001/2/3. La norma a aplicar depende del alcance de la actividad de la empresa, no de una elección a voluntad.

➤ **ISO 9004-1/ ISO 9004-2**

Establecen condiciones y pautas para guiar a las empresas en la implementación de su propio sistema de aseguramiento de calidad. Su desarrollo no es válido para certificación o registro.

1.3.4 SPICE.

SPICE es una norma que trata los procesos de ingeniería, gestión, relación cliente-proveedor, de la organización y del soporte. Se creó por la alta competencia del mercado de desarrollo de software, a la difícil tarea de identificar los riesgos, cumplir con el calendario, controlar los costos y mejorar la eficiencia y calidad. Engloba un modelo de referencia para los procesos y sus potencialidades sobre la base de la experiencia de compañías grandes, medianas y pequeñas.

La norma posee una arquitectura basada en dos dimensiones: de proceso y de capacidad de proceso. 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. Desde la dimensión de capacidad el modelo define una escala de 6 niveles para determinar la capacidad de cualquier proceso:

- Nivel 0: Incompleto
- Nivel 1: Realizado
- Nivel 2: Gestionado
- Nivel 3: Establecido
- Nivel 4: Predecible
- Nivel 5: En optimización

1.3.5 IEEE.

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

Según el mismo IEEE, su trabajo es promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para beneficio de la humanidad y de los mismos profesionales.

El objetivo de IEEE es fundamental ya que se propone promover el avance de las teorías y las prácticas de electro tecnología, fomentar el progreso y el desarrollo profesional de su membrecía, mejorar la calidad de vida a través de la aplicación de electro tecnología, así como promover el entendimiento de electro tecnología ante el público.

1.3.6 W3C.

El Consorcio World Wide Web (W3C) (consorcio de internet) es un consorcio internacional donde las organizaciones miembros, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web. La misión del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web.

Uno de los objetivos a largo plazo del W3C es promover tecnologías que hagan posible un entorno más colaborativo, una Web donde la responsabilidad, la seguridad, la confianza, y la confidencialidad sean posibles, y en la que las personas participen de acuerdo con sus requisitos y preferencias de privacidad.

El valor social que aporta la Web, es que ésta hace posible la comunicación humana, el comercio y las oportunidades para compartir conocimiento. Uno de los objetivos principales del W3C es hacer que estos beneficios estén disponibles para todo el mundo, independientemente del hardware, software, infraestructura de red, idioma, cultura, localización geográfica, o habilidad física o mental.

1.3.7 Los que definen Ciclo de vida.

La mayor parte de las organizaciones cuando se refieren a un ciclo de vida de un software se describe al período de tiempo que comienza, cuando se concibe la idea de generar el programa hasta que finalmente se produce. Los ciclos de vida ofrecen información general de las etapas por las que atraviesa un software. Es dentro de ellos que se llevan a cabo actividades y tareas correspondientes a diferentes áreas de trabajo siendo estas de vital importancia. 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. En este epígrafe se mencionan normas, modelos y estándares que definen actividades correspondientes al ciclo de vida de las aplicaciones Web.

ISO/IEC 12207 Tecnología de la información. Procesos del ciclo de vida del software.

Esta Norma Internacional se encarga de establecer una estructura común para los procesos del ciclo de vida del software, con una terminología bien definida, pudiendo ser tomada como referencia por la industria del software. Contiene procesos, actividades y tareas que se aplican durante la adquisición de un sistema que contiene software, un producto de software, un servicio de software y durante el suministro, desarrollo, operación y mantenimiento de productos de software.

Los procesos se clasifican en tres grandes grupos. En la figura 2 se muestran, al igual que los procesos que integran estos grupos.

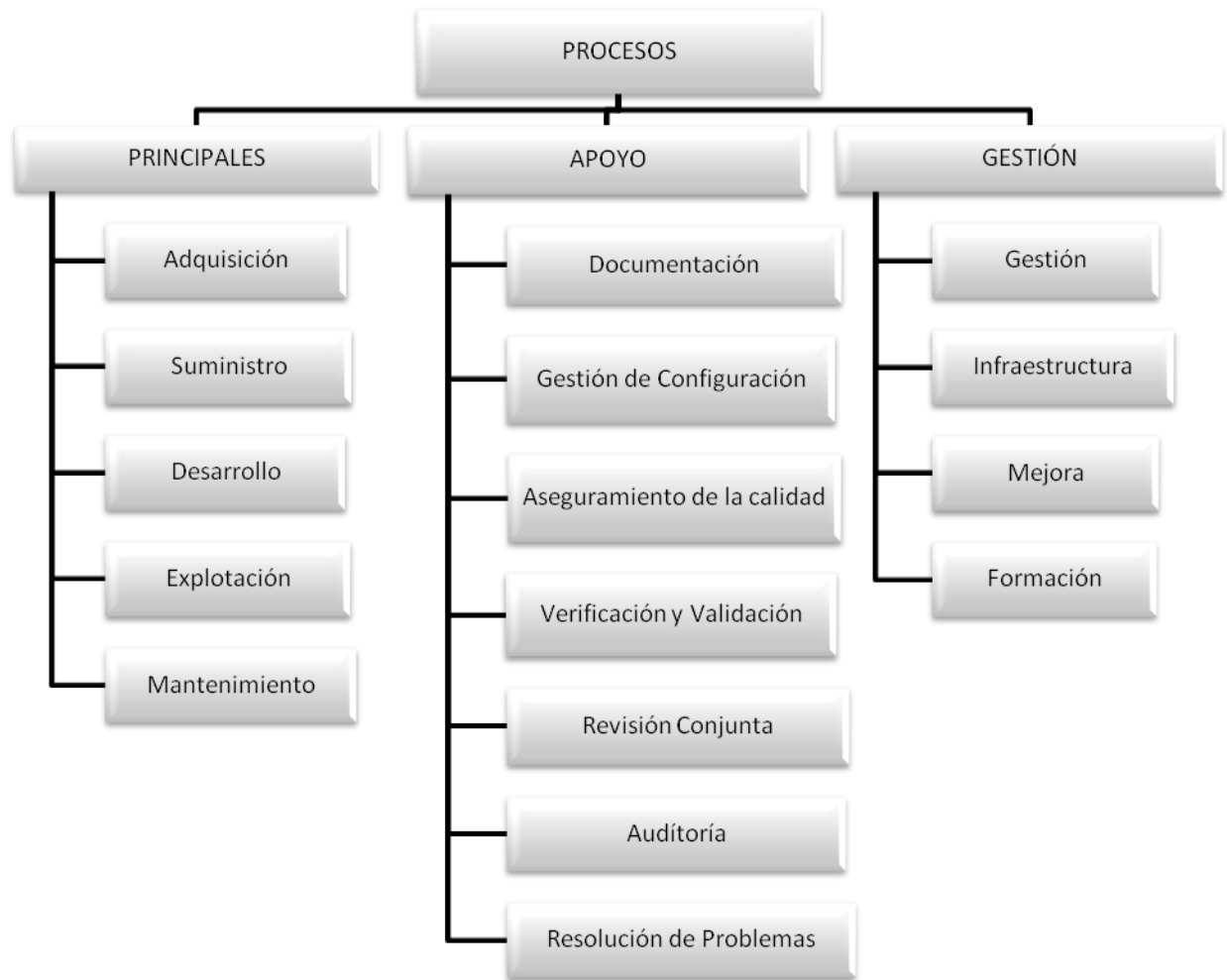


Figura 2: Procesos que integran a la ISO 12207.

La ISO 12207 puede emplearse para definir, controlar y mejorar procesos del ciclo de la vida del software. Describe la arquitectura de los procesos, pero no especifica los detalles de cómo implementar o ejecutar las actividades y tareas incluidas en los procesos. Está escrita preferentemente para adquiridores de sistemas así como de productos y servicios de software, para suministradores, desarrolladores, operadores, mantenedores, gerentes, gestores del aseguramiento de la calidad y usuarios de los productos de software.

Los procesos que desarrolla esta norma son los primarios (apoyo y organizacionales). Basado en esta norma se encuentra el WQM. Modelo donde se analizan los elementos para definir el ciclo de vida de una aplicación Web (más adelante se dan detalles de este modelo).

WQM (Web Quality Model).

WQM es un modelo tridimensional de calidad de los sitios Web que se desarrolló con el objetivo de paliar la situación existente en el ámbito de la Web. Se puede utilizar para clasificar tanto las métricas Web, como los trabajos de investigación realizados sobre la Web. No es un modelo excluyente con otros modelos de calidad existentes, sino que pretende ser aglutinador de los mismos, tratando de ser por tanto un modelo global de calidad de la Web.

Está caracterizado por tres elementos:

- ❖ La característica de calidad (basada en Quint2 y en la ISO 9126).
- ❖ El proceso del ciclo de vida (basado en la ISO12207).
- ❖ Características (contenido, presentación y navegación).

A continuación se describen brevemente en qué se basan cada una, enfocando la descripción principalmente en el proceso del ciclo de vida que es el elemento de interés.

La característica de calidad: Como base utiliza el modelo Quint2 que se basa en la ISO 9126 y amplía el estándar con nuevas características, apropiadas para productos Web: (1) Funcionalidad, (2) Fiabilidad, (3) Usabilidad, (4) Eficiencia, (5) Portabilidad, (6) Mantenibilidad.

El proceso del ciclo de vida: Basada en el estándar ISO 12207.

WQM define tres procesos principales incluidos en la ISO 12207, que son los que especifican el ciclo de vida para aplicaciones Web, por lo que se tendrán en cuenta para asegurar la calidad de cada una de las actividades y tareas que engloban estos procesos. Es válido destacar que los procesos determinados forman parte de los procesos principales dentro de la

ISO 12207. A continuación se muestran:

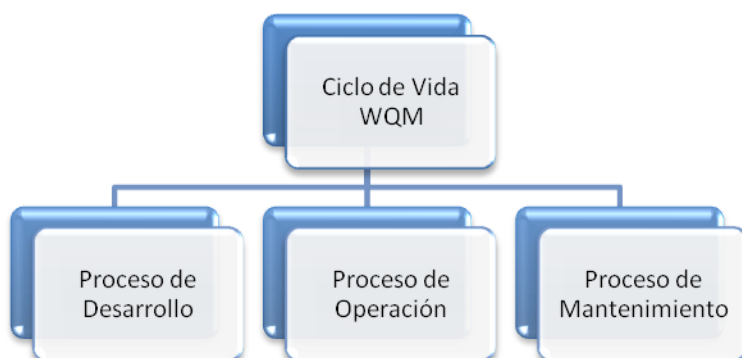


Figura 3: Procesos del ciclo de vida del modelo WQM.

Cada proceso define un cúmulo de actividades las cuales serán analizadas en próximos epígrafes.

Características: Se basa en tres de los factores más utilizados en la literatura para caracterizar un sitio Web: (1) Contenido (2) Presentación (3) Navegación. Además de englobar otros datos, validaciones formales, validaciones empíricas, automatización, entre otros.

Para definir el ciclo de vida de las aplicaciones Web las autoras proponen WQM por ser un modelo de calidad del software aplicado específicamente a la Web.

1.4 Aseguramiento de la calidad del software.

1.4.1 Descripción del aseguramiento de la Calidad.

Cualquier acción que es tomada con el fin de dar a los consumidores productos de calidad adecuada, ya sea bienes o servicios, se puede resumir en aseguramiento de la calidad.

El aseguramiento de la calidad es una actividad de protección, que se utiliza durante todo el proceso de desarrollo del software. Comprende procedimientos para una mejor aplicación de métodos y herramientas, revisiones técnicas formales, técnicas y estrategias de pruebas, procedimientos de garantía de ajuste a los estándares y mecanismos de medida e información. Y por supuesto, incluye la garantía de que un sistema cumpla las especificaciones y los requisitos para el uso y desempeño deseado. (Dapena, 2005).

El aseguramiento de la calidad contiene todas aquellas actividades que se ejecutan con el objetivo de asegurar un nivel de calidad en el producto desarrollado realizándose de forma independiente al equipo de desarrollo. (Pressman, 2005)

Según la Norma ISO 9000:2000, el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad. (ISO 9000, 2000)

El objetivo del aseguramiento de la calidad es facilitar la gestión para comunicar los informes necesarios referentes a la calidad del producto, por lo que se va consiguiendo una visión más

amplia y segura de que la calidad del producto está cumpliendo sus objetivos. Si los datos proporcionados a través del aseguramiento de la calidad identifican problemas, es responsabilidad de la gestión enfrentar los problemas y utilizar los recursos necesarios para solucionar aspectos de calidad. (Pressman, 2005).

El aseguramiento de la calidad de software se entiende como el conjunto de acciones planificadas y sistemáticas que son necesarias para proporcionar la confianza suficiente de que un producto satisface los requisitos de calidad dados. Está basado en la prevención y es más productivo prevenir los fallos de calidad que modificarlos. Además existe la necesidad de seleccionar y aplicar una misma forma de controlar el desarrollo del software para todas las etapas de vida de un proyecto.

El aseguramiento de calidad del software está presente en:

- ❖ Métodos y herramientas de análisis, diseño, programación y prueba.
- ❖ Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- ❖ Estrategias de prueba multiescala.
- ❖ Control de la documentación del software y de los cambios realizados.
- ❖ Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- ❖ Mecanismos de medida (métricas).
- ❖ Registro de auditorías y realización de informes.

1.4.2 Actividades de aseguramiento de la calidad.

Las actividades de aseguramiento de la calidad se pueden categorizar en:

- ❖ Software testing. Donde se recogen tanto pruebas de verificación como de validación y que suponen una de las más populares estrategias de gestión del riesgo. Usadas para comprobar que los documentos de diseño del software, el propio software y la documentación cumplen con todos los requisitos. El plan de pruebas recoge y clasifica todas las pruebas del software, manteniendo la trazabilidad entre las mismas y los requisitos que cubre.

- ❖ Control de la calidad. Consiste en los procesos y procedimientos empleados para monitorizar el trabajo realizado. Se centra en la revisión y subsanación de defectos antes de realizar las entregas al cliente. Típicamente consiste en revisiones de documentos, inspecciones de código y comprobaciones previas a cualquier entrega al usuario. Los criterios de control suelen adaptarse para cada proyecto en el plan de calidad del proyecto software, haciendo al cliente partícipe del mismo para establecer los criterios que mejor respondan a sus necesidades.
- ❖ Gestión de configuración del software. Relacionado con el etiquetado de los elementos de los proyectos software y el seguimiento y el control de cambios que permite monitorizar la evolución de dichos elementos y sus interrelaciones a lo largo del ciclo de vida del producto. [10]

Para definir las actividades correspondientes al aseguramiento de la calidad del software se hace un estudio de aquellas que proponen algunos estándares, modelos, normas, autores y alguno que otro plan de aseguramiento de la calidad del software.

ROGER S. PRESSMAN propone las siguientes actividades:

- ✓ Establecimiento de un plan de aseguramiento de la calidad para el proyecto.
- ✓ Participación en el desarrollo de la descripción del proceso de software del proyecto.
- ✓ Revisión de las actividades de ingeniería del software para comprobar su ajuste al proceso de software definido.
- ✓ Auditorías de los productos de software escogidos para verificar el ajuste con los definidos como parte del proceso del software.
- ✓ Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- ✓ Registrar lo que no se ajuste a los requisitos y comunicar a sus superiores.

Actividades que define la ISO 12 207:

- ✓ Implementación del proceso.
- ✓ Aseguramiento del producto.
- ✓ Aseguramiento del proceso.
- ✓ Aseguramiento del sistema de calidad.

Actividades que define el Plan de Aseguramiento de la Calidad del software de proyectos que desarrollan aplicaciones Web en la universidad.

- ✓ Establecimiento del Plan de Aseguramiento de la calidad.
- ✓ Mantener actualizados los documentos del proyecto (Expediente de proyecto).
- ✓ Revisión de los requerimientos.
- ✓ Revisiones a la documentación.
- ✓ Certificación final del producto.
- ✓ Certificación por parte del Cliente.
- ✓ Evaluación de la satisfacción del cliente.
- ✓ Participar en las Auditorías y Revisiones de Calidad.

1.4.3 Aseguramiento de la calidad del producto.

Según [Calero 2005 a] “la calidad es la adecuación del producto al uso. Conformidad con requisitos y confiabilidad en el funcionamiento. Cero defectos...” para ello el aseguramiento de la calidad debe proveer la gestión con una adecuada visibilidad en los productos que están siendo construidos. Este incluye las revisiones, las auditorías de los productos y de las actividades para verificar que ellas cumplen con los procedimientos y estándares y suministra los resultados al jefe de proyecto y otros administradores.

1.4.4 Métricas asociadas al aseguramiento.

Las métricas pueden ser utilizadas para que los profesionales e investigadores puedan tomar las mejores decisiones (Pfleeger, 1997).

El IEEE “Standard Glossary of Software Engineering Terms” define como métrica como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [Len O. Ejiogo '91]

La medición persigue tres objetivos fundamentales (Fenton y Pfleeger, 1997)

- ✓ Entender qué ocurre durante el desarrollo y el mantenimiento.
- ✓ Controlar qué es lo que ocurre en nuestros proyectos.
- ✓ Mejorar nuestros procesos y nuestros productos.

Las métricas de software proveen la información necesaria para la toma de decisiones técnicas y son importantes para:

- ✓ Para indicar la calidad del producto.
- ✓ Para evaluar la productividad de las personas.
- ✓ Para evaluar los beneficios derivados del uso de nuevos métodos y herramientas.
- ✓ Para establecer una línea de base para la estimación.
- ✓ Para justificar el uso de nuevas herramientas y la necesidad de formación.

Ejemplos de atributos a medir en el producto:

- La fiabilidad del código.
- La entendibilidad de un documento de especificación.
- La mantenibilidad del código fuente.
- La longitud, funcionalidad, modularidad o corrección sintáctica de los documentos de especificación.

1.4.5 Factores de Calidad.

Los factores de calidad son parámetros para medir la calidad de un producto de software. Un proyecto que maneja información confidencial puede que tenga la seguridad muy alta, una página de internet tendrá muy alta la facilidad de uso, mientras que un software bancario tomará más en cuenta la fiabilidad que la portabilidad. Para determinar el uso correcto que se le puede dar se debe estar seguro de la real necesidad del cliente, para posteriormente lograr que estas puedan ser modificadas, probadas y adaptables a cualquier entorno de trabajo.

McCall propone los siguientes factores de calidad:

El modelo se basa en 11 factores de calidad, que se organizan de la siguiente forma teniendo en cuenta la visión del usuario:

1. Operaciones del producto: características operativas.

Corrección. ¿Hace lo que se quiere?

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

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

Seguridad (Integridad). ¿Es seguro?

Facilidad de Uso. ¿Está diseñado para ser usado?

2. Revisión del producto: capacidad de soportar los cambios.

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

Flexibilidad. ¿Se puede cambiar?

Facilidad de Prueba. ¿Se puede probar?

3. Transición del producto: adaptabilidad a nuevos entornos.

Portabilidad. ¿Se podrá usar en otra máquina?

Reusabilidad. ¿Se podrá reutilizar alguna parte del software?

Interoperabilidad. ¿Se podrá hacerlo interactuar con otro sistema?

Factores propuestos por Olsina:

Olsina y sus colaboradores han preparado un árbol de requisitos de calidad que identifica un conjunto de atributos que conduce a aplicaciones Web de alta calidad [Pressman, 2002]. La Figura 4 resume su trabajo.

