

Universidad de las Ciencias Informáticas
“Facultad 1”



**Título: “Propuesta de Plan de Aseguramiento
de la Calidad del Proyecto Servicios
Comunitarios”**

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

***Autora:** Yanet Coba Santiesteban*

***Tutora:** Ing. Ilmaris Delia López Arias*

“Año 50 de la Revolución “

Ciudad de la Habana

Junio, 2008

“El ignorante afirma, el sabio duda y reflexiona”

Aristóteles

DECLARACIÓN DE AUTORÍA

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

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

Yanet Coba Santiesteban

Ilmaris Delia López Arias

Firma del Autor

Firma del Tutor

Datos de Contacto

Tutor: Ilmaris Delia López Arias

Graduado de la UCI, año 2007. Instructor recién graduado.

Agradecimientos

Agradezco:

A mis padres, por ser tan especiales, mejor no los podría pedir.

A mi tío Turcio, mi abuelito y Walter.

A mis tías Martha y Lilia, Alfredo y Paco.

A mis queridas primas, Melva, Tamara, Ileana, Tania y Ania.

A mi primito querido Fidel Ernesto.

A mis amigas Anny, Yaimara, Laritza, Miriela, Yanet Álvarez, nunca me olvido de ustedes.

A mis compañeras y amigas durante estos 5 años, Edicta, Adriana, Meylin, Yadira Morales, Linet, Lili, Maylen, Yadira Corrales, Arlenys, Iyugnis y Yen.

A mis tíos y primos, Eslinda, Elio, Walter, Gladis, Maylen, Rolando, Tatico, Waltersito, Yadier, Mariam, Mauri, Tania y mi hermano Adrian.

A mis otros amigos, Yanet García, Yenisey, Nimia, Kizzy, Aliesky, William y Braisis.

A toda mi familia y mis vecinos.

A mis amistades del pre y todas las personas que he conocido en estos últimos años.

A mi tutora Ilmaris.

A la Revolución y nuestro Comandante.

Dedicatoria

Dedico este trabajo:

Especialmente a mi abuelita Gloria, siempre estarás conmigo.

A mi mamá y mi papá, ustedes son mi vida.

Resumen

Al desarrollar un software, el principal objetivo es obtener un producto con la calidad requerida. Actualmente, la calidad del software requiere de un mayor estudio y trabajo, pues en algunos proyectos de la Universidad de las Ciencias Informáticas no se lleva un buen aseguramiento de la misma, esto trae como consecuencia que el producto final no cumpla con las expectativas y necesidades del cliente.

El presente trabajo se realizó con el objetivo de proponer un plan, el cual sirva como guía para lograr el Aseguramiento de la Calidad del proyecto Servicios Comunitarios. Para realizar la propuesta se hizo necesario el estudio de los principales aspectos concernientes al tema de calidad y los objetivos del proyecto como tal.

En el plan se propone el uso de métricas, normas, revisiones y auditorías, pruebas, y una serie de actividades que deben ser realizadas dentro del proyecto para el entrenamiento del equipo de aseguramiento de la calidad, todo esto basándose en las necesidades y objetivos del proyecto.

PALABRAS CLAVE

“Calidad, Aseguramiento de la Calidad del Software, Plan de Aseguramiento de la Calidad del Software”