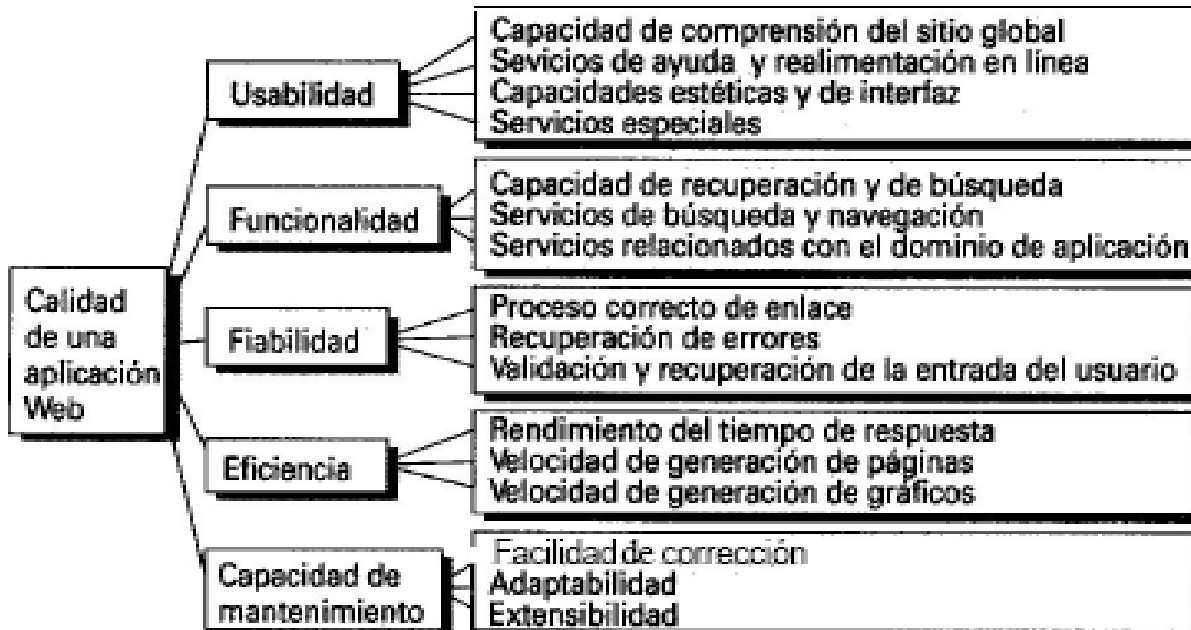


Figura 4: Factores de Calidad.

A pesar de Olsina proponer cinco factores de calidad que sirven para cualquier software que se quiera desarrollar, las autoras en el presente escrito solo trabajan con cuatro de ellos, pues estos son los más medidos en las aplicaciones Web y los más exigidos por los usuarios (usabilidad, funcionalidad, fiabilidad, eficiencia).

1.4.6 Verificación y Validación.

La Verificación y Validación son dos términos de gran importancia a la hora de asegurar la calidad del producto durante su ciclo de vida.

La verificación comprueba que el producto cumple con los requisitos establecidos y ayuda a asegurar que el producto se está desarrollando de la forma correcta. Es un proceso incremental ya que ocurre a lo largo del desarrollo del producto, empezando en la verificación de los requisitos, avanzando a la verificación de la evolución del mismo y finalizando mediante la verificación del producto una vez que está completo. Una importante parte de la verificación son las revisiones entre pares (*peer reviews*).

Sin embargo, la validación cambia el enfoque. Evalúa el producto en función de las necesidades del cliente asegurando que el producto satisface el uso por el cual se creó cuando trabaja sobre el entorno adecuado.

Las actividades de validación son similares a las de verificación, por ejemplo ambas llevan a cabo pruebas, análisis, inspecciones, y normalmente se ejecutan de forma concurrente.

CMMI presenta varias 22 áreas de proceso. En el trabajo se hace uso específico de solo dos de ellas, Validación (VAL) y Verificación (VER), correspondiente al nivel 3 de CMMI. Cada una describe y dice a través de prácticas y objetivos tanto específicas como generales para que el producto, componentes del producto y artefactos correspondan a lo esperado para su uso y cumplan con los requerimientos establecidos respectivamente.

La VER y VAL son similares pero tienen diferencias.

Verificación	Validación
Se enfoca en que los artefactos reflejen los requerimientos especificados.	Demuestra que el producto que se entrega satisface su uso.
Asegurar que el producto se construya correctamente.	Asegurar que se construya el producto correcto.
Asegurar que los artefactos cumplen con los requerimientos especificados.	Demstrar que un producto o componentes del producto cumplen su uso planeado cuando es ubicado en el ambiente de usuario final.

Tabla 1: Diferencias entre VER y VAL.

CMMI en el área de VER y VAL propone algunas actividades las cuales se definen por prácticas específicas para el cumplimiento de cada meta, siendo estas un componente principal para asegurar la calidad de cualquier software.

Actividades de la VER:

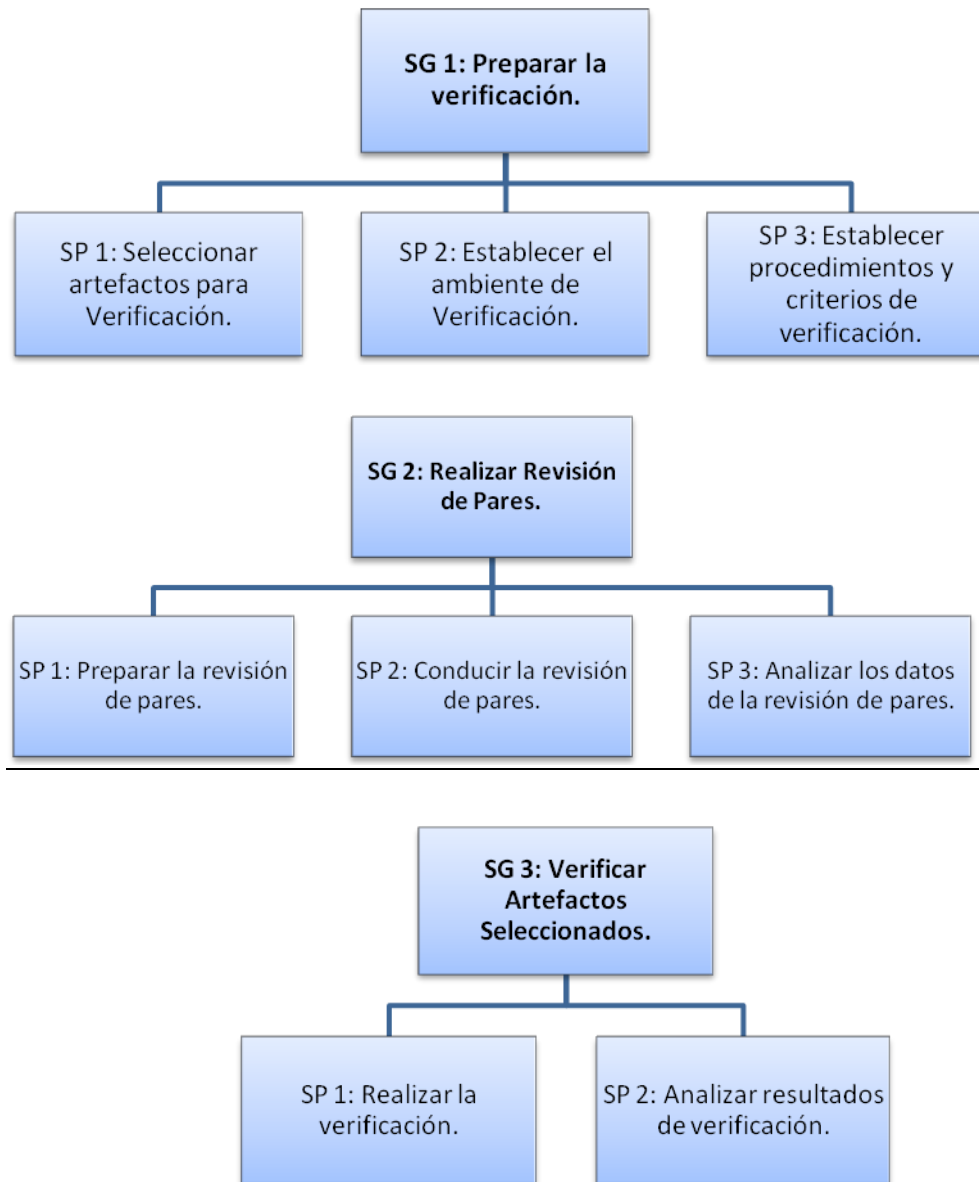


Figura 5: Actividades de la Verificación.

Actividades de VAL:

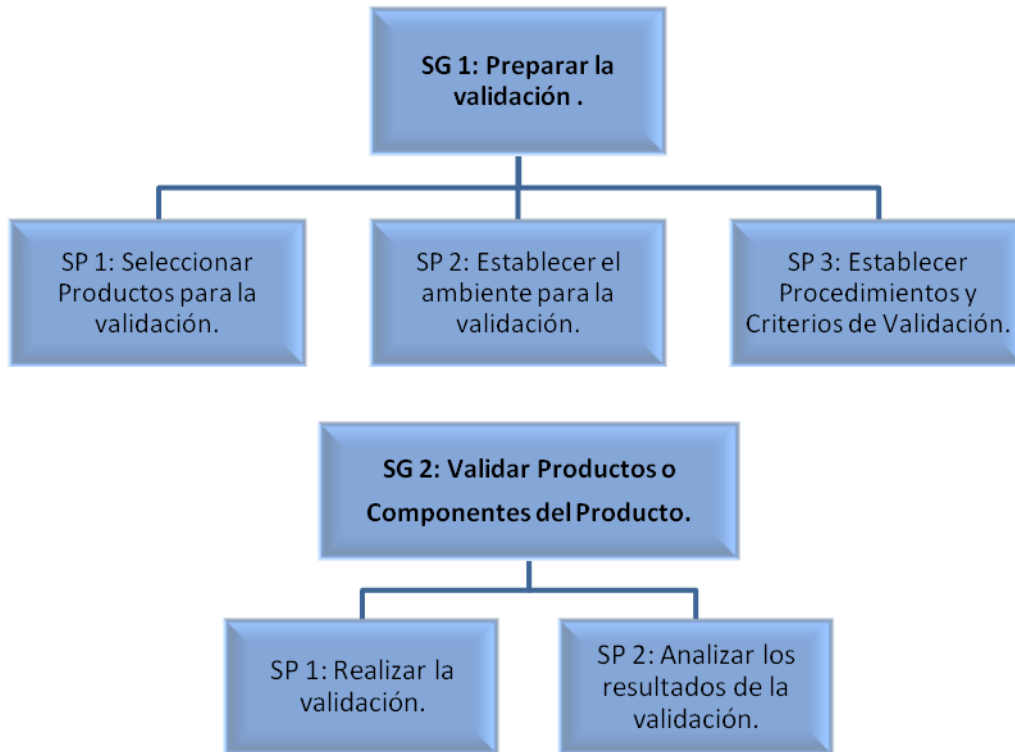


Figura 6: Actividades de la Validación.

1.4.7 Control de la Calidad.

El Control de la Calidad (CA): Es una forma de realizarle auditorías, inspecciones y pruebas al producto, con el fin de verificar o cerciorarse de que el producto cumple con los requerimientos ya establecidos. Torregrosa la define como: *“El control de la calidad es el conjunto de técnicas y actividades, de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad del producto o servicio” (Torregrosa).*

La realización del control de la calidad consiste en el monitoreo de resultados específicos del proyecto para determinar si cumple con los estándares de calidad relevantes. Además de identificar medios para eliminar las causas de resultados insatisfactorios. Es por ello que tiene como objetivo fundamental detectar y corregir los errores que van saliendo en todo el proceso de desarrollo del producto, además de asegurar que el producto final cumpla con los intereses del cliente.

1.4.8 Gestión de la configuración.

Se denomina Gestión de la Configuración al conjunto de procesos destinados a asegurar la validez de todo producto obtenido durante cualquiera de las etapas del desarrollo de un Sistema de Información (S.I.), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo.

La gestión de la configuración se realiza durante todas las fases del desarrollo de un sistema, y continúa registrando los cambios hasta que éste deja de utilizarse, incluyendo el mantenimiento y control de cambios, una vez realizada la puesta en producción. Su objetivo es mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan.

Según la interfaz Gestión de la Configuración definida en Métrica v3, los elementos de configuración del software incluyen:

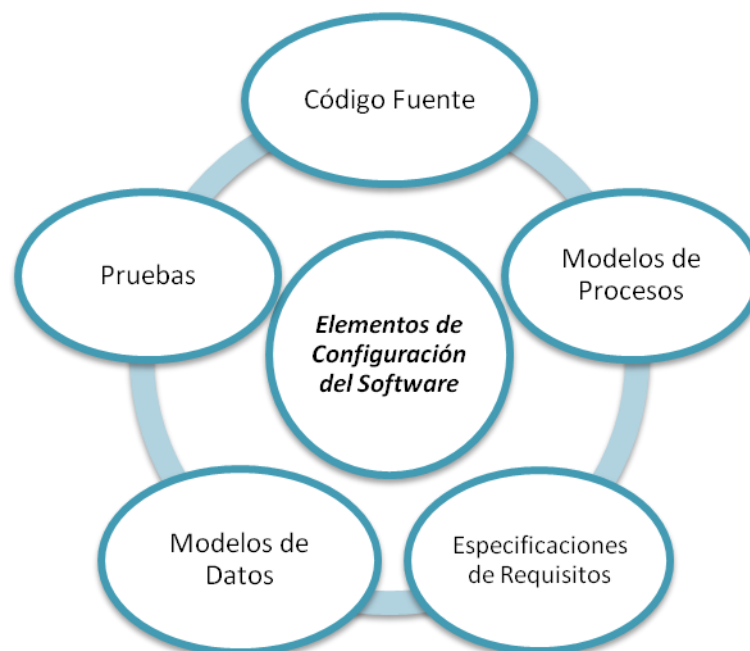


Figura 7: Elementos de Configuración.

La gestión de configuración facilita el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo. Asimismo, permite controlar el sistema como producto global a lo largo de su desarrollo, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de mejora de la organización.

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

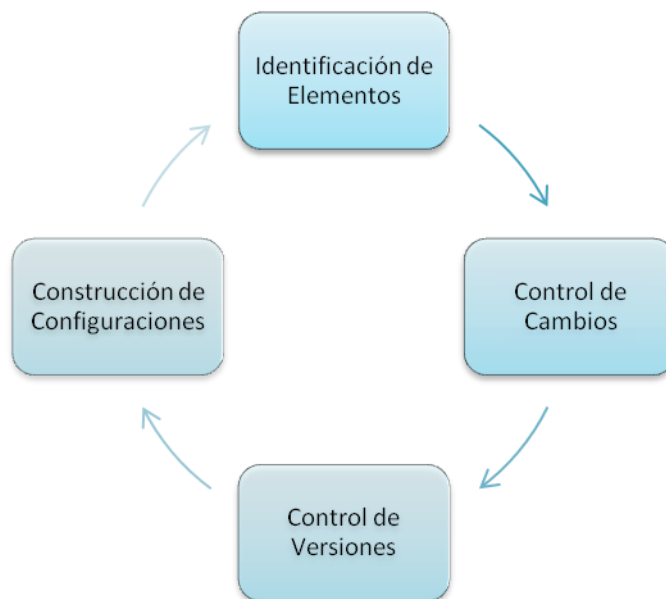


Figura 8: Actividades de los procesos de Gestión de Configuración.

En el presente trabajo se hace uso principalmente del control de cambio. Este 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.

Se aplica durante el desarrollo del software y, posteriormente, durante su mantenimiento. Como el cambio se puede producir en cualquier momento, las actividades de la gestión de configuraciones del software sirven para:

(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.

1.5 Sistema de Gestión de Conocimiento.

Un sistema de gestión de conocimiento se basa entre otras acciones, en dirigir y controlar una organización. Es un medio mediante el cual se efectúan diversas actividades, tales como: gestionar condiciones y ambientes para la creación, transformación continua y transferencia de datos. Presenta como pilares fundamentales, los recursos humanos, los conocimientos y las tecnologías de la información y las comunicaciones.

Su objetivo principal es lograr la eliminación de deficiencias, existentes hoy, en el aprendizaje, con el fin de brindar las condiciones y ambiente necesario para que el conocimiento fluya de manera eficaz. Para el centro universitario UCI, esta propuesta juega un papel fundamental, ya que ayuda a adquirir y perfeccionar el conocimiento de cada persona que se desarrolle en esta rama. [FERNANDEZ; GUERRERO. 2008].

La Gestión del conocimiento es el proceso mediante el cual se adquiere, se genera, almacena, se comparte y se utiliza conocimiento, información, ideas y experiencias para mejorar la calidad en el cumplimiento y desarrollo del trabajo de la organización. Obteniendo como resultado:

- ❖ Aprender de él.
- ❖ Mejorar la toma de decisiones, como riqueza de crecimiento personal y también organizacional.
- ❖ Capitalizando el potencial intelectual de los recursos humanos, y así enriquecer de conocimiento al personal con el que se está trabajando.
- ❖ Mejorar la calidad del trabajo.

Modelo dML-UCI para gestionar conocimientos.

dML es un modelo para la gestión del conocimiento con el fin de eliminar las deficiencias en cualquier área que lo requiera así como ambientes y condiciones, con el único fin de que el conocimiento fluya eficazmente. A continuación se muestra una figura que lo describe.

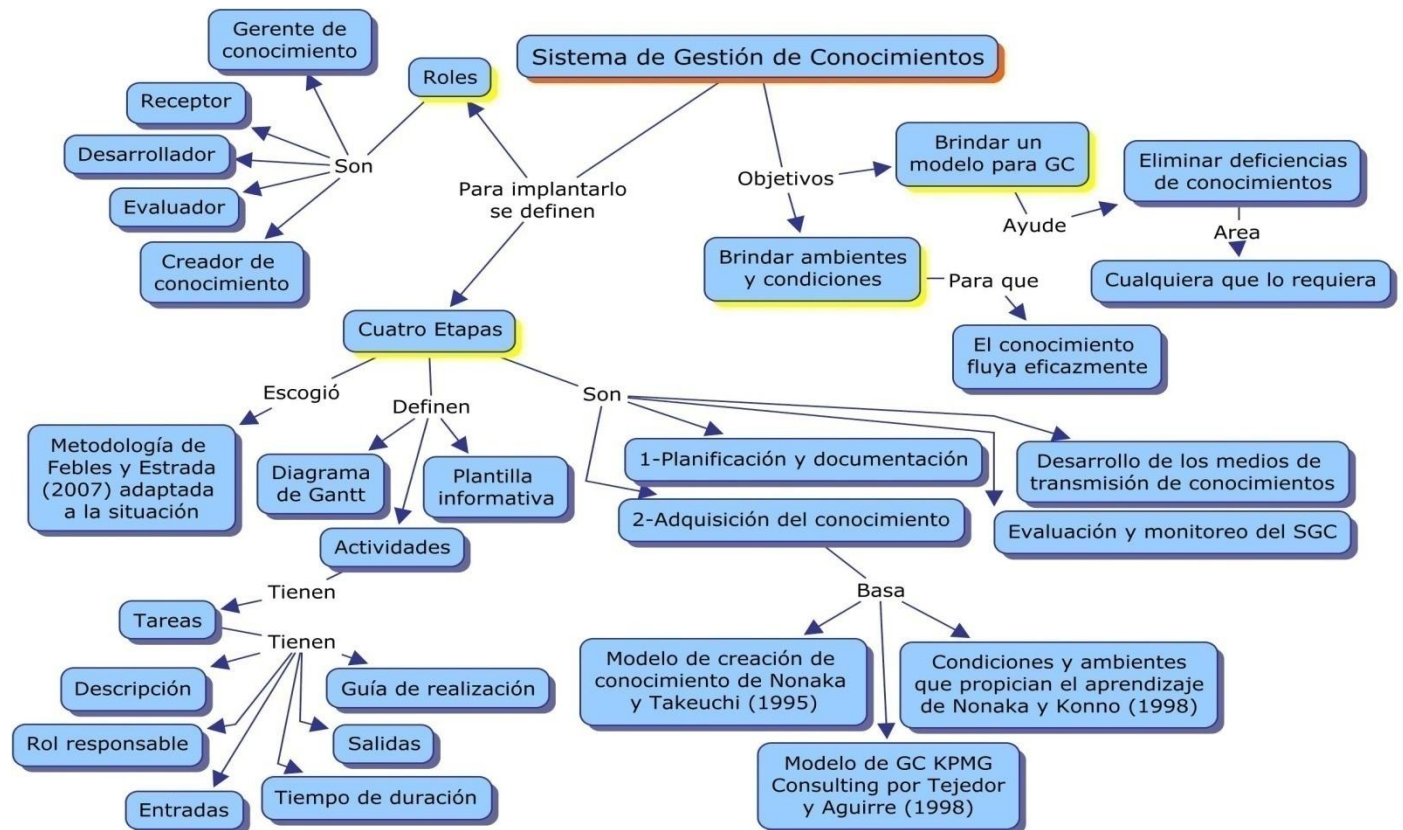


Figura 9: Metodología dML.