Índice

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.2 Antecedentes.....	6
1.3 Situación actual	7
1.4 Calidad del software en Cuba.....	9
1.5 ¿Qué es Calidad?	9
1.5.1 ¿Qué es Calidad de Software?	10
1.6 Aseguramiento de la Calidad del Software.	12
1.6.1 Control de la Calidad	13
1.6.2 La Garantía de la Calidad	13
1.6.3 Gestión de la Calidad	14
1.7 Plan de Aseguramiento de la Calidad del Software	14
1.8 Estudio de planes existentes	15
1.8.1 Plan de Aseguramiento de la Calidad propuesto por el proyecto Akademos.....	15
1.8.2 Plan de Aseguramiento de la Calidad propuesto por la IEEE 730-1998	17
1.9 Estándares.....	20
1.9.1 ISO 9001.....	21
1.9.2 ¿Qué es CMMI?.....	21
1.9.3 Proceso Personal de Software (PSP).....	23
1.9.4 Proceso de Software del Equipo (TSP)	23
1.9.5 ISO	24
1.10 Métricas de Software.....	24
1.11 Factores que determinan la calidad del software	26
1.11.1 Factores de calidad	27
1.12 Verificación y validación.....	29
1.13 Gestión de configuración del software.	31
1.14 Conclusiones.....	32
CAPÍTULO 2 PROPUESTA DE SOLUCIÓN	33
2.1 Introducción	33

Índice

2.2 Descripción del proyecto productivo Servicios Comunitarios.....	33
2.3 Proceso de Aseguramiento de la Calidad del Software en el Proyecto Servicios Comunitarios.....	34
2.4 ¿Qué objetivo se persigue con la propuesta del Plan de Aseguramiento de la Calidad del Software?	34
2.5 Necesidad del establecimiento de un Plan de Aseguramiento de la Calidad del Software.	34
2.6 Actividades que dan lugar al Plan de Aseguramiento de la Calidad del Software	35
2.6.1 Objetivos de la Calidad.....	36
2.6.2 Organización del equipo de Aseguramiento de la Calidad.....	37
2.6.2.1 Tareas y responsabilidades	38
2.6.3 Métricas propuestas	40
2.6.3.1 Métricas externas de funcionalidad.....	41
2.6.3.2 Métricas de calidad para la especificación de requisitos	46
2.6.3.3 Otras métricas	47
2.6.4 Estándares.....	49
2.6.4.1 Organización Internacional de Normalización (ISO)	49
2.6.4.2 Modelo de madurez de capacidad integrada.....	56
2.6.5 Proceso de revisiones y auditorías	66
2.6.5.1 Plan de revisiones y auditorías.....	68
2.6.6 Herramientas, Técnicas y Metodologías	75
2.6.7 Gestión de Configuración.....	75
2.6.8 Pruebas y evaluaciones	75
2.6.9 Entrenamiento	78
2.7 Conclusiones	79
CAPÍTULO 3 EVALUACIÓN DE LA PROPUESTA	80
3.1 Introducción	80
3.2.1 Determinar la cantidad de expertos	82
3.2.2 Confeccionar el listado de expertos	82
3.2.3 Confirmar la participación de los candidatos.....	83
3.3 Elaboración de los cuestionarios.....	83
3.4 Resultados de la evaluación	83
3.5 Conclusiones	86

Índice

CONCLUSIONES.....	87
RECOMENDACIONES.....	88
BIBLIOGRAFÍA.....	89
ANEXOS.....	92
GLOSARIO DE TÉRMINOS.....	95

Índice

Índice de Tablas

Tabla 1 Tareas y responsabilidades del equipo SQA	40
Tabla 2 Revisiones y auditorías por fase.	71
Tabla 3 Organización y responsabilidades dentro del plan de revisiones y auditorías.	75
Tabla 4 Prioridad de las actividades	85

Índice de Figuras

Figura 2.1 Representaciones de CMMI	59
Figura 2.2 RUP en dos dimensiones	62
Figura 3.1 Método Delphi.....	81
Figura 3.2 Coincidencia de los expertos en actividades más importantes.....	85

Introducción

La tecnología ha ido avanzando continuamente, influyendo en la rama de la ciencia y en los sectores de la sociedad. Una de las industrias que ha tenido un gran desarrollo es la del software, la cual ha evolucionado rápidamente. Son muy pocos los países que hoy en día no desarrollan software. Al existir tantas industrias productoras de software surge la siguiente pregunta ¿Los productos de las diferentes industrias de software tienen la misma demanda?