La propuesta de solución desarrollada en este trabajo, es el resultado de tratar de eliminar deficiencias existentes hoy en el área de aseguramiento de la calidad del producto en el proceso de desarrollo de software de la facultad 10. Para ello hace uso de modelo dML representado, utilizando de este las siguientes fases y roles.

Fases que utiliza la guía en su desarrollo.

La 1ra Etapa.

Está basada en encontrar las deficiencias que existen en el aseguramiento de la calidad de las aplicaciones Web. Además de acumular toda la información y documentación necesaria para la creación de la guía y planificar las tareas definidas en el desarrollo del trabajo.

La 2da Etapa.

Se estudia toda la información seleccionada hasta llegar a una teoría propia y adquirir a su vez todo el conocimiento necesario.

La 3ra Etapa.

Se usa para desarrollar el sistema planteado para la transmisión de la investigación realizada.

Las etapas forman parte de un plan de iteración para implantar el SGC, teniendo como objetivo guiar a los roles en la continuidad de las actividades y tareas a cumplir. Cada etapa presenta objetivos específicos y se definen actividades que permiten llevar a cabo de forma controlada y organizada las tareas.

Roles que intervienen en el desarrollo de la guía.

Roles definidos por el Sistema	Funcionalidad
Creador de Conocimientos.	Encuentra información fiable y eficaz acerca del tema en cuestión y busca la manera más eficiente de representar la información, logrando que esta sea mucho más accesible para el Receptor.
Desarrollador.	Encargado de elaborar el medio mediante el cual se va a mostrar la información.
Receptor.	Como usuario final debe nutrirse de los conocimientos que ponen a su disposición por ello su objetivo es aprender e instruirse de toda la información posible.

Tabla 2: Roles que intervienen en el desarrollo de la guía.

1.5.1 Descripción de la propuesta.

El concepto de Guía es un término amplio cuando a la definición de la palabra se hace referencia. Es la representación basada en indicar, enseñar, conducir, dirigir, orientar, aconsejar, encaminar con el objetivo de listar u organizar datos o informaciones referentes a determinada materia. Es una herramienta analítica que tiene como fin facilitar información sobre un sector o actividad concreta. Sirve de apoyo para la realización de cualquier trabajo que se desee desarrollar, siempre y cuando tenga existencia.

Ventajas que ocasiona el trabajo con una guía.

Las Guías son una valiosa herramienta, que pueden facilitar el aprendizaje ayudando a organizar el tiempo y a darse cuenta de qué falta, tanto en las tareas cómo en las actividades prácticas que se deben realizar, enseña a encontrar lo esencial de cada tema.

Las ventajas fundamentales que presenta se resumen en:

- ❖ Facilitan planear las actividades de manera secuencial, evitando improvisar.
- ❖ Gestiona todo tipo de documentación.
- ❖ Proporciona al candidato una perspectiva real de su posición y de sus posibilidades en la fase de trabajo en la que se encuentra trabajando.
- ❖ Dependiendo de la colaboración de la Web se podrán encontrar actualizaciones, ampliaciones, y enlaces, de bibliografía sobre la materia.

Hoy en día el principal problema que se encuentra al inicio de los proyectos, es la falta de información para la puesta en marcha de la idea de negocio. Esta carencia es uno de los principales motivos de fracaso de las nuevas empresas. Con el objetivo de aportar conocimientos y estudios realizados, se piensa la propuesta de esta Guía, con el fin de ayudar a aquellos proyectos que desarrollen aplicaciones Web, a llevar a cabo el proceso de aseguramiento de la calidad organizado y así garantizar la calidad de sus productos.

Se presenta una Guía razonable y cuidadosamente desarrollada, que procura ser útil sobre todo a quienes están dando sus primeros pasos en el mundo del aseguramiento de la calidad del software. Contiene la información necesaria y suficiente para que personas sin grandes conocimientos en la materia, puedan desarrollar en un corto periodo de tiempo la funcionalidad requerida en esa área de trabajo.

El cuerpo central de la Guía estará integrado por un conjunto de actividades, con el propósito de ayudar a novatos, profesionales y a clientes a tomar decisiones en el área del aseguramiento de la calidad del software. La Guía, de forma general, basa su desarrollo en concentrar y estructurar como teoría propia todo lo necesario para asegurar que el producto software sea elaborado con la calidad requerida.

1.6 Conclusiones parciales.

- El análisis del estado del arte permitió definir al modelo de calidad CMMI, el área de proceso de VER y VAL del nivel tres, como el más apropiado para ser empleado en la realización de la guía para el aseguramiento de la calidad, debido a que este no solo busca el éxito en un proyecto en particular, sino que sea mantenido para todos los demás de la organización, con costos cada vez menores, permitiendo aumentar las ganancias de la institución, y debido al procesos de mejora que se lleva a cabo en la universidad para el alcance en el 2010 de una certificación internacional del nivel 2 del modelo CMMI.
- Se determinó la utilización del modelo WQM basado en la norma ISO/IEC 12 207 para la definición del ciclo de vida de una aplicación.
- Se estudiaron los factores relacionados con la calidad de software escogiendo como más apropiados los factores propuestos por Olsina, lo que permitió conocer las características que debía tener una aplicación Web para que tuviera niveles adecuados de calidad.

CAPÍTULO II: Guía de aseguramiento de la calidad para aplicaciones web.

En el presente capítulo se efectúa la propuesta de solución para la guía de aseguramiento de la calidad de software en proyectos que desarrollan aplicaciones Web. Se parte de un análisis previo realizado en el capítulo anterior correspondiente a los diferentes términos seleccionados. Primeramente se realiza una descripción de la propuesta que se plantea para la universidad. Se selecciona un ciclo de vida apropiado para las aplicaciones Web que se desarrollan. A partir del ciclo de vida selecto, se definen las actividades y tareas enfocadas específicamente al producto, además de los artefactos, entregables o productos que se van obteniendo por actividad. Se eligen los estándares, normas y modelos de calidad que se puedan emplear. Se elabora esta propuesta analizando de igual forma, los lineamientos plateados por la Dirección de Calidad de la Infraestructura Productiva, el plan de aseguramiento de la calidad y otros definidos dentro de los proyectos que desarrollan aplicaciones Web.

2.1 Descripción de la propuesta para la universidad.

Con el objetivo de garantizar la calidad de los proyectos que desarrollen aplicaciones Web en todas sus etapas y acorde con los objetivos establecidos dentro de la institución, se propone elaborar un guía (Ver Anexo 7) en la cual se establezca el propósito, alcance, definiciones y aspectos significativos como el ciclo de vida y las fases del software, que normen la acción de asegurar la calidad de las aplicaciones Web.

La Guía es básicamente un documento corto que explica un proceso u operación resuelta. La idea de este artículo es explicar brevemente los pasos necesarios que debe realizar el asegurador de la calidad del software para asegurar la calidad en el proyecto que se encuentre desempeñando su labor, por lo que está dirigida directamente a él o cualquier otro rol autorizado de llevar a cabo esta tarea.

Las dificultades afrontadas hoy en los proyectos, la poca experiencia y el poco manejo que tienen en el tema los encargados de la calidad del software, constituyen la fuente de esta guía. Por ello su objetivo principal es brindar pasos, técnicas y recomendaciones para asegurar la calidad de software en los proyectos que desarrollan aplicaciones Web, que ayuden al

asegurador de la calidad a organizar el trabajo y eliminar las deficiencias existentes para lograr un producto con la calidad requerida.

Además de aconsejar y proponer buenas prácticas que ayuden al trabajo sistemático del asegurador. Esta se basa en casos de estudio para explicar en detalles los pasos necesarios que se deben tener en cuenta.

¿Cómo está estructurada?

La estructura de la guía está constituida como se muestran en la figura:

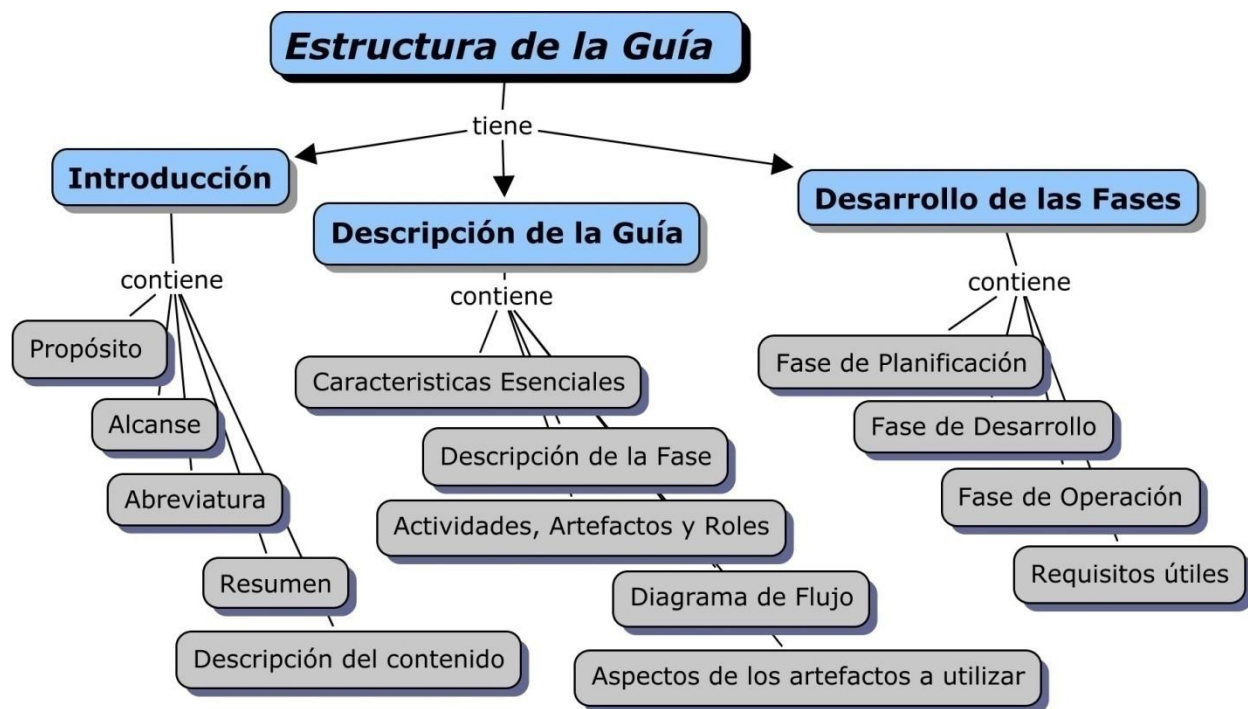


Figura 10: Estructura de la Guía.

Se deberán seguir estas técnicas de aseguramiento de calidad desarrolladas en organizaciones, modelos, y planes reconocidos dentro de la calidad del software. A continuación se especifican:

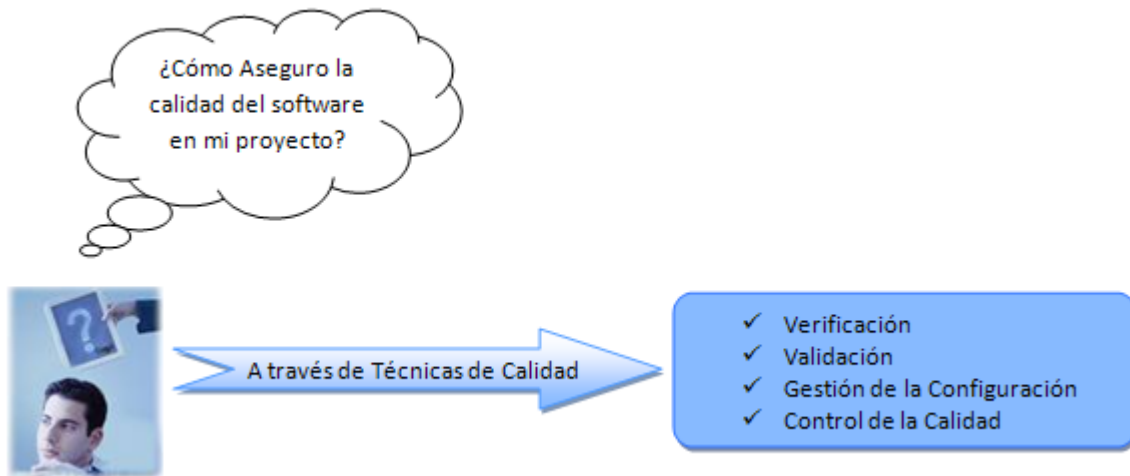


Figura 11: Técnicas de Aseguramiento de la Calidad

A través de cada una de estas técnicas ya descritas en el apartado (1.4) se realiza el proceso de aseguramiento de la calidad del software para cada artefacto asociado a las actividades definidas dentro de los procesos que representan al ciclo de vida determinado.

Para ello se basa en normas, modelos, estándares y libros. Ellos son los encargados de brindar datos y pasos asociados correspondientes a las aplicaciones Web. Ejemplo:

- ❖ WQM: define Ciclo de vida de las aplicaciones Web.
- ❖ CMMI: define el Área de procesos VER y VAL.
- ❖ ISO 12207: define procesos para el aseguramiento de la calidad del producto.
- ❖ R. Pressman: define actividades para el aseguramiento de la calidad del producto.
- ❖ McCall: define factores de calidad para aplicaciones Web.

La figura siguiente resume los aspectos esenciales de la propuesta:

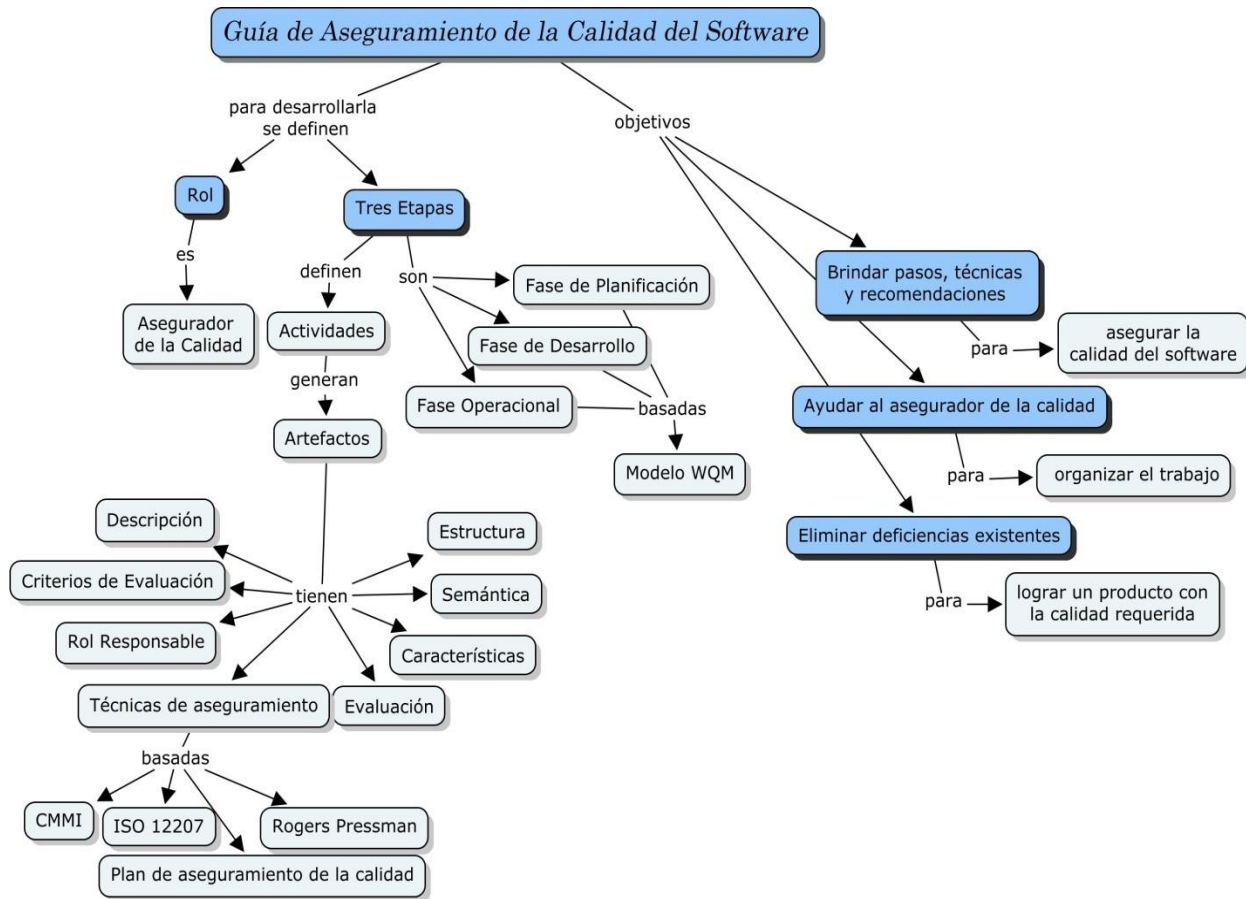


Figura 12: Propuesta de Solución (Ver Anexo 7).

2.2 Análisis del ciclo de vida de la propuesta.

El modelo empleado en el presente trabajo para definir el ciclo de vida es el conocido *Model Quality Web (WQM)*, descrito en el epígrafe (1.3.4) correspondiente al Capítulo I.

El modelo propone tres procesos específicos. (Ver Figura 6) pero solo los dos primeros (Proceso de Desarrollo, Proceso de Operación) se utilizan en la guía, ya que las actividades que se desarrollan dentro del proceso de mantenimiento apenas son realizadas en la universidad por lo que no se cree importante su inclusión.

Los procesos seleccionados se describen como sigue a continuación:

Proceso de Desarrollo: El proceso contiene las actividades para el análisis de los requisitos, diseño, codificación, integración, prueba e instalación y aceptación relacionadas con los productos de software. Puede contener actividades relacionadas con el sistema si está estipulado en el contrato.

Proceso de Operación: El proceso comprende la operación del producto de software y el soporte operacional a los usuarios. Las actividades y tareas que contiene se basan en las pruebas, apoyo al usuario y liberación del producto.

Los procesos están compuestos por tres fases de desarrollo, con el objetivo de organizar las actividades y tareas correspondientes a estos.



Figura 13: Ciclo de Vida de Aplicaciones Web.

Actividades asignadas a cada fase:



Figura 14: Actividades asignadas a cada fase.

Las actividades que se realizan dentro del ciclo de vida del software generan un grupo de artefactos que se mencionan a continuación:



Actividades	Artefactos
Implementación del proceso.	Plan de proyecto. Plan de Aseguramiento de la Calidad. Plan de mitigación de riesgos. Visión del Proyecto.
Análisis de los requisitos del software.	Especificación de requisitos. Plan de Gestión de Requisitos.
Diseño de la arquitectura del software.	Arquitectura de software. Arquitectura de información. Diagrama de iteración.

Diseño detallado del software.	Modelo de Diseño. Modelos de Diseño Web. Documento de Análisis. Diagrama de Despliegue.
Codificación y prueba del software.	Modelo de implementación. Plan de pruebas. Manual de Usuario. Entregables.
Integración del software.	Plan de integración. Entregables.
Prueba de calificación del software.	Pruebas basados en Requisitos.
Instalación del software.	Plan de Instalación.
Apoyo a la aceptación del software.	Plan de Capacitación.
Prueba Operacional.	Plan de prueba.
Operación del Sistema.	Entregables.
Apoyo al Usuario.	Producto.

Tabla 3: Actividades y Artefactos de las fases del ciclo de vida

Se definen cinco roles dentro del ciclo de vida seleccionado, con el objetivo de lograr organizar el trabajo en un equipo de desarrollo de software.

Roles Definidos

-  Jefe de Proyecto.
-  Planificador.

- ✚ Desarrollador.
- ✚ Diseñador.
- ✚ Analista.
- ✚ Probador.
- ✚ Asegurador de la Calidad.

A los roles se les asigna cada una de las actividades en correspondencia de las tareas específicas que deban desarrollar:

Roles	Actividades
Jefe de Proyecto. Planificador.	Implementación del Proceso.
Analista.	Análisis de los Requisitos del Software.
Diseñador.	Diseño de la arquitectura del Software. Diseño Detallado del Software.
Desarrollador.	Codificación y Prueba del software. Integración del Software.
Probador.	Prueba de Calificación del Software. Instalación del Software. Apoyo a la Aceptación del Software. Prueba Operacional. Operación del sistema.

Tabla 4: Roles y Actividades de las fases del ciclo de vida

El Asegurador de la calidad de software es el único rol que no tiene asignada actividades, pues él será el encargado de asegurar que el producto software salga con la calidad requerida. Para ello realiza diferentes acciones que debe desarrollar en el transcurso del ciclo de vida de la aplicación Web y que se plasman en el documento como parte de la propuesta solución al problema existente hoy en los proyectos de la universidad. Las actividades que realiza el asegurador son muy similares a las del probador ya que ambos realizan sus tareas en base a proporcionar la confianza de que el producto satisfaga los requisitos del cliente, pero con la diferencia de que el asegurador trabaja desde que se inicia el ciclo de desarrollo del software. Su trabajo se alimenta de la planificación, verificación, validación, control, inspección entre otros. Algunos proyectos solo cuentan con la presencia de probadores. Ellos son los responsables de asegurar la calidad en, dichos proyectos a través de pruebas que le realizan al software. Pero, ¿Realmente aseguran la calidad del software?

La respuesta a esta pregunta se realiza a medida que sea leída e interiorizada la guía de aseguramiento de la calidad del software que se propone.

2.3 Desarrollo de las fases

2.3.1 Estructura de las Fases.

Las fases que se mencionan en la figura 14 se estructuran de la siguiente manera:

- ✚ Descripción de la fase.
- ✚ Objetivos específicos.
- ✚ Actividades correspondientes.
- ✚ Artefactos.
- ✚ Preguntas de control.
- ✚ Buenas Prácticas.
- ✚ Resumen de la fase.

Descripción de la fase: Se hace una breve descripción de las funcionalidades y tareas de la fase correspondiente.

Objetivo específico: Se dan a conocer los objetivos específicos que presenta la fase que se esté desarrollando.

Actividades correspondientes: Se mencionan todas las actividades que se desarrollan dentro de la fase correspondiente.

Artefactos: Se describen y se mencionan los artefactos asociados a cada actividad. Además de puntualizar aspectos esenciales (descripción del artefacto, criterios de evaluación, técnicas de aseguramiento, entre otros)

Preguntas de Control: Se realizan varias preguntas con el fin de consolidar lo aprendido en cada fase.

Buenas Prácticas: Se mencionan un conjunto de prácticas que le servirán de ayuda al asegurador de la calidad del software.

Resumen de la fase: Se acumula los aspectos más importantes de la fase.

2.3.2 Aspectos más importantes a desarrollar dentro de las fases.

Dentro de cada una de una de las fases los aspectos que toman más importancia para el aseguramiento de la calidad en los proyectos o equipos de trabajo son:



Figura 15: Aspectos más importantes dentro de las fases

Cada una de las actividades genera un conjunto de artefactos a los cuales se le realizara un proceso de aseguramiento de calidad para que estos sean concebidos con la calidad que se requiere.

Para ello se tiene en cuenta un conjunto de aspectos mencionados a continuación:

- ❖ Estructura del documento.
- ❖ Semántica del documento.
- ❖ Descripción del artefacto.

- ❖ Criterios de evaluación.
- ❖ Descripción de las Técnicas de aseguramiento.
- ❖ Evaluación del artefacto.

Estructura del documento: Este debe estar acorde con la plantilla estándar del proyecto o del expediente de proyecto y contener cada una de las secciones obligatorias definidas en el expediente (Ver expediente de proyecto). Se debe exigir que se cumpla con estas especificaciones ya que forma parte de la organización y seriedad con que se trabaja en el proyecto.

Semántica del documento: La semántica del documento tiene en cuenta los siguientes puntos.

- Identificar errores ortográficos.
- Entender claramente lo que se ha especificado en el documento.
- El número de página que aparece en el índice debe coincidir con el contenido que se refleja realmente en dicha página.
- El total de páginas que aparecen en las reglas de confidencialidad debe coincidir con el total de páginas que tiene el documento.

Descripción del artefacto: Los artefactos que necesitan ser evaluados deben ser descritos correctamente para su posterior uso.

Se clasifica en tres aspectos:

- Descripción correcta del artefacto.
 - Si se detalla y se explica el artefacto.
- Descripción incorrecta del artefacto.
 - Si no se detalla el artefacto.

Criterios de evaluación: Cada artefacto presenta un cumulo de criterios de evaluación.

Para realizarle la evaluación a cada artefacto y sugerir técnicas de aseguramiento será necesario que los criterios de evaluación estén clasificados:

Se clasifican en dos aspectos:

- Bien
 - Si son realizados y descritos correctamente.
- Mal
 - Si están especificado incorrectamente.

Descripción de las Técnicas de aseguramiento: Las técnicas de aseguramiento se basan en los siguientes aspectos:

- Verificación y Validación.
- Gestión de la Configuración (Control de Cambio).
- Control de la Calidad.

Se describen las actividades que realiza el asegurador para garantizar la calidad del software. Para ello el asegurador lleva a cabo el siguiente diagrama de flujo. Este diagrama es la representación grafica de la secuencia de pasos que se realizan para obtener un cierto resultado del producto.

:

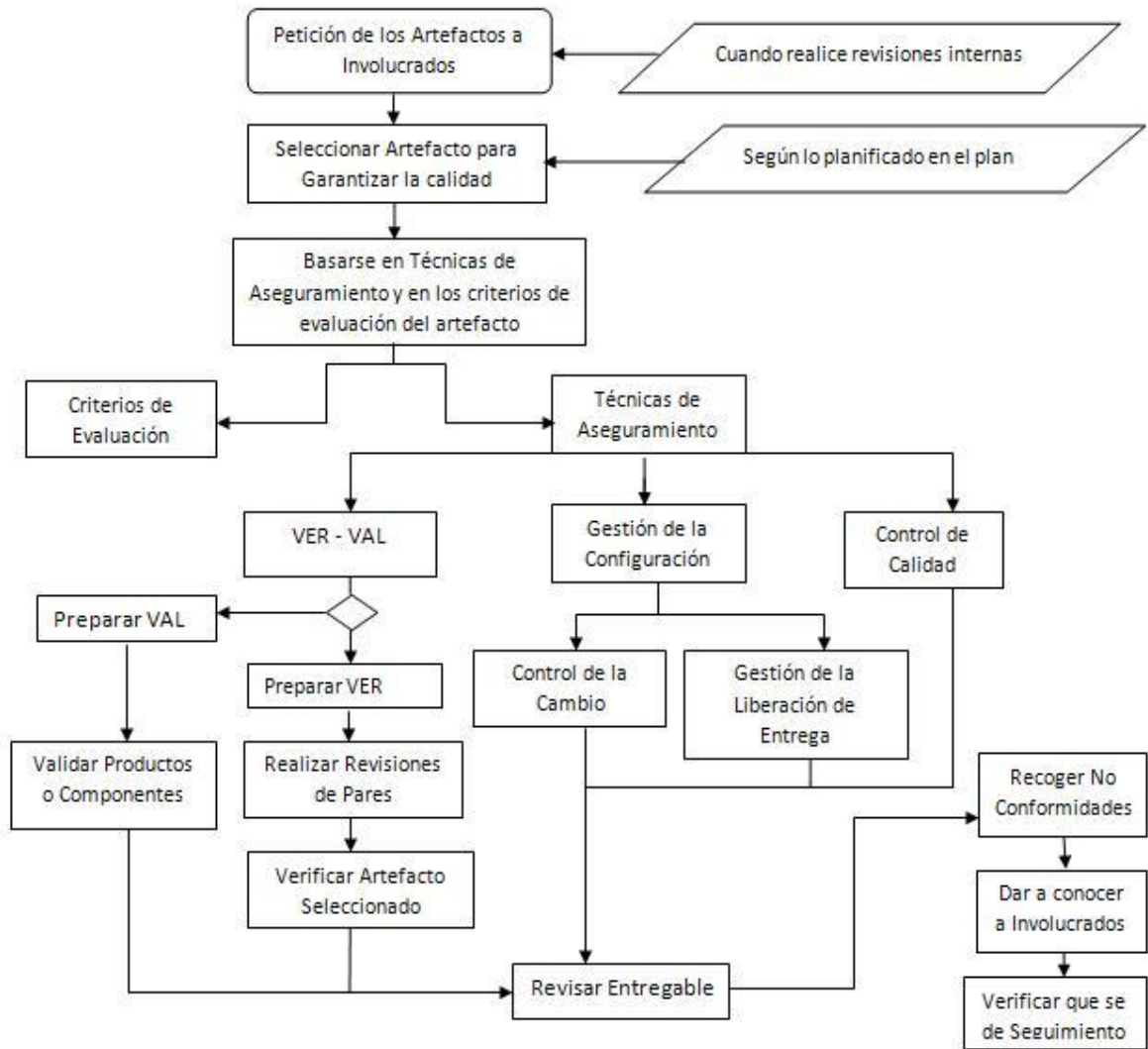


Figura 16: Diagrama de flujo de las actividades del asegurador de la calidad

Para cada entregable o versión del producto debe tener en cuenta la siguiente secuencia de pasos:



Figura 17: Diagrama de flujo para evaluar el artefacto.

El asegurador dentro de las técnicas de aseguramiento que emplea está también autorizado a participar en la solicitud de cambio que se realiza internamente en el proyecto. La figura muestra los pasos específicos que debe realizar.

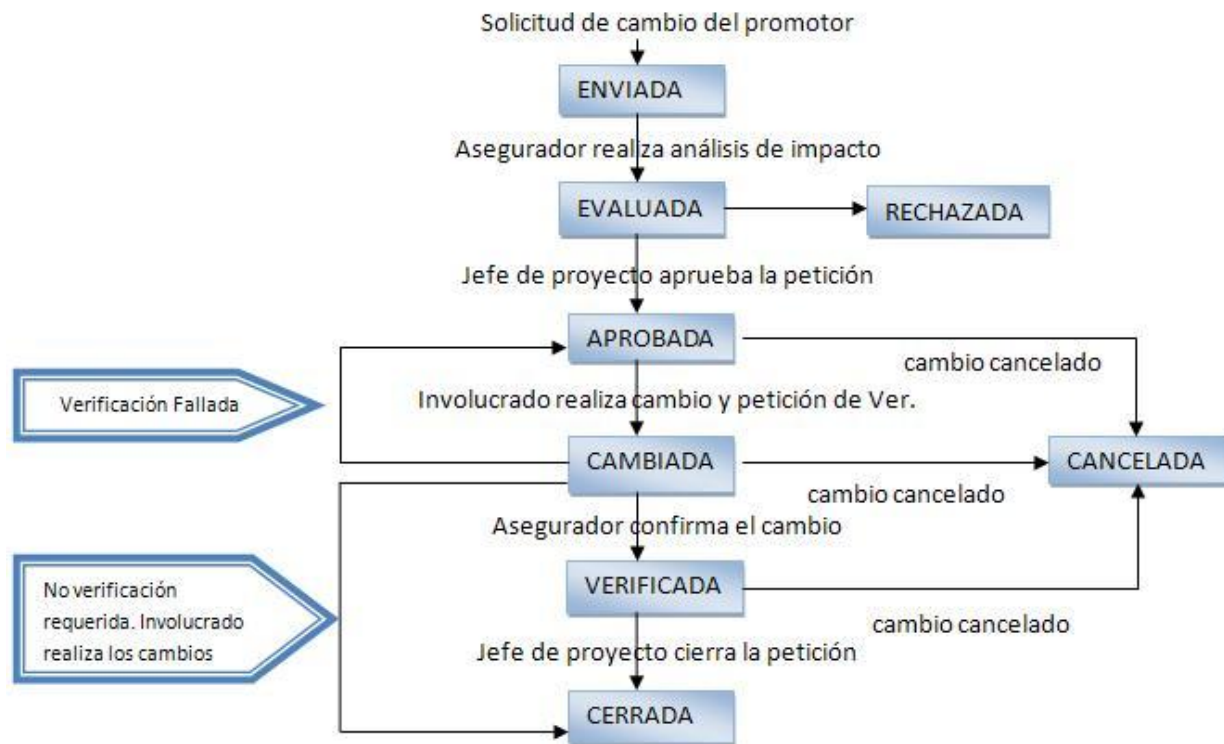


Figura 18: Posibles resultados de una petición de cambio

Evaluación del artefacto: Los artefactos como parte del proceso de aseguramiento de la calidad deben ser evaluados.

Cada uno de los artefactos que necesiten ser evaluados se rige por la siguiente descripción

Se aborta la revisión del artefacto revisado si:

- El promedio de las No Conformidades críticas por elementos es superior a uno.

Este criterio es empleado únicamente para los elementos. En caso de otro artefacto la revisión se aborta si:

- Existen al menos dos indicadores críticos evaluados de mal.
- Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.

- Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.
- Se mantienen las No Conformidades de una revisión a otra.

Se evalúa de regular la calidad del artefacto revisado si el artefacto no cumple los criterios para ser abortado y:

- Existe una No Conformidad crítica.
- La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.

Estos criterios se cumplen para todas las secciones que tiene la lista de chequeo.

El artefacto es evaluado de bien si no cumple ninguno de los criterios anteriores y:

- No existe ninguna No Conformidad relacionada con indicadores con peso crítico.
- Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.

Estos aspectos se tendrán en cuenta para asegurar la calidad del producto y de los entregables que se vayan generando en todo el ciclo de vida del software. Para ello se hace uso del registro del asegurador de la calidad, donde debe quedar plasmada las actividades que realiza. (Ver anexo 6).

2.3.3 Fase Inicial: Planificación.

Descripción de la Fase:

Es la fase inicial del proyecto donde se planifican de forma cuidadosa y detallada cada una de las actividades que se deben desarrollar en el transcurso de la elaboración de producto software. Se realizan estimaciones del trabajo a realizar, los recursos necesarios y el tiempo de duración desde el inicio hasta el final del software. Se analizan los requisitos y se diseñan la arquitectura del software.

Objetivo Específico:

El objetivo de esta fase es proporcionar al asegurador de la calidad del software, aspectos, herramientas, descripciones y pasos característicos que deberá realizar el rol con el fin de lograr junto al equipo de desarrollo la calidad que requiere el producto.

Actividades realizadas en esta fase:

- ❖ Implementación del proceso.
- ❖ Análisis de los requisitos del software.
- ❖ Diseño de la arquitectura del software.
- ❖ Diseño detallado del software.

Implementación del proceso: Se debe definir o seleccionar un modelo para el ciclo de vida del software apropiado para el alcance, magnitud y complejidad del proyecto. Se selecciona y se ajustan las normas, métodos, herramientas y lenguaje de programación que estén documentados, son apropiados y están establecidos por la organización para ejecutar las actividades definidas en los procesos (se realiza todo esto si no está estipulado en el contrato). Se elaboran planes para la conducción de las actividades. Los planes deben incluir normas, métodos, herramientas, acciones específicas y la responsabilidad asociada con el desarrollo y la calificación de todos los requisitos, incluyendo la protección y la seguridad. Los planes deberán ser documentados y ejecutados.

Análisis de los requisitos del software: Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge un escrito llamado SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta etapa se deben consensar todo lo que se requiere del producto.

Diseño de la arquitectura del software: Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Diseño detallado del software: En esta actividad el diseñador deberá realizar un diseño detallado de cada componente de software del artículo de software. Refinará los componentes hasta niveles más bajos conteniendo las unidades de software que pueden ser codificadas,

compiladas y probadas. Deberá asegurar que todos los requisitos de software son asignados a partir de los componentes de software. Es donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación. El diseño detallado debe ser documentado.

Actividades y Artefactos asociados a cada actividad:

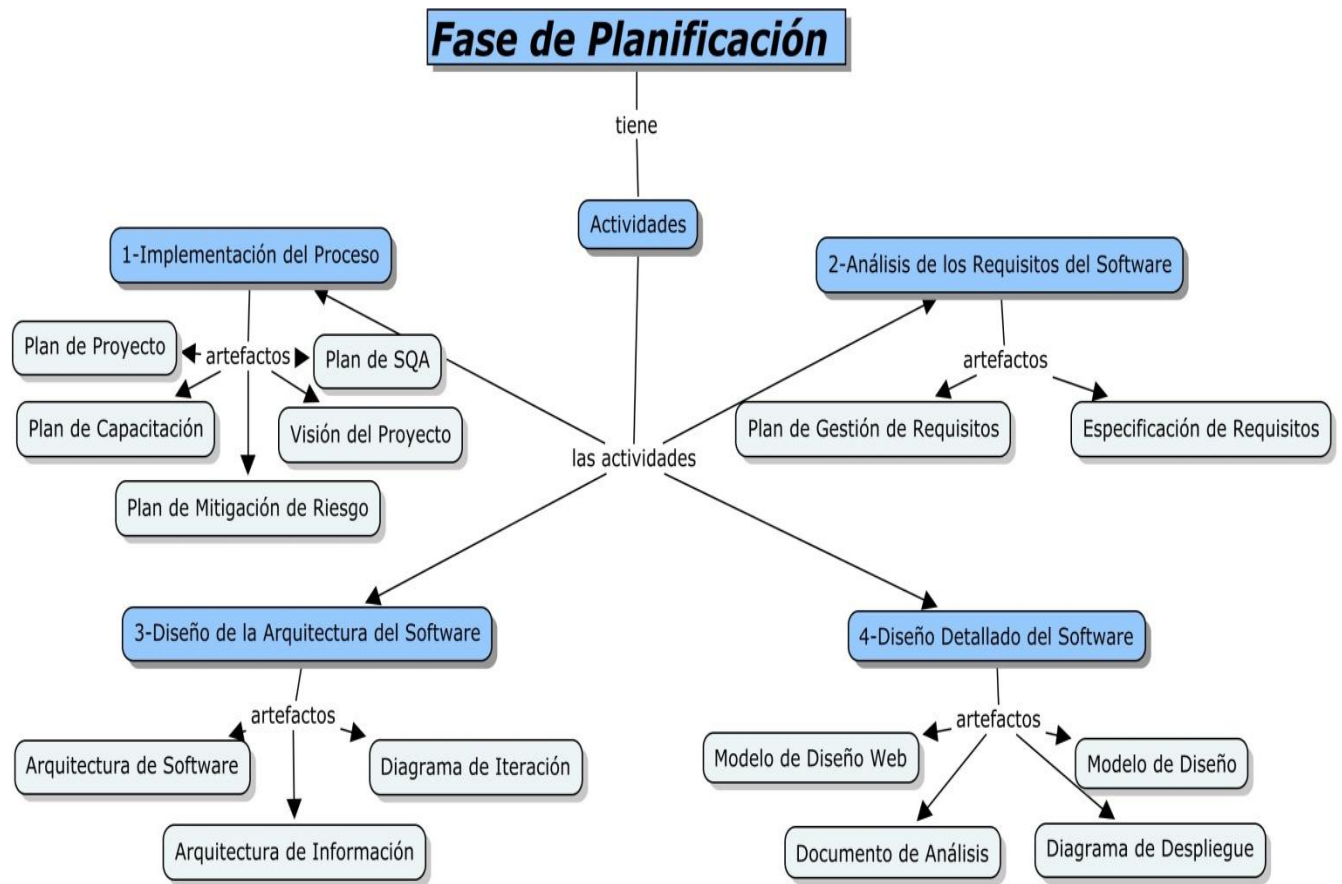


Figura19: Fase de Planificación. Actividades y Artefactos asociados

Descripción de los Artefactos:

- ✚ Plan de aseguramiento de la Calidad: Esfuerzo total para plantear, organizar, dirigir y controlar la calidad en un sistema de producción con el objetivo de dar al cliente productos con la calidad adecuada.
- ✚ Plan de Proyecto: El plan de proyecto es una herramienta de planificación para el equipo y los patrocinadores del proyecto.
- ✚ Plan de Mitigación de Riesgos: Es una lista de las acciones específicas que son llevadas el reparto con riesgos específicos.
- ✚ Visión de Proyecto: La visión del proyecto también conocida como Caso de Desarrollo, es la visión de los modos, hitos y tareas que se ejecutarán durante el proyecto.
- ✚ Especificación de requisitos: Es una descripción completa del comportamiento del sistema que se va a desarrollar.
- ✚ Plan de Gestión de Requisitos: Es un componente vital en el desarrollo de un proyecto software ya que provee la dirección y alcance del proyecto.
- ✚ Arquitectura de software: Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.
- ✚ Arquitectura de Información: Es la disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos.
- ✚ Modelo de Diseño: El Modelo de Diseño es la única manera de materializar con precisión los requerimientos del cliente.
- ✚ Documento de Análisis: Método, plan o procedimiento de clasificación para hacer algo.