Como resultado de la gran competencia que existe siempre hay algunas empresas que tienen más demandas que otras, esto se debe a que algunas cumplen mejor los parámetros por los cuales el cliente valora cual empresa escoger. Los principales parámetros para preferir un producto son la rapidez de entrega, el costo y muy importante la calidad del producto.

En nuestro país se realizan algunas inversiones en la industria del software, pues informatizar la sociedad y convertir dicha industria en un sector importante de la economía nacional son objetivos estratégicos definidos por el gobierno para los próximos años, de ahí se hace necesaria la obtención de productos competitivos que permitan introducir el producto cubano en el mercado internacional. La calidad es hoy una de las mayores ventajas competitivas para las empresas desarrolladoras de software.

La Universidad de las Ciencias Informáticas fue creada al calor de la Batalla de Ideas con el objetivo de formar ingenieros en ciencias informáticas que sean capaces de informatizar el país completo , en este centro universitario se producen software para el beneficio de la sociedad cubana y del mismo centro. La Universidad esta compuesta por 10 facultades donde cada una se dedica específicamente a un perfil en la rama de la informática.

La facultad 1 se especializa en la informatización, su objetivo es digitalizar los servicios que brinda la Universidad. Existen varios proyectos en la facultad que se encargan de esta tarea, uno de estos proyectos es Servicios Comunitarios el cual se dedica específicamente a informatizar algunos servicios, ejemplo de estos es la reservación de transporte, reservación del gas, entre otros.

Una de las mayores preocupaciones de las organizaciones que desarrollan software es el nivel de calidad del producto, y nuestra Universidad no está ajena a este problema. Las causas que provocan una mala calidad del trabajo es que existan prácticas pobres de ingeniería, carencia de métricas de calidad, un inadecuado entrenamiento en calidad, así como la no existencia de una planificación real en algunos proyectos, todo esto trae como consecuencia que el tiempo de pruebas sea impredecible y que los productos tengan muchos defectos.

Situación Problemática

Se está desarrollando el proyecto Servicios Comunitarios, capaz de automatizar los servicios que se prestan en la Universidad. Por la importancia que reviste la automatización de servicios que influyen directamente en la vida universitaria, se necesita de un sistema que garantice, no solo la automatización de los procesos sino que lo haga con la máxima calidad.

En el desarrollo del sistema se generan varios artefactos los que constituyen productos de software, la realización con calidad de cada artefacto asegura la calidad en si del producto final.

En estos momentos, la calidad se lleva en el proyecto de forma empírica, no existen planes que garanticen el aseguramiento de la calidad, ni cronogramas de pruebas o validación de los artefactos que se generan. De aquí que el **problema de investigación** sea: ¿Cómo lograr que el plan de aseguramiento de la calidad que se propone para el desarrollo del proyecto Servicios Comunitarios garantice la calidad del producto final?

Como **Objetivo General** se plantea garantizar la calidad del producto a través de una propuesta de un Plan de Aseguramiento de la Calidad del Software. Como **objetivos específicos** se tienen los siguientes:

- ❖ Realizar un plan de calidad que se ajuste a las normas y estándares expuestos por calidad.
- ❖ Efectuar diferentes revisiones en el proyecto Servicios Comunitarios, con el propósito de lograr que este se ajuste a los requisitos expuestos por el cliente.
- ❖ Proponer métricas de software.

- ❖ Reunir los requisitos convenidos con el cliente y superarlos.

Para cumplir los objetivos se plantean las siguientes **tareas científicas**:

- ❖ Estudio de diferentes enfoques del aseguramiento de la calidad según algunas de las normas, modelos y estándares más importantes a nivel internacional para el desarrollo de software.
- ❖ Estudio de planes de aseguramiento de la calidad existentes.
- ❖ Estudio de el estado del arte de herramientas informáticas existentes para el Aseguramiento de la Calidad.
- ❖ Entrevistas a especialistas en Calidad.
- ❖ Elaboración de un plan de actividades para el aseguramiento de la calidad del proyecto.
- ❖ Realización de la documentación de todo el trabajo.