Técnicas de Aseguramiento de la Calidad:

El asegurador de la Calidad del Software es el encargado de confeccionar el plan de aseguramiento de la calidad del proyecto. En caso de este plan ser entregado por alguna organización o empresa, debe adaptarlo a las necesidades del equipo de trabajo al que pertenece. Esta, es una de las actividades primordiales que se realiza dentro del equipo ya que este plan le sirve de apoyo para planificar las actividades que implican las revisiones, verificaciones y validaciones de los documentos, entregables y productos correspondientes al trabajo realizado por los implicados. Cada una de las actividades realizadas en esta fase genera un artefacto, los cuales son mencionados en la figura 18.

Para asegurar la calidad de estos, el asegurador de la calidad debe realizar técnicas de aseguramiento. Tales son mencionadas en la figura 14.

En esta fase el asegurador deberá realizar una inspección o auditora interna para revisar cada uno de los documentos enfocados al producto. En los cuales deberá hacer tanto para uno como para otro, los siguientes pasos característicos:

- ✓ Solicitar el documento al involucrado.
- ✓ Verificar que cumple con criterios de evaluación especificados en el proyecto.
- ✓ En caso de no tenerlos, recoger en un informe las no conformidades.
- ✓ Dárselas a conocer al involucrado y al jefe de proyecto.
- ✓ Dar un tiempo límite de entrega al involucrado para que corrija los errores encontrados.
- ✓ Verificar que cumpla con la fecha señalada.
- ✓ Verificar los cambios.
- ✓ En caso de estar completado, darle seguimiento.
- ✓ Documentar todo e informar los resultados al jefe del proyecto.

Resumen de la Fase:

Esta es una de las fases principales dentro del ciclo de vida del software ya que forma parte del fundamento básico que tendrá el producto. Sin una buena planificación se cometerán errores en todo el transcurso del software, pero además de eso todas las actividades que se realicen después de ello estarán expensas a fallos. Es donde se recogen los requisitos del usuario con el fin de satisfacer sus expectativas y necesidades, los cuales son diseñados y modelados para una posterior implementación. El asegurador de la calidad debe tener en orden y actualizado la documentación y entregables del equipo de desarrollo al cual él tiene acceso, para que el proyecto esté funcionando de acuerdo a lo planificado.

2.3.4 Fase Intermedia: Desarrollo.

Descripción de la Fase:

Durante esta fase se realizan las actividades y tareas para el análisis de la codificación, integración, prueba e instalación y aceptación relacionadas con los productos de software. Se ejecutan las actividades de esta fase de acuerdo con el contrato y las propuestas dentro del equipo de desarrollo. Se describen los artefactos asociados a las actividades y se le da

cumplimiento en dependencia de las tareas que se hayan planificado en los planes correspondientes.

Es en esta fase donde el asegurador de la calidad de software aparte de asegurarle la calidad a los diferentes artefactos asociados al producto.

Objetivo Específico:

Esta fase tiene como objetivo específico integrar al asegurador de la calidad a las actividades que realicen otros roles, como por ejemplo, auxiliará de cierta forma al probador cuando se realicen las pruebas planificadas.

Actividades realizadas en esta Fase:

- ❖ Codificación y prueba del software.
- ❖ Integración del software.
- ❖ Prueba de calificación del software.
- ❖ Instalación del software.
- ❖ Apoyo a la aceptación del software.

Codificación y prueba del software: Se implemente y se realizan pruebas planificadas y sistemáticas.

Integración del software: Se integran las unidades de software y los componentes de software y se prueban a medida que estos agregados son desarrollados de acuerdo con el plan de integración.

Prueba de calificación del software: Para la calificación del software se realizan dos pruebas principales:

- ✓ *Pruebas Unitarias:* Consisten en probar o testear piezas de software elementales. Se utilizan para asegurar el correcto funcionamiento de secciones de código reducidas.

- ✓ **Pruebas de Integración:** Se realizan una vez concluidas las pruebas unitarias. Con ellas se intenta asegurar que el sistema o los subsistemas que componen las piezas individuales del software funcionen correctamente al operar en conjunto.

Instalación del software: Es el proceso por el cual las aplicaciones desarrolladas son transferidas apropiadamente a su destino, inicializadas, y, eventualmente configuradas con el propósito de ser utilizado por el usuario final o cliente.

Apoyo a la aceptación del software: Es donde se apoya la revisión y prueba de aceptación del adquiridor o comprador al producto software, los cuales deben ser documentados.

Artefactos asociados a cada Actividad:

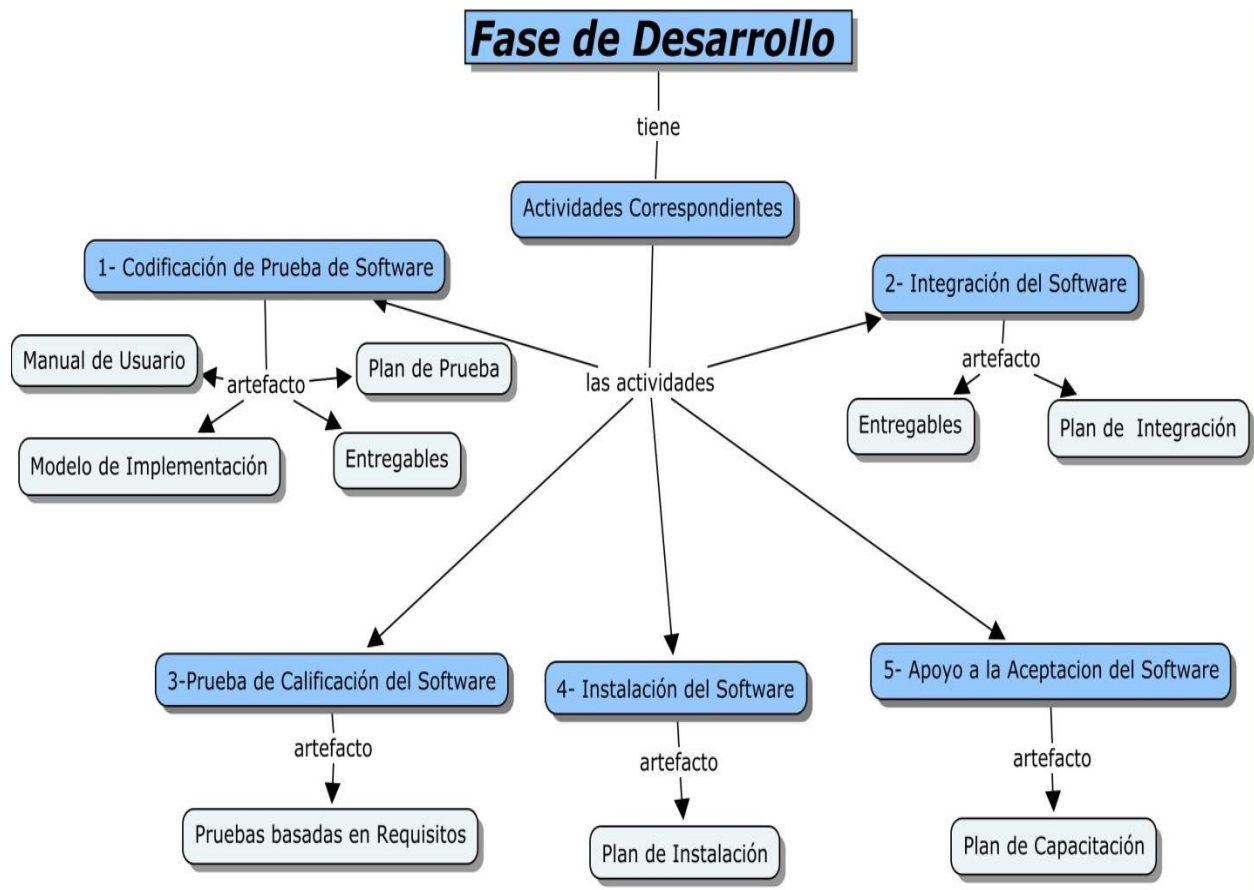


Figura 20: Fase de Desarrollo. Actividades y Artefactos asociados

Descripción de los Artefactos:

- ✚ Modelo de implementación: El modelo de implementación describe cómo los elementos de diseño se implementan en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.
- ✚ Plan de Prueba: Es un documento donde se plasman las pruebas que se deben realizar en el trascurso del software.
- ✚ Entregables: Versión del producto que se está desarrollando.
- ✚ Plan de Integración: Este documento muestra un plan detallado de la integración dentro de una iteración. El propósito de este plan es definir el orden en que los componentes del sistema deben llevarse a cabo, los resultados al integrar el sistema y cómo serán evaluados.
- ✚ Pruebas basadas en requisitos: Intenta llevar a cabo pruebas basadas directamente en la especificación de requisitos.
- ✚ Plan de Instalación: Es un documento donde se encuentra planificado las instalaciones que se deben desarrollar.
- ✚ Plan de Capacitación: Se disminuye la tasa de rotación de personal, y permite entrenar sustitutos que puedan ocupar nuevas funciones rápida y eficazmente.

Técnicas de Aseguramiento:

En esta fase el Asegurador de la calidad verifica los documentos, realizando los mismos pasos especificados en la fase anterior. Además de inspeccionar la entrega de estos documentos, le da asistencia al probador y al jefe de proyecto, en la realización de las pruebas planificadas y en la capacitación al personal correspondientemente. Para ello también hace uso de las técnicas de aseguramiento a través de estas actividades:

- ✓ Verifica que se realicen las pruebas en fecha establecida.
- ✓ Verifica que se documenten cada resultado.
- ✓ Lleva el control de todas las pruebas que son realizadas.
- ✓ Lleva el control de todos los cambios que puede sufrir el producto.
- ✓ Lleva el control del personal que recibe capacitación.
- ✓ Debe verificar nuevamente en esta fase que se haga un buen uso de los estándares, normas, modelos y metodologías especificadas dentro del proyecto.

Resumen de la Fase:

La fase de desarrollo es iniciada con la codificación del software, donde se realizan pruebas sistemáticas y planificadas tanto en el expediente de proyecto, plan de aseguramiento de la calidad como en el plan de prueba. Además son documentadas. Posterior a los resultados se realiza la integración del producto, se ejecutan las pruebas de calificación del software y se instala. Una vez realizada exitosamente la instalación del software se efectúa el apoyo a la aceptación del software. En fin se ejecutan y se apoyan un conjunto de actividades relacionadas entre sí, en dependencia a lo estipulado en el contrato.

2.3.5 Fase Final: Operación.

Descripción de la Fase:

La fase comprende la operación del producto de software y el soporte operacional a los usuarios. Es la fase más corta dentro del ciclo de vida. Contiene las actividades y tareas del probador, desarrollador y por supuesto del asegurador de la calidad. Se les proporciona asistencia a los usuarios. Las actividades y tareas de esta fase se aplican al software completo. El software debe ser operado de acuerdo con la documentación de usuario en su entorno previsto.

Objetivo Específico:

El objetivo específico de esta fase está comprendido al igual que la fase de desarrollo, respecto a integrar al asegurador de la calidad a las actividades que realicen otros roles. Además de mostrar que el producto gracias al trabajo realizado cumple las funcionalidades para las que fue desarrollado, es decir se produce el producto esperado por el usuario.

Actividades realizadas en esta Fase:

- ❖ Prueba Operacional.
- ❖ Operación del Sistema.
- ❖ Apoyo al Usuario.

Prueba Operacional: Se realiza con el fin de verificar si se satisfacen los criterios especificados, en caso de que se cumplan se libera el producto de software para su uso operacional.

Operación del Sistema: En esta actividad se le realizan las últimas operaciones al software asegurando que se tengan en cuenta los factores de calidad.

Apoyo al Usuario: Se le facilita ayuda al usuario cuando este lo solicite.

Artefactos asociados a cada Actividad:



Figura 21: Fase de Operación. Actividades y artefactos asociados

Descripción de los Artefactos:

- ✚ Plan de Prueba: Es un proceso usado para identificar posibles fallos.
- ✚ Entregable: Es una versión del producto que consiste en una colección de componentes ejecutable, documentos (material de soporte para usuarios, notas de la versión) y elementos de instalación.
- ✚ Producto: Es el producto aprobado y liberado por producción, preservando siempre su calidad.

Técnicas de Aseguramiento:

En esta fase final el asegurador de la calidad realiza un cúmulo de actividades relacionadas con las pruebas, operaciones y apoyo al producto ya elaborado. Por ello tiene que estar al tanto de todos los detalles y cambios que se realicen. La pruebas operacionales que son realizadas en la fase serán documentadas, igualmente las verificaciones y mediciones que se le realizan al producto para comprobar su usabilidad, funcionalidad fiabilidad y eficiencia por lo que el

asegurador debe realizar las mismas actividades definidas en la primera fase, con respecto a las actividades que debe realizar para verificar los artefactos documentados.

Esta fase resume el trabajo realizado durante todo el ciclo de vida del software, es decir la elaboración total del producto final, donde lograr que este salga con la calidad que se requiere es responsabilidad de todos los involucrados, pero con la máxima dirección del asegurador de la calidad y jefe de proyecto.

El asegurador debe llevar el control específicamente de que se cumpla con lo que está establecido, en tiempo y forma, lo que constituye parte de todas las actividades que debe realizar durante el ciclo de vida del software. Todas las actividades descritas y otras que se mencionan a continuación conllevan a que la calidad del producto software sea asegurada en los proyectos o grupos de trabajo con el fin de satisfacer los requisitos del cliente y elevar la popularidad de la empresa.

Otras Actividades del asegurador:

- ✓ Debe verificar que el producto cumpla con todos los factores que haya exigido el cliente o que se haya decidido en el equipo de trabajo.
- ✓ Debe darle todo el tiempo posible de seguridad del producto elaborado al cliente, a través de consejos y recomendaciones.
- ✓ Debe garantizar el cumplimiento de la normativa y estándares establecidos.
- ✓ Debe documentar cambios imprevistos que sean realizados.

Resumen de la Fase:

En la fase de operación o fase final se parte de las pruebas operacionales que se le realizan al software. Esta actividad consta de la realización de la liberación de producto de software para su uso operacional, siempre que se satisfagan los criterios que se hayan especificados. Se asegura que las tareas sean ejecutadas y terminadas tal como está descrito en el plan. Se realizan actividades para facilitar ayuda y asesoría a los usuarios cuando son solicitadas. Estas se registran y se controlan al igual que las acciones subsiguientes. Son monitoreadas todas las soluciones hasta su conclusión. Se realizan correcciones y mejoras aplicadas al producto en caso necesario.

2.4 Técnicas de uso y estudio de la Guía.

En el presente trabajo se detallan las principales funcionalidades en las diferentes fases del ciclo de vida de las aplicaciones Web que debe realizar el asegurador de la calidad del software. Desde la fase Inicial, atravesando todo el desarrollo del software, hasta la fase de operaciones donde se obtiene el producto final. El trabajo que desempeña el asegurador dentro del proyecto, se caracteriza como sistemático y continuo, por lo que se recomienda que el uso de la guía se desarrolle de igual forma.

A los interesados en su uso:

La guía puede ser adaptada a cualquier proyecto que desarrolle aplicaciones Web, independientemente de la metodología o modelo que utilicen.

Se debe usar desde el comienzo del software realizando cada una de las actividades planificadas en correspondencia de la fase en la que se encuentre. Se debe definir los planes, aspectos, recomendaciones, buenas prácticas y consejos sugeridos, para alimentar la mejora continua en los proyectos, provocando organización y calidad en lo que se realiza. Vale destacar que para los proyectos que solo realizan etapas específicas del software, también pueden hacer uso de esta guía. Su elaboración fue concebida para que fuese adaptable en proyectos, equipos de trabajo, investigativos o de innovación y desarrollo o a cualquier organización que desarrolle aplicaciones web.

La guía propuesta puede ser utilizada tanto para usuarios que comiencen a desarrollar el rol de asegurador de la calidad como para aquellos con un tanto de experiencia en el tema.

Debe ser aplicada haciendo uso del Registro del Asegurador de la calidad definido en la guía que recoge los aspectos que lo ayudaran a llevar el control para asegurar la calidad del producto software en los proyectos. El uso del registro dentro del proceso para asegurar la calidad del producto juega un papel fundamental ya que de cierta forma quedara plasmado todo el trabajo del asegurador en todo el ciclo de vida del software respecto a las verificaciones, inspecciones, solicitudes de cambio, revisiones tanto formales, como informales, recomendaciones, buenas prácticas empleadas entre otras actividades.

A los interesados en su estudio:

La guía sirve de material de estudio para el asegurador de la calidad o representante de la calidad dentro de un proyecto. Se debe estudiar por fases desde el comienzo de esta hasta su

finalización. Cada una de las fases propone lineamientos de calidad los cuales para una mejor comprensión de estos se debe acudir al sitio de calidad central de la universidad calisoft.prod.uci.cu. En otro apartado se definen un conjunto de ejercicios que deber ser resueltos al finalizar el estudio de cada fase. Se definen un conjunto de buenas prácticas que deberá interiorizar y poner en práctica. Por último se dan un conjunto de requisitos útiles para el aseguramiento de la calidad del producto, que incumben para ampliar mas los conocimientos sobre el tema.

2.5 Conclusiones parciales.

- Se escogen solo los procesos de desarrollo y operación del ciclo de vida definido en el modelo WQM, debido a que las actividades definidas en el proceso de mantenimiento no son realizadas en la universidad.
- Se propuso una serie de actividades para el aseguramiento de calidad de los artefactos generados en cada una de las actividades del ciclo de vida de una aplicación Web. Para que estos se desarrollen de la manera correcta y beneficien el proceso de producción.
- Se logró desarrollar una guía para el aseguramiento de la calidad del producto en aplicaciones Web que aplicada correctamente contribuirá a obtener un producto con niveles adecuados de calidad.

CAPÍTULO III: Validación de la propuesta.

En el presente capítulo se realiza la validación de la propuesta la cual se ejecuta por el método caso de estudio por dos variantes diferentes, la primera para comprobar que tan buena es la guía aplicando una parte de ella (asegurar la calidad de software en el documento especificación de requisitos), y de esta forma demostrar que tan efectiva es, la otra variante es para evidenciar hasta qué punto los usuarios de la guía pueden aprender con ella.

3.1 Método de validación caso de estudio.

Un caso de estudio es “una investigación empírica que estudia un fenómeno contemporáneo dentro de su contexto de la vida real, especialmente cuando los límites entre el fenómeno y su contexto no son claramente evidentes. (...) Una investigación de estudio de caso trata exitosamente con una situación técnicamente distintiva en la cual hay muchas más variables de interés que datos observacionales; y, como resultado, se basa en múltiples fuentes de evidencia, con datos que deben converger en un estilo de triangulación; y, también como resultado, se beneficia del desarrollo previo de proposiciones teóricas que guían la recolección y el análisis de datos.” [Yin. 1994]

Un caso de estudio es un método particular de investigación cualitativa. Se usa en muestras siguiendo un rígido protocolo para examinar un número limitado de variables. Los casos de estudio tienen como característica la profundización y examen longitudinal de una situación, caso o evento. Consiste en una forma sistemática de observar los eventos, coleccionando datos, analizando información y mostrando resultados.

El método caso de estudio es una herramienta que consiste en ejemplos reales en los que se presenta una historia positiva sobre los beneficios que un producto o servicio le ha significado a determinados usuarios.

3.2 Ventajas de utilizar la guía en un proyecto real.

Con la finalidad de validar la propuesta realizada y comprobar que tan buena es la guía, así como con el objetivo de minimizar los problemas existentes en los proyectos que desarrollan aplicaciones Web en cuanto al aseguramiento de la calidad del producto, a través del método caso de estudio se le aplica una porción de la guía, específicamente la parte de aseguramiento de la calidad del artefacto especificación de requisitos, a tres proyectos dedicados al desarrollo

de aplicaciones Web en la universidad los cuales se encontraban en la fase final de desarrollo de la aplicación(transición). Se escoge el artefacto especificación de requisitos por la importancia que este tiene en el desarrollo del software, ya que contiene los requerimientos que debe tener la aplicación, requisitos imprescindibles para la realización y análisis del desarrollo de todo software. Es en la especificación de requisitos donde se realiza una descripción completa del comportamiento del sistema que se va a desarrollar y de ella depende el resultado final del producto y aquellos artefactos que se van obteniendo en el ciclo de vida de la aplicación.

Para validar por el método caso de estudio para analizar las ventajas de utilizar la guía en un proyecto real se realiza un análisis del problema existente antes de existir una guía para asegurar la calidad del software en proyectos que desarrollan aplicaciones Web y después de facilitar una a cada uno de los tres proyectos. La guía elaborada tiene como finalidad eliminar todas las deficiencias existentes en cuanto a aseguramiento de la calidad en aplicaciones Web.

Se definen las siguientes variables:

- Conocimientos en aseguramiento de la calidad del documento especificación de requisitos: Dice el estado del documento, si se realiza de forma completa y correcta.
- Existencia de una guía que ayude al aseguramiento de la calidad del producto en aplicaciones Web: Dice de los problemas que se pueden resolver mediante el uso de la guía para el aseguramiento de la calidad del producto.

Para el aseguramiento de la calidad del documento especificación de requisitos y saber si este se realiza de la forma correcta y óptima se deben tener en cuenta una serie de criterios de evaluación descritos en la guía, los cuales son:

1. Realización del registro de control de versiones del documento.
2. Definición del alcance del documento.
3. Realización de una descripción de la funcionalidad de los requisitos en lenguaje natural.
4. Descripción de todos y cada uno de los requisitos.
5. Identificación de los requisitos que afectan la usabilidad, fiabilidad, eficiencia o soporte del sistema.
6. Especificación, las restricciones relacionadas con el diseño.
7. Especificación de los requisitos para la documentación y ayuda de los usuarios.

8. Identificación de los componentes que han sido adquiridos o comprados para incorporarlos al sistema.
9. Definición de todas las interfaces que deberán ser soportadas por el sistema (Usuario, Hardware, Software, Comunicación)
10. Especificación de todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo entre otras así como cualquier requisito de licencia o restricción de uso.
11. Existencia de evidencias o referencias de las normas o estándares aplicables al sistema, por ejemplo estándares legales, de calidad, normas de usabilidad, etc.
12. Especificación de requisitos correcta. Un conjunto de requisitos software es correcto sólo si todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño”
13. Especificación de requisitos no-ambigua. Un requisito no es ambiguo cuando está sujeto a una única interpretación.
14. Especificación de requisitos completa. Se considera que una especificación de requisitos es completa si describe todos los requisitos relevantes para el usuario, incluyendo requisitos asociados con funcionalidad, actuación, restricciones de diseño, atributos o interfaces externas.
15. Especificación de requisitos consistente. Se entiende por consistente cuando no hay ningún subconjunto de requisitos descrito dentro de la especificación que esté en conflicto con cualquier otro.
16. Especificación de requisitos ordenada por prioridad y estabilidad.
17. Especificación de requisitos verificable: Se considera que una especificación es verificable “si lo son cada uno de los requisitos constituyentes”. A su vez, se considera que un requisito individual es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”.
18. Especificación de requisitos modificable. Se considera que una especificación es modificable “si su estructura es tal que permite realizar cambios sobre los requisitos que contiene de forma sencilla, completa y consistente, manteniendo la estructura inicial del conjunto”.
19. Especificación de requisitos trazable. Una especificación se considera trazable si el origen de cada requisito individual está claro y existe algún mecanismo que permita

seguir el impacto de dicho requisito a lo largo del resto de actividades del ciclo productivo.

A continuación se muestra como se evidencian tales criterios antes y después de existir una guía para el aseguramiento de la calidad del producto en aplicaciones Web.

3.2.1 Antes del uso de la propuesta.

La realización del documento especificación de requisitos en muchas ocasiones es limitado y suplantado por otras tareas, los integrantes del proyecto se dedican al desarrollo de la aplicación, sin realizar la documentación sobre la ingeniería del software y luego de concluir la implementación y tener el resultado final es que se realizan la documentación, incluyendo la especificación de requisitos lo que trae consigo la mala calidad del artefacto influyendo negativamente en siguientes actividades.

En la siguiente tabla se muestra el comportamiento de cada criterio de evaluación en cada uno de los proyectos, se recogen una serie de dificultades contenidas en el documento, a los errores de cada proyecto y la no realización de algunos de los criterios especificando con una (X) el criterio no tenido en cuenta para la realización del documento especificación de requisitos.

Proyectos	Proyecto 1	Proyecto 2	Proyecto 3
Criterios			
Realización del registro de control de versiones de los cambios del documento.	X		
Definición del alcance del documento.	X		
Realización de una descripción de la funcionalidad de los requisitos en lenguaje natural.			
Descripción de todos y cada uno de los requisitos.	X	X	
Identificación de los requisitos que afectan			

la usabilidad, fiabilidad, eficiencia o soporte del sistema.	X		
Especificación, en caso de existir, de todas las restricciones relacionadas con el diseño del sistema.	X	X	
Especificación de los requisitos para la documentación y ayuda de los usuarios.			X
Identificación de los componentes que han sido adquiridos o comprados para incorporarlos al sistema.	X	X	X
Definición de todas las interfaces que deberán ser soportadas por el sistema (Usuario, Hardware, Software, Comunicación)	X		X
Especificación de todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo entre otras así como cualquier requisito de licencia o restricción de uso.			X
Existencia de evidencias o referencias de las normas o estándares aplicables al sistema, por ejemplo estándares legales, de calidad, normas de usabilidad, etc.	X	X	X
Especificación de requisitos correcta.	X	X	X
Especificación de requisitos no-ambigua.			X
Especificación de requisitos completa.	X	X	X
Especificación de requisitos consistente.	X	X	
Especificación de requisitos ordenada por prioridad y estabilidad.			X
Especificación de requisitos verificable:			

Especificación de requisitos modificable.			
Especificación de requisitos trazable.			X

Tabla 5: Evaluación de Criterios en proyectos.

La tabla anterior evidencia cómo generalmente todos los criterios son afectados, es decir no se realizan por un proyecto u otro, excepto la realización de una descripción de la funcionalidad de los requisitos en lenguaje natural y especificación de los requisitos verificable, modificable y trazable.

¿Qué representa no tener en cuenta estos criterios?

- En la realización del documento “Especificación de Requisitos” no tener en cuenta estos criterios representa problemas en posteriores actividades influyendo en el producto final que se desea obtener, elevando los costos del producto y la pérdida de ganancia.
- La no realización del registro de control de cambio impide examinar y hacer seguimiento de las diferentes versiones, así como no poder revertir o comparar versiones diferentes del documento.
- La descripción de cada uno de los requisitos de software es un elemento y criterio indispensable para la especificación de requisitos ya que explica detalladamente cada funcionalidad y restricciones no funcionales que debe tener la aplicación, y de esta forma emplearlos en siguientes actividades que se desarrollen sin mal comprensión de lo que se debe hacer.
- En uno de los proyectos que no realizó la descripción de cada uno de los requisitos ingresó un nuevo integrante, al cual se le comenzaron a dar tareas sin muchas referencias del proyecto. Debido a no poseer la descripción de cada funcionalidad que debía implementar desarrolló alguna de las mismas de forma incorrecta por lo que tuvo que realizarlas nuevamente, lo que implicó pérdida de tiempo y falta de aseguramiento de la calidad.
- En el criterio especificación de los requisitos que afectan la usabilidad, fiabilidad, eficiencia o soporte del sistema uno de los proyectos incurre de forma negativa en él ya que no se definen requisitos de fiabilidad y eficiencia, esto trae consigo que no identifiquen:
 - Para requisitos de fiabilidad: Disponibilidad, horas de uso, acceso para mantenimiento, modo de funcionamiento degradado, tiempo medio entre fallos(horas, días, meses o

años), tiempo medio de reparación teniendo en cuenta cuanto tiempo está permitido que el sistema quede fuera de operación luego de haber fallado así como el máximo de errores permitidos. Para requisitos de eficiencia no se especifica el tiempo de respuesta por transacción, rendimiento, capacidad es decir número de clientes o transacciones que el sistema puede alojar), utilización de recursos (memoria, disco, comunicación, etc.). Toda esta falta de requisitos conlleva a la consecuencia probable que no se tengan en cuenta a la hora del diseño e implementación del software afectando la obtención de un producto final con la adecuada calidad.

- La restricción relacionada con el diseño del sistema es un requisito imprescindible que todo documento de “Especificación de requisito” de proyectos que desarrollan aplicaciones Web debería poseer debido a la importancia que tienen las mismas donde su arquitectura y diseño son elementos que exigen los usuarios y si no son especificadas no se tendrán en cuenta. Uno de los proyectos que no definió este tipo de restricciones llegó a su fase final y con esta la entrega del producto donde el usuario no estuvo satisfecho debido al mal diseño de la aplicación por lo que el producto fue devuelto.
- En la definición de todas las interfaces que deberán ser soportadas por el sistema no se realiza en dos de los proyectos la especificación de la interfaz de comunicación lo cual representa una afectación debido a que no se describen interfaces de comunicaciones a otros sistemas o dispositivos como las redes de área locales y los dispositivos remotos que puede presentar el producto y sistema.
- La no realización de especificaciones de todos los requisitos legales tales como patentes, derechos de autor, marca comercial, logotipo entre otras así como cualquier requisito de licencia o restricción de uso, así como la no existencia de referencias de las normas o estándares aplicables al sistema implica que el documento no esté bajo responsabilidades de las normas político-culturales que existen dentro de la universidad, implicando que tampoco responde a las normas del país, pues no se rige por estándares legales, de calidad, regulatorios, normas de la industria para la usabilidad e integración con el sistema operativo.
- Se puede decir que el documento “Especificación de Requisitos” de los tres proyectos no es correcto ni completo así como la especificación de uno de ellos es ambigua ya que algunos de los requisitos no están sujetos a una única interpretación. Se considera que la especificación de los requisitos de uno de los proyectos no es consistente pues existen subconjuntos de requisitos descritos que están en conflicto con otros ya que

existen incoherencias lógicas entre ellos, así como la especificación del proyecto 3 no es organizada por prioridad y estabilidad.

Es de gran importancia destacar que los tres proyectos se encuentran en sus fases finales y aún presentan problemas en documentos como “Especificación de Requisitos” que es uno de los aspectos que se deben tener en cuenta y tenerlos bien definidos de forma correcta desde los inicios del proyecto de software para que ahora en este momento de fase final el producto adquiera un desempeño superior, sea modificable, seguro y tolerante a fallas o que reúna todas estas cualidades simultáneamente. La mala calidad del documento especificación de requisitos influye de forma negativa en otros artefactos generados de próximas actividades, artefactos como arquitectura del software, modelo de diseño, modelo de diseño Web, modelo de implementación y entregables. Es de gran importancia una excelente especificación de requisitos ya que es esta la que le aporta al usuario la seguridad de que la aplicación se realizará de forma correcta de acuerdo a sus necesidades.

En los proyectos los cuales se le realizó la revisión y verificación del documento especificación de requisitos se detectó que la cantidad de dificultades, errores cometidas en el documento y la no realización de algunas especificaciones representan un 57.89%, 36.84%, 52.63% respectivamente. A continuación se realiza una traficación de esto.

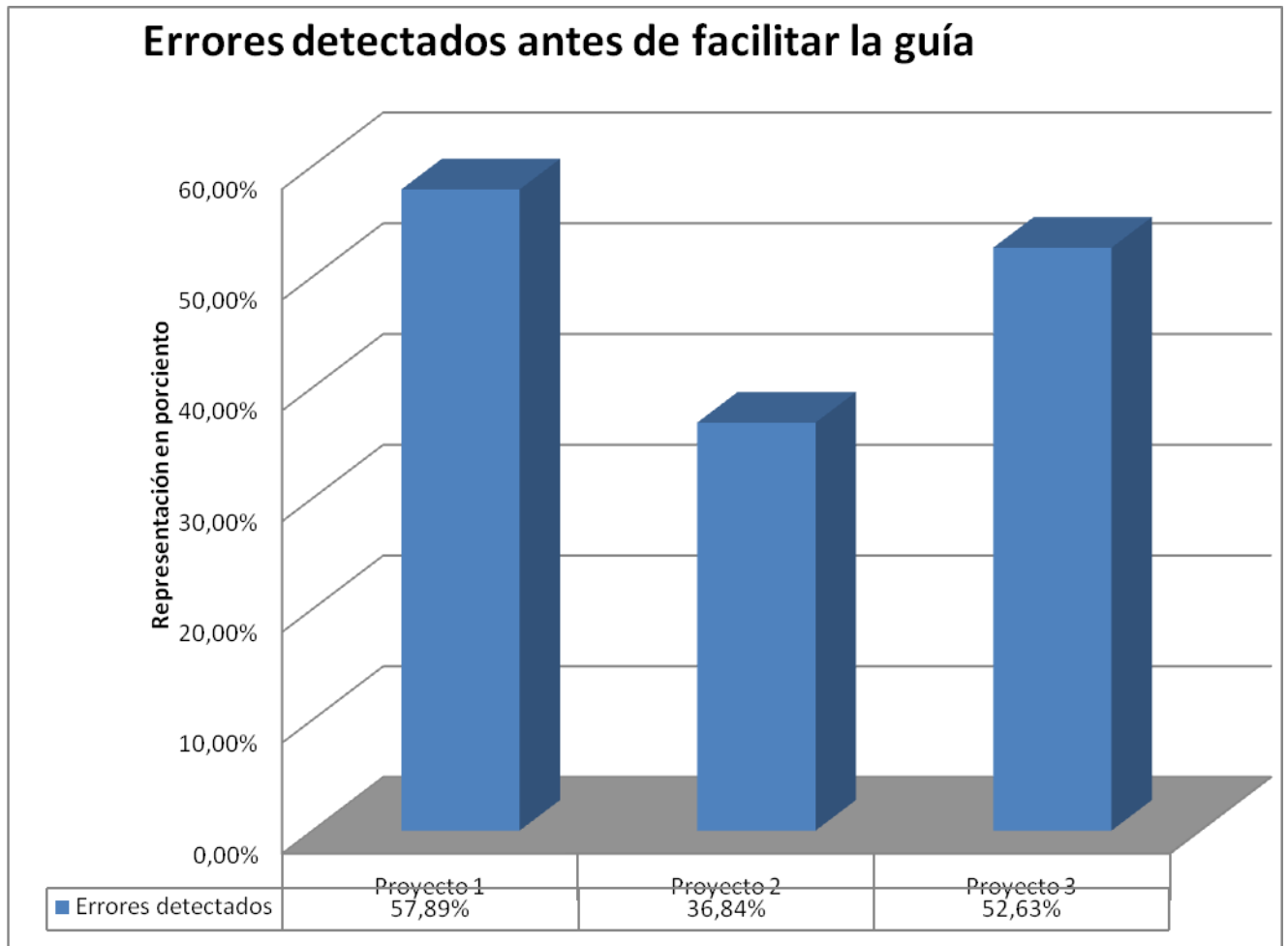


Gráfico 1: Porcentaje de errores al realizar la revisión de “Especificación de requisitos” antes de poseer la guía.

Los resultados obtenidos por la revisión fueron entregados al encargado de la realización del documento “Especificación de Requisitos” (analista) de cada proyecto, conjuntamente con la porción de la guía que trata sobre el aseguramiento de la calidad del artefacto especificación de requisitos para que el analista realice una nueva versión teniendo en cuenta los criterios de evaluación.

3.2.2 Resultados obtenidos después del uso de la guía.

Luego de realizar la nueva versión del documento especificación de requisitos teniendo en cuenta los criterios de evaluación contenidos en la guía propuesta se detectó que los proyectos solo incurrieron dos aspectos (tabla 6).

Proyectos	Proyecto 1	Proyecto 2	Proyecto 3
Criterios			
Realización del registro de control de versiones de los cambios del documento.			
Definición del alcance del documento.			
Realización de una descripción de la funcionalidad de los requisitos en lenguaje natural.			
Descripción de todos y cada uno de los requisitos.			
Identificación de los requisitos que afectan la usabilidad, fiabilidad, eficiencia o soporte del sistema.			
Especificación, en caso de existir, de todas las restricciones relacionadas con el diseño del sistema.			
Especificación de los requisitos para la documentación y ayuda de los usuarios.			X
Identificación de los componentes que han sido adquiridos o comprados para incorporarlos al sistema.		X	X
Definición de todas las interfaces que deberán ser soportadas por el sistema (Usuario, Hardware, Software, Comunicación).			

Especificación de todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo entre otras así como cualquier requisito de licencia o restricción de uso.			
Existencia de evidencias o referencias de las normas o estándares aplicables al sistema, por ejemplo estándares legales, de calidad, normas de usabilidad, etc.			
Especificación de requisitos correcta.			
Especificación de requisitos no-ambigua.			
Especificación de requisitos completa.			
Especificación de requisitos consistente.			
Especificación de requisitos ordenada por prioridad y estabilidad.			
Especificación de requisitos verificable:			
Especificación de requisitos modificable.			
Especificación de requisitos trazable.			

Tabla 6: Evaluación de Criterios en proyectos. Resultados

En la tabla anterior se puede observar como los proyecto no tuvieron en cuenta simplemente dos criterios de evaluación por no ser considerados de importancia para el desarrollo de la aplicación. Esto representando un 0 %, 5.26 % y 10.53 % de errores respectivamente para los proyectos 1, 2 y 3 (gráfico 2), por lo que hubo un avance en 57.89 %, 31.58 %, 42.1 %.

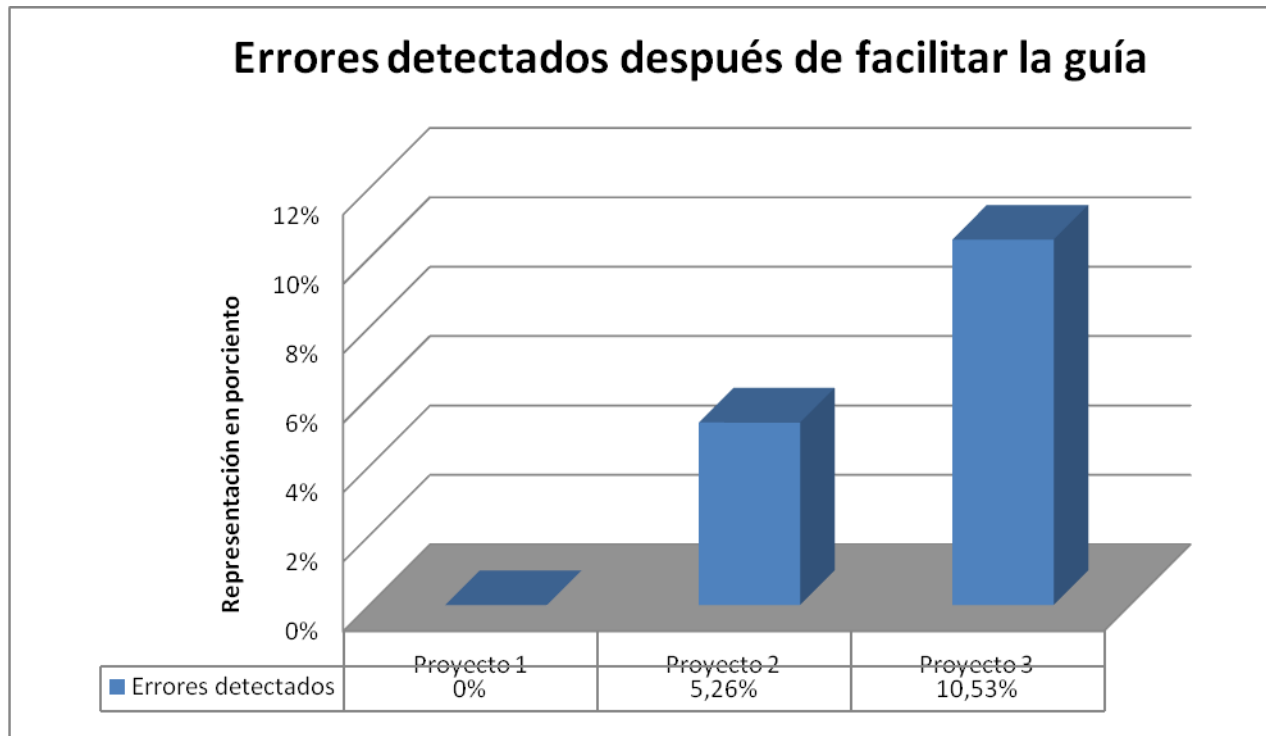


Gráfico 2: Porcentaje de errores al realizar la revisión de “Especificación de requisitos”.

Se demuestra la importancia que tiene la utilización de una guía para el aseguramiento de la calidad del producto en aplicaciones Web en los proyectos de la universidad que se dediquen al desarrollo Web, pues los errores son minimizados ganando en calidad, costo y tiempo pues la especificación es realizada de forma más correcta y completa, y de esta forma no hay que realizar nuevamente el trabajo por lo que se considera que el producto final será de mejor calidad, cumpliendo con todas las expectativas del cliente y satisfaciendo sus necesidades.

3.2.3 Validación por caso de estudio para demostrar que tanto se aprende con la utilización de la guía.

Para la validación por caso de estudio para demostrar que tanto se aprende con la utilización de la guía se plantean dos situaciones, antes de existir la guía y después de poner a disposición una porción de la misma a una muestra de estudiantes escogidos aleatoriamente, teniendo como finalidad la eliminación de las insuficiencias en el conocimiento en cuanto al tema.

Para la realización del caso de estudio se tienen en cuenta variables, tales como:

- Tiempo: Da la medida del tiempo que se necesita para aprender.
- Conocimientos sobre aseguramiento de la calidad en el producto en aplicaciones Web: Representa los conocimientos reales que existen sobre el tema.
- Existencia de una guía para los aseguradores de la calidad de los proyectos que desarrollen aplicaciones Web: Muestra los problemas que se pueden resolver mediante el uso de la guía, facilitando información para asegurar la calidad de los artefactos generados en cada actividad de desarrollo de la aplicación.

Antes de utilizar la guía ¿Qué sucedió?

En los proyectos dedicados al desarrollo de aplicaciones Web solo se cuenta con las listas de chequeo establecidas por la universidad para el aseguramiento de la calidad de algunos de los artefactos generados en cada actividad de desarrollo de la aplicación, por lo que la bibliografía sobre el tema no es abundante, afectando en la producción de un producto totalmente satisfactorio.

Para la realización de la validación se tomó como muestra 30 estudiantes de la facultad escogidos aleatoriamente, algunos de ellos con conocimientos sobre el tema, para aplicarle un cuestionario evaluando los conocimientos reales y actuales sobre el tema.

Se considera que el tiempo empleado para el aprendizaje sobre el aseguramiento de la calidad del producto en aplicaciones Web es más extenso debido a la no existencia de una guía que reúna todas las características y técnicas para el aseguramiento de la calidad.

Al aplicar el cuestionario (Anexo 4: Cuestionario para la validación por el método caso de estudio antes de facilitar parte de la guía para el aseguramiento de la calidad del producto en aplicaciones Web) a los estudiantes de la muestra se observó que no había abundante conocimiento sobre el tema, la gran mayoría no supo responder correctamente sobre qué elementos se deben tener presente para la realización del plan de aseguramiento de la calidad. Debido a la importancia del artefacto plan de pruebas teniendo en cuenta que es en este donde se planifican las pruebas a realizar para detectar fallos de implementación, calidad o usabilidad, se decide conocer si existen conocimientos generales sobre como asegurar la calidad del mismo, lo cual se detectó que solo la minoría tenía conocimientos, cuya minoría representa la porción de la muestra de estudiantes que desempeñan el rol de encargado de la calidad en su proyecto. De los estudiantes encuestados solo 7 de ellos respondieron correctamente a las

preguntas realizadas para medir los conocimientos en cuanto a factores de calidad para aplicaciones Web y métricas existentes para la medición de los mismos y calidad del producto, no ocurriendo así con los estudiantes restantes. (Ver gráfico 3, 4 y 5).



Gráfico 3: Conocimientos sobre criterios a tener en cuenta para realización del plan de aseguramiento de la calidad antes de facilitar parte de la guía.



Gráfico 4: Conocimientos sobre criterios para revisión del plan de pruebas antes de facilitar parte de la guía.

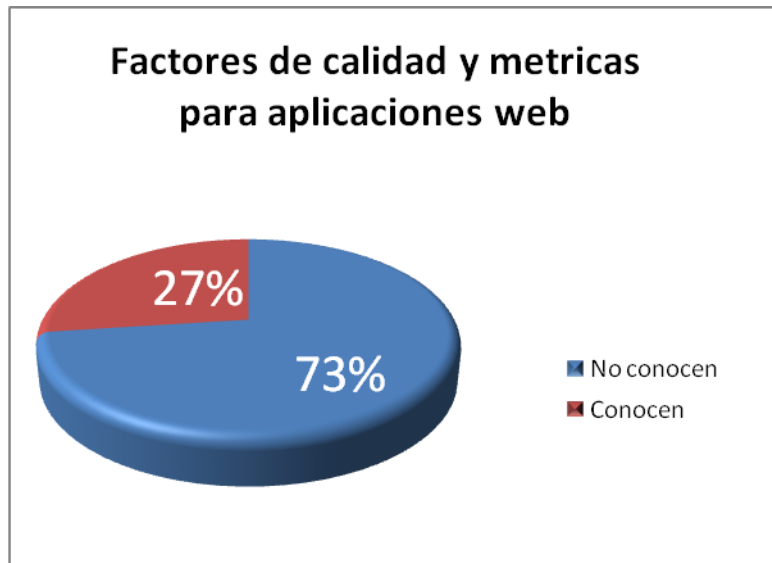


Gráfico 5: Conocimientos sobre factores de calidad y métricas para medir los factores antes de facilitar parte de la guía.

Después de utilizar la guía. ¿Qué resultados se obtuvieron?

Al ser analizados los resultados del cuestionario realizado a la muestra de estudiantes se les facilita una porción de la guía para el aseguramiento de la calidad del producto en aplicaciones Web para que adquieran conocimientos sobre el tema y sepan cómo se desenvuelve el rol de aseguramiento de la calidad en un proyecto. Luego de establecer un tiempo de 5 días se les aplica a los mismos estudiantes otro cuestionario (Anexo 5: Cuestionario para la validación por el método caso de estudio después de facilitar parte de la guía para el aseguramiento de la calidad del producto en aplicaciones Web) para comprobar que tanto aprendieron con la guía y si la misma es eficiente y fácil de comprender.

Para la primera pregunta realizada sobre selección de los criterios que se deben tener en cuenta para la realización del plan de aseguramiento de la calidad, se puede decir que los resultados son satisfactorios ya que se logró que gran parte de los estudiantes sean capaces de identificar los criterios del plan. Con la segunda pregunta del cuestionario se deseaba conocer los conocimientos adquiridos para la evaluación del artefacto plan de pruebas donde los estudiantes contestaron en su mayoría correctamente al igual que para los factores de calidad más importantes y exigidos por los clientes para la realización de una aplicación Web. Con la cuarta pregunta realizada se deseaba conocer si la guía es lo suficientemente

explicativa como para que los estudiantes pudieran aplicar los conocimientos adquiridos en ella donde los resultados arrojados fueron satisfactorios ya que solo 10 estudiantes no respondieron correctamente. En los gráficos (6, 7, 8 y 9) se hace una representación detallada de los resultados obtenidos en el cuestionario realizado.



Grafico 6: Conocimientos sobre criterios a tener en cuenta para realización del plan de aseguramiento de la calidad después de facilitar parte de la guía.



Gráfico 7: Conocimientos sobre criterios para revisión del plan de pruebas antes de facilitar parte de la guía.

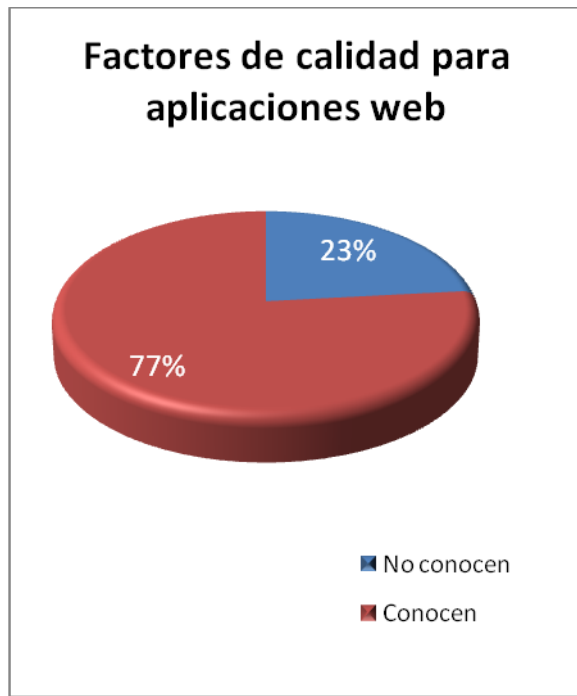


Gráfico 8: Conocimientos sobre factores de calidad para aplicaciones Web antes de facilitar parte de la guía.



Gráfico 9: Conocimientos sobre métricas de calidad para aplicaciones Web antes de facilitar parte de la guía.

Resultados

Luego de realizar un análisis de la situación antes y después de facilitar una guía para el aseguramiento de la calidad del producto en aplicaciones Web, se puede considerar que la guía aumenta los conocimientos en cuanto al tema y los estudiantes son capaces de aprender con ella. Los aseguradores de la calidad de cada proyecto ahorrarán tiempo para documentarse y asegurar los artefactos del proyecto, pues tendrán la información más imprescindible contenida en la guía.

3.3 Conclusiones parciales

- Se realiza la validación por el método caso de estudio para comprobar las ventajas de la utilización de la guía y que tanto se aprende con ella.
- La validación permitió corregir y perfeccionar el artefacto especificación de requisitos de tres proyectos de la universidad dedicados al desarrollo de aplicaciones Web.

- Se concluyó que la guía aumenta los conocimientos en cuanto al aseguramiento de la calidad, es efectiva y de fácil entendimiento.

CONCLUSIONES

Con la aspiración de darle solución a los problemas presentes en los proyectos que desarrollan aplicaciones Web en cuanto a asegurar los artefactos generados por cada actividad de desarrollo de la aplicación con la guía desarrollada, se concluye este trabajo investigativo, en el cual se llevan a cabo las siguientes acciones:

- Se propuso la utilización de una guía para el aseguramiento de la calidad del producto software en proyectos que desarrollen aplicaciones Web en la UCI, utilizando varios modelos de calidad de software como CMMI para las técnicas de verificación y validación, WQM basado en la ISO/IEC 12207 para definir el ciclo de vida de las aplicaciones Web.
- Se aplicó un fragmento de la guía dedicada al aseguramiento de la calidad del artefacto especificación de requisitos a proyectos dedicados al desarrollo de aplicaciones Web.
- Se logró una mejor especificación de requisitos de forma que este documento quedó realizado de forma correcta, completa y optima en cada proyecto al que se le aplicó la porción de la guía.
- Se validó la guía para el aseguramiento de la calidad del producto software en aplicaciones Web obteniendo como resultado la importancia de la utilización de la misma.
- Se comprobó que la guía es de fácil entendimiento y se adquiere mayor conocimiento en cuanto al aseguramiento de la calidad del producto software con su utilización.

RECOMENDACIONES

Debido a la importancia que posee aseguramiento de la calidad en los proyectos de la universidad, se ha considerado realizar algunas recomendaciones con el objetivo de mejorar este aspecto:

- ✓ Se recomienda aplicar la propuesta realizada en los proyectos que se dediquen al desarrollo de aplicaciones Web y evaluar su efectividad para todos los artefactos generados.
- ✓ Dar seguimiento al desarrollo de la guía enfocándola al aseguramiento de calidad del software en el proceso y servicio.
- ✓ Cumplir con lo planteado durante todo el ciclo de desarrollo de una aplicación Web.
- ✓ Realizar estudios sobre los estándares existentes a nivel internacional para el Aseguramiento de la Calidad de proyectos de software.

REFERENCIAS BIBLIOGRÁFICAS

[Dávila 2006] Dávila, A., Melendez, K., Flores, L. Determinación de los Requerimientos de Calidad del Producto Software Basados en Normas Internacionales, 2006. Extraído de: http://www.ewh.ieee.org/reg/9/etrans/vol4issue2April2006/4TLA2_6Davila.pdf el 31 de marzo de 2007.

[FERNANDEZ; GUERRERO. 2008]. Sistema de Gestión de Conocimientos. Ciudad de la Habana: UCI, 2008.

[SCALONE. 2006]. Estudio comparativo de los modelos y estándares de calidad del software. 2006. Disponible en: <http://www.fi.uba.ar/laboratorios/lsi/scalone-tesis-maestria-ingenieria-en-calidad.PDF>

[SEI. 2008]. Software Engineering Institute. [Online] 2008.

Disponible en: <http://www.sei.cmu.edu/cmml/general/index.html>

[DAPENA, Delgado Martha Dunia y otros. 2005] Una Propuesta de Introducción de las Revisiones en el Proceso de Desarrollo de Software. Ciudad de La Habana: Instituto Superior Politécnico "José Antonio Echeverría": s.n., 2005.

[EJIOGO O, Len. 1991] Software Engineering whit Formal Metrics, QED Technical Publishing Group, 1991. Disponible en: <http://www.ipipan.gda.pl/~marek/objects/TOA/moose.html>

[YIN. 1994]. Case Study Research: Design and Methods. Sage Publications, Thousand Oaks, CA. Disponible en:

<http://www.cema.edu.ar/publicaciones/download/documentos/296.pdf>

[PRESSMAN, Roger S. 2002] Ingeniería de Software. Un Enfoque Práctico. [ed.] Mc Graw Hill

[PRESSMAN. 2005]. Ingeniería del Software, Un enfoque práctico. La Habana: Félix Varela, 2005.

[FENTON E. Norman. 1991]. Software Metrics A rigorous approach .Chapman & Hall,

Primera Edición.

[ISO. 9000]: "Normas ISO 9000 y calidad". 2001

[ROMERO Arturo Luis y MIRANDA Sandor Luis] LA CALIDAD, SU EVOLUCIÓN HISTÓRICA Y ALGUNOS CONCEPTOS Y TÉRMINOS ASOCIADOS, 2009
.http://www.gestiopolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm.

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

[Calero 2005 a] Calero, C., Genero, M., Fernández-Medina, E., Piattini, M., Serrano, M. Calidad de sistemas de información, 2005. Extraído de: <http://alarcos.inf-cr.uclm.es/doc/Calidad/capitulo01.ppt> el 7 de diciembre de 2006.

[Calero 2005 b] Calero, C. Modelos de calidad. WQM, PQM, e-commerce, portlets, 2005. Extraído de: <http://alarcos.inf-cr.uclm.es/doc/calidadSI/Metodos%20De%20Calidad.ppt> el 7 de diciembre de 2006.

[Calero 2005c] Calero, C., Ruiz, J., Piattini, M. Classifying web metrics using the web quality model, 2005.

[Quiñones 2007] Quiñones, E. Modelos de Calidad de Software y Software Libre. 2007. Extraído de: http://www.eqsoft.net/presentas/modelos_de_calidad_y_software_libre.pdf el 8 de diciembre de 2006.

[Salanova 2006] Salanova, P. E. Modelos de calidad Web. Clasificación de Métricas. Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingeniería Informática. España, 2006. pp. 87-103. Extraído de: http://www.issi.uned.es/CalidadSoftware/Noticias/PFC_1.doc el 8 de diciembre de 2006.

BIBLIOGRAFÍA

BRAVO Lillo, CRISTIAN y GUERRERO, Luis A. Métricas de Funcionalidad: una taxonomía para sistemas Web. Departamento de Ciencias de la Computación, Universidad de Chile. Extraído el 12 de noviembre de 2007. Disponible en:

<http://www.dcc.uchile.cl/~luguerre/papers/WCIS-01.pdf>.

CUEVA Lovelle, J.M. Métricas de usabilidad en la web. 2007. Extraído el 6 de noviembre de 2007. Disponible en:

<http://www.di.uniovi.es/~cueva/asignaturas/doctorado/2004/MétricasUsabilidad.pdf>

LAFUENTE, G. J. (2000). Automatizando Métricas en la web. Buenos Aires, Argentina, Universidad Nacional de Luján. Extraído el 19 de noviembre de 2007. Disponible en: <http://mdk.ing.unlpam.edu.ar/~labweb/downloads/pdfs/tesiswebsitema.pdf>

CHILE, G. D. and M. S. G. D. Gobierno. Guía para el Desarrollo de Sitios Web 2004. Consultado el 16 de marzo del 2009. Disponible en:

<http://www.guiaweb.gob.cl/guia/capitulos/tres/accesorapido.htm>

DAVILA Nicanor, Leticia y Mejía Álvarez, Pedro. Evaluación de la Calidad de Software en Sistemas de Información en Internet. Sección de Computación. Zacatenco. México, DF. Extraído el 8 de mayo de 2009. Disponible en:

<http://speedvirtualextr.googlepages.com/CalidaddeunSistemadeInformacion.pdf>

NORMALIZACION, O. N. D. Ingeniería de software —Calidad del producto—. Abril 2005, La Habana, Cuba. Consultado el 25 de abril del 2009.

Métricas. Métricas de calidad. Universidad de Guadalajara. Extraído el 8 de mayo de 2009. Disponible en: http://148.202.148.5/cursos/cc321/fundamentos/unidad2/tema2_1.html

OLSILA, L. A. Metodología Cuantitativa para la Evaluación y Comparación de la Calidad, 1999. Consultado el 19 de noviembre del 2007. Disponible en:

<http://www.di.uniovi.es/~cueva/investigacion/tesis/WebsiteQEM.pdf>

LAFUENTE, Guillermo Javier y OLSILA, Luis. “Catalogando Métricas Web”. Facultad de Ingeniería, Argentina. Extraído el 12 de noviembre de 2007. Disponible en:

<http://www.informandote.com/jornadasIngWEB/articulos/jiw07.pdf>

CUELLAR, Yilitbet; y CAPOTE Maylen. “Gestión de requisitos en el desarrollo del software”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Junio 2007.

GUEVARA, Betsy. “Procedimiento Propuesto para medir la Calidad en la Gestión de Requisitos”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Julio 2007.

DIAZ, Andy; y RUIZ Osmel. “Proceso de Gestión de Requisitos en el proyecto 'Sistema Integrado para Bibliotecas'”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Junio 2007.

BACALLAO, Maylín; y PINO, Yaumarys. “Propuesta de un proceso para la Captura de Requisitos de la segunda fase del desarrollo del proyecto de Modernización de los Registro y Notarías de la República Bolivariana de Venezuela”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Mayo 2007.

TEJERA, D. Caridad; y SANCHEZ, L. Bárbara. “Ingeniería de Requisitos para el desarrollo del Sistema de Gestión de Inventario Almacén (SIGIA). Módulo Nomencladores”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Junio 2007.

TABASCO, P. Mailen; MARRERO, Yadira. “Procedimiento para el control y aseguramiento de la calidad en los flujos de trabajo Modelamiento del negocio y Requerimientos de los proyectos de software de la facultad 2 de la Universidad de las Ciencias Informáticas”. [Tesis de graduación]. Universidad de las Ciencias Informáticas. Junio 2007.

ESPASA, -Calpe S.A. 2005. Diccionario de la lengua española. 2005.
<http://www.wordreference.com/definicion/simulador>.

ISO 9000. 2000. "Sistema de Gestión de la Calidad". 2000.
http://fabetsia.dmpa.upm.es/solo_alumnos/sp2/Tablon_sp2/TransparenciasCALIDAD06.pdf.

MENDOZA, Luis Eduardo. 2005. Sistemas de Información III. 2005.
[http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/PS6117%20Calidad%20del%20Software.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/PS6117%20Calidad%20del%20Software.pdf).

NAVARRO, José Luis Tomás y Sergio Pérez Paredes. 2003. Estandar ISO 9000-3. 2003.
<http://www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/092000.ppt#256,1,ESTANDAR>

SCALONE. 2006. Estudio comparativo de los modelos y estándares de calidad del software. 2006. <http://www.fi.uba.ar/laboratorios/lsi/scalone-tesis-maestria-ingenieria-en-calidad.PDF>.

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

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

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

Tecnom maestros. 2008. Tecnom maestros, Estandares de Calidad ISO para Desarrollo de Software . [Online] 2008. http://tecnom maestros.awardspace.com/estandares_iso.php.

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

SEI. 2008. Software Engineering Institute. [Online] 2008.
<http://www.sei.cmu.edu/cmami/general/index.html>.

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

Navarro, J. 2005. Escuela de Ingeniería Pontificia Universidad Católica de Chile. [Online]

2005. <http://www2.ing.puc.cl/~jnavon/IIC2142/Clases2005/11.IntroTesting.pdf>.

Mendoza, G. M. (2007). Métricas Internas de la Calidad del Producto de Software.

Universidad Autónoma de Querétaro.

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

Dergarabedian, C. (2008). <http://tecnologia.infobaeprofesional.com>. Obtenido de

<http://tecnologia.infobaeprofesional.com/notas/25157-Normas-de-calidad-en-softwareuna-llave-que-abre-mercados.html>

LIBROS CONSULTADOS

TOIRAN, Y. 2006. Organización en Procesos del Departamento de Detención a Estudiantes Extranjeros. 2006.

PRESSMAN, Roger S. 2005. Ingeniería del Software, Un enfoque práctico. La Habana: Félix Varela, 2005

ARTER, D. "Auditorías De Calidad Para Mejorar La Productividad". 2003.

BEDINI, A. Extracto del libro en formato digital "Calidad tradicional y de Software".

CUEVAS, A. B. M. Y. J. M. "Estándares y Guías", 2001.

DIEZ, E. "Calidad del Software. CMM: Capability Maturity Model".

- HUMPHREY, W.** “Introducción al Proceso de Software Personal”, 2001.
- HUMPHREY, W.** “A self improvement Process for Software Engineers”, 2005.
- CASERES, E. 2006.** El método Delphi.Características. 2006. <http://www.codesyntax.com>.
- CARRASCO, F. y O. M, et al. 2004.** Un enfoque actual sobre la calidad del software. 2004.
- BUADES, Rubio Gabriel. 2002.** Calidad en ingeniería del Software. 2002.
- CMMI. 2006.** CMMI for Development, Version 1.2. 2006.
- ISO/IEC 9126:** Tecnología de la información. Evaluación del producto software. Características y directrices de calidad para su uso, 1991.
- JACOBSON, I.; BOOCH, G. y RUMMBAUGH, J.** “El Proceso Unificado de Desarrollo de software”. 2000.
- MINGUET, J; Hernández, J;** “La calidad del software y su medida”. 2003.
- ANDUJAR U, Juan Antonio. 2004.** Métricas de Calidad para Proyectos de Software. 2004. <http://www.pmies.org/ponencias/2004/ponencia%20Juan%20A%20Andujar.pdf>.
- ANTONIO, A. D. 2002.** Gestión, Control y Garantía de la Calidad de Software. 2002. http://www.inf.uach.cl/rvega/asignaturas/info265/G_Calidad.pdf.
- BAEZA, Yates R. A. y otros. 1995.** Computing in Chile:The Jaguar of the Pacific Rim Communications of the ACM. 1995. Vol. 38.
- IEEE. 1990.** IEEE Standard Glossary of Software Engineering Terminology. 1990.
- JACOBSON, Ivar, Booch, Grady y Rumbaugh, James. 2000.** El Proceso Unificado de Desarrollo de Software. 2000.
- Dapena, Delgado Martha Dunia y otros. 2005.** Una Propuesta de Introduccion de las Revisiones en el Proceso de Desarrollo de Softwre. Ciudad de La Habana : Instituto Superior Politecnico "José Antonio Echeverría" : s.n., 2005.

FEBLES, Estrada Ailyn. 2000. Calidad de Software y la empresa, enseñanza de un tema imprescindible para el Ingeniero Informático. Instituto Superior Politécnico “José Antonio Echeverría: s.n., 2000.

FERRE, Xavier rGrau. 2005. Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software. 2005.

ANEXOS

Anexo 1: Guía de observación.

- Capacitación a los encargados de la calidad del software en su proyecto.
- Actividades que realizan los encargados de la calidad del software en su proyecto.
- Metodologías usadas encargados de la calidad del software en su proyecto.
- Bibliografía consultada por los encargados de la calidad del software en su proyecto.

Anexo 2: Encuesta a los líderes y encargados de la calidad de software.

Con la ayuda de este cuestionario nos hemos propuesto identificar las potencialidades y deficiencias existentes en el proceso de calidad de software en los proyectos de la Universidad de las Ciencias Informáticas (U.C.I.). Le pedimos su más sincera colaboración con respecto a las respuestas nosotros a su vez le garantizamos confidencialidad y anonimato a sus respuestas, solo es necesario que se especifique el rol que desempeña dentro del proyecto al que usted pertenece.

Rol _____

Responda las siguientes preguntas y marque con una x en el caso que sea necesario:

1-¿Están definidos los roles y responsabilidades de los miembros del proyecto de forma que garanticen la calidad requerida del producto?

Sí__ No__

En caso afirmativo responda la siguiente pregunta:

¿Cuáles son los roles que se encargan de esta actividad?

2-¿Cuenta su proyecto con información que garantice el Aseguramiento de la calidad del producto?

Sí__ No__

a- En caso afirmativo responda la siguiente pregunta:

¿Qué beneficios le aporta a su trabajo?

3-¿Se les ofrece la capacitación necesaria a los estudiantes que ingresan por primera vez al proyecto que adoptan el rol con respecto a la calidad del producto software?

Sí__ No__ A veces__ Nunca__

A través de cual o cuales, guías o documentos lo hacen.

4-¿Existe una especial presión en los desarrolladores a aceptar con facilidad formación de pruebas al producto?

Sí__ No__ A veces__ Nunca__

¿Cómo lo hacen?

5-Se realizan actividades referentes al Aseguramiento de la calidad del producto.

Si ___ No ___ Nunca___

a-Mencione Algunas

6- ¿Su proyecto cuenta con una guía que se ajuste a sus necesidades y que sea conveniente y fácil de usar para asegurar eficientemente la calidad del producto? Sí___ No___

a- En caso de tenerla ¿Considera que resuelve el problema de la calidad del software?

Sí___ No___ A veces___ Nunca___

b-¿Qué otras cosas usted considera que deba tener?

7- ¿Se usa en su proyecto una metodología en específico que garantice la mejora de la calidad del software?

Sí___ No___

a-Si su respuesta es afirmativa mencione cual o cuales emplean en su proyecto.

b-En caso de que su respuesta sea negativa. ¿Cual piensa que se podría utilizar?

8. ¿Qué estrategia utilizan para la mejora del producto?

9. ¿Se usan Estándares de Calidad? Si _____ No _____

a-Si su respuesta es afirmativa mencione cual o cuales emplean en su proyecto.

10. Aporte brevemente su opinión con respecto a cómo se lleva a cabo el proceso de Aseguramiento de la calidad del producto en su proyecto. Especifique si se realiza bien, mal o si necesita algunos cambios. Mencione en caso de existir que cambios deberían hacerse.

11. Mencione que necesita el proyecto para que mejore el proceso de Aseguramiento de la calidad del producto.

Como clasifica usted a la universidad con respecto a la calidad de los productos que se desarrollan en los proyectos.

Anexo 3: Entrevista realizada a especialistas en aseguramiento de la calidad del software.

Para el desarrollo del software se tienen en cuenta un conjunto de actividades para ello es de suma importancia que usted como especialista manifieste su opinión en las preguntas realizadas a continuación:

Nombre del entrevistado:

Proyecto donde ejerce el trabajo:

- 1- ¿Cuál es su rol dentro del proyecto?
- 2- ¿Qué función o funciones ejerce en este?
- 3- ¿Cómo definen el ciclo de vida del software a desarrollar?
- 4- ¿Cuáles son los métodos, herramientas que utilizan o los más utilizados en el mundo para las aplicaciones web?
- 5- ¿Qué procedimientos cree usted que se deban desarrollar para asegurar la calidad del producto software?
- 6- ¿Cuáles son, en caso de existir las técnicas de aseguramiento más utilizadas en el proyecto?
- 7- ¿Cuáles son los errores más frecuentes encontrados en los proyectos que afectan la calidad? ¿Se encuentran con más frecuencia en el producto o en los procesos? ¿A qué se debe?

Anexo 4: Cuestionario para la validación por el método caso de estudio antes de facilitar parte de la guía para el aseguramiento de la calidad del producto en aplicaciones Web.

Para la realización de la validación de la guía para asegurar la calidad del software en proyectos que desarrollan aplicaciones Web es necesaria su ayuda y colaboración. Se le pide que llene el siguiente cuestionario.

1. ¿Qué elementos se deben tener presente para la realización del plan de aseguramiento de la calidad? Responda con no menos de 4 elementos.

2. Cómo aseguraría la calidad del artefacto Plan de Pruebas. Responda teniendo en cuenta no menos de 4 elementos.

-
-
3. Cuáles son los factores de calidad para aplicaciones Web que los clientes más exigen.

4. Qué métricas se utilizan para medir los factores de calidad que los clientes más exigen. Responda para al menos un factor.

Anexo 5: Cuestionario para la validación por el método caso de estudio después de facilitar parte de la guía para el aseguramiento de la calidad del producto en aplicaciones Web.

Para la realización de la validación de la guía para asegurar la calidad del software en proyectos que desarrollan aplicaciones Web es necesaria su ayuda y colaboración. Se le pide que llene el siguiente cuestionario.

1. De los criterios descritos a continuación diga cuales son los que se deben tener en cuenta para la realización del plan de aseguramiento de la calidad:

_____ Define claramente los objetivos de calidad

_____ Compromiso

_____ Describe la estructura organizacional de todos los responsables de la calidad así como las tareas y los responsables

_____ Expectativas

Riesgos

_____ Describen las métricas utilizadas como parte del monitoreo y control del trabajo

_____ Referencia cada uno de los estándares y guías utilizados por el Plan de Calidad

_____ Entorno

_____ Describe los tipos de revisiones y auditorias planificadas en el proyecto así como un cronograma con la planificación de las mismas

_____ Cambios Permitidos

_____ Líneas de Comunicación

_____ Identifica los involucrados o los grupos específicos que intervienen en las revisiones y auditorias programadas

_____ Crear un procedimiento o referencia al mismo de las tareas a realizar para la corrección de los defectos detectados

_____ Definir un plan de pruebas o referencias al mismo

_____ Describe las herramientas, técnicas y metodologías a utilizar durante las revisiones y auditorias

_____ Cronograma

_____ Describe los registros que se mantendrán durante el proyecto, incluyendo su ubicación.

_____ Procedimientos.

_____ Describe las actividades de formación o capacitación necesarias para que el personal responsable ejecute las tareas correspondientes del Plan de Aseguramiento de la Calidad.

_____ Recursos.

2. Teniendo en cuenta los criterios de evaluación siguientes diga cuales corresponden a la evaluación del artefacto Plan de Pruebas:

_____ Realización de la descripción del equipo de pruebas así como los roles y responsabilidades de cada uno de sus miembros.

_____ Todos los estereotipos de componentes deben ser correctamente identificados (subsistema, componentes).

_____ Realización de una descripción del escenario en el que se ejecutarán las pruebas incluyendo modelo de despliegue o referencia al mismo, requerimientos de software y hardware del sistema.

_____ Debe existir relación entre los componentes.

_____ Lista de los requerimientos que serán objetos de las pruebas o se hace referencia al documento "Especificación de Requisitos".

_____ Cada componente debe implementar la funcionalidad que se especifica en las clases de diseño.

_____ Descripción la estrategia que se llevará a cabo para ejecutar las pruebas.

_____ Especificación de los criterios de evaluación de las pruebas, así como la clasificación de las no conformidades, pedidos de cambios y listas de chequeo.

_____ Especificación de la parte del cronograma del proyecto que abarca la etapa de pruebas.

_____ Se identifican todos los componentes ejecutables.

3. Cuáles de los siguientes factores de calidad son los más exigidos por los clientes para su presencia en una aplicación Web.

___ Capacidad de mantenimiento.

___ Facilidad de uso.

___ Compatibilidad.

___ Extensibilidad.

___ Verificabilidad.

___ Funcionalidad.

___ Reparabilidad.

___ Reutilización.

___ Oportunidad.

___ Portabilidad.

___ Corrección.

___ Usabilidad.

___ Economía.

___ Eficiencia.

___ Integridad.

___ Fiabilidad.

___ Robustez.

___ Confiabilidad.

4. Considere que en la facultad 10 en el proyecto Portales se ha realizado una aplicación Web para las casas protocolos del consejo de estado. Calcule teniendo en cuenta los siguientes datos cuán usable es la aplicación.

Cantidad de tareas terminadas = 15.

Cantidad de tareas planificadas = 10.

Cantidad de tareas a medio terminar = 5.

Cantidad total de tareas = 30.

Cantidad total de palabras = 100.

Número de frases = 5.

Número de palabras de 3 ó más sílabas = 75.

Anexo 6: Registro del Asegurador de la calidad del software.

REGISTRO DEL ASEGURADOR DE LA CALIDAD V1.0

**<Nombre del proyecto>
<Nombre del producto>
<Versión>**

Artefacto/ Entregable	No	Estructura del Documento	Semántica del documento	Descripción de artefacto	Criterios de evaluación	Técnicas de aseguramiento utilizadas

Criterios de evaluación	Observaciones

Artefacto/ Entregable	Evaluación del artefacto

Anexo 7: Guía para asegurar el software en proyectos que desarrollan aplicaciones web.

La guía está estructurada por:

- Introducción.
- Descripción de la guía.
- Desarrollo de las fases.
- Conclusiones.

Autoras: Zuleydis Caridad Baralt Rivera / Yodelys castro Perez

Fecha: 5/2/2009

Universidad de las Ciencias Informáticas

GLOSARIO DE TERMINOS

Prueba unitaria: Consiste en probar o testear piezas de software elementales. Dichas pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código reducidas.

Acción correctiva: Acción para eliminar la causa de una no conformidad detectada u otra situación no deseada.

Artefacto: Pedazo de información producido, modificado o usado por un proceso. Son resultados tangibles del proyecto, las cosas que se van creando y usando hasta obtener el producto final.

Calidad: Capacidad de cumplir con las características inherentes de un producto, componente de proceso, o proceso para satisfacer las exigencias de los clientes.

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

Control de la calidad: actividad que se realiza para detectar los defectos del producto que se desarrolla.

Conformidad: Cumplimiento de un requisito.

CMMI: Modelo de Madurez de Capacidad Integrado.

DCS: Dirección de Calidad del Software.

Defectos: Es el incumplimiento de los requerimientos de uso previstos. Fallo detectado después de la entrega al cliente.

Estructura cognitiva: Conjunto de conceptos, ideas que un individuo posee en un determinado campo del conocimiento, así como su organización.

ISO: Organización Internacional para la Estandarización.

Inspección: Revisiones independiente a un producto de trabajo o conjunto de productos de trabajo para evaluar si está acorde con las especificaciones, estándares, acuerdos contratados u otros criterios.

LMC: Lineamientos Mínimos de Calidad.

Metodología: Un sistema de principios y normas generales de organización y estructuración teórico-práctica de actividades.

Métricas: Continúa aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa significativa y oportuna.

No conformidad: Incumplimiento de un requisito.

Proceso: Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados, consumiendo recursos.

Proyecto: Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos.

Registro: Documentación que proporciona evidencia objetiva de las actividades realizadas o de los resultados logrados.

Requisito: Condición o cualidad que debe cumplir alguien o algo.

Revisión: Reuniones de un grupo definido de personas cuyo objetivo es encontrar errores en un artefacto de software.

Requerimiento: Condición o capacidad necesitada por un usuario para resolver o lograr un objetivo.

Releases: Liberaciones de un producto de software.

RTF: Revisión Técnica Formal.

RoI: Define el comportamiento y responsabilidades de un individuo o un grupo de individuos que trabajan en equipo.

SQA: Aseguramiento de Calidad.

SGC: Sistema de gestión de conocimientos.