Objeto de estudio:

La calidad de los sistemas de software.

Campo de Acción:

Plan de Aseguramiento de la Calidad del Software.

Para guiar la investigación se plantean como **preguntas científicas** las siguientes:

¿Permitirá la propuesta de un plan de aseguramiento lograr una buena calidad del producto?

¿Garantizará el plan propuesto asegurar la calidad del producto final?

Y como **idea a defender**:

Con una buena propuesta de un plan para el aseguramiento de la calidad se garantizará la calidad del producto final.

Modelo metodológico:

Población:

En el desarrollo de este trabajo investigativo, basado en la calidad del software, se necesitó trabajar con datos reales, siendo muy útil el uso de una población que recogiera lo referente al área de movimiento. Como esta investigación persigue el objetivo de asegurar la calidad del proyecto Servicios Comunitarios, la población está constituida, en este caso, por el proyecto Servicios Comunitarios.

Métodos científicos de investigación:

En el desarrollo de esta investigación se utilizaron, teniendo en cuenta la línea de estudio, métodos teóricos y empíricos, a través de los cuales se podrá tener una idea más detallada acerca de lo que se está investigando, pues este estudio estará dedicado a analizar las partes del objeto de investigación que no se observan directamente. Para ello se hace necesario que las estrategias utilizadas estén de acuerdo con las condiciones específicas que predominan en el problema planteado para así orientar y organizar de forma general esta actividad científica.

Después de lo antes expresado los métodos utilizados son:

Métodos teóricos

El Analítico Sintético, pues a partir de un estudio detallado de las teorías, tendencias y documentos relacionados con el tema, se pudo sintetizar los elementos más importantes y de mayor utilidad para el desarrollo del trabajo.

El Histórico Lógico, pues estudia toda la trayectoria, evolución y desarrollo de los fenómenos.

Métodos empíricos

Se pondrán también en práctica los métodos empíricos de encuesta y entrevista que permiten obtener información cualitativa sobre el fenómeno que se investiga.

El Método de la observación, con la investigación en pie se hace necesaria una observación científica del fenómeno para poder tener una percepción planificada dirigida a un fin. Es un paso importante para el científico que la realiza de forma consciente, y es orientada siempre a un objetivo determinado.

La presente tesis consta de 3 capítulos, el capítulo I contiene la fundamentación teórica, se hace un estudio de diferentes bibliografías para ver el estado del arte de la calidad del software, en el capítulo 2 se hace la propuesta de solución y en el tercer capítulo se evalúa la misma.

Capítulo 1 Fundamentación Teórica

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En la actualidad el término calidad de software está siendo ampliamente difundido y utilizado, un aspecto fundamental que se plantea de un software que tenga calidad es que esté libre de errores en su etapa de ejecución y que además sea fácil su interacción así como que cumpla con las expectativas del cliente, es decir que cumpla con los requisitos explícitos e implícitos. El Aseguramiento de la Calidad pretende dar confianza en que el producto reúne las características necesarias para satisfacer todos los requisitos.

En el presente capítulo se hace un análisis crítico del concepto de calidad del software y aseguramiento de la calidad del software, todo lo referente a este tema en cuanto al estado del arte en el Mundo, y en Cuba, además se abordan los términos que sirven de soporte teórico a la investigación desarrollada y que estarán presentes a lo largo de este trabajo de diploma.

1.2 Antecedentes

A diferencia de otras industrias, la industria del software tiene muy poco tiempo de existir. La velocidad con que ha crecido y su alcance han hecho que el mercado fije su atención en ella. Desde sus inicios existieron personas en distintos campos que vieron el avance que para ellos representaba hacer uso de software especializado que les permitiera automatizar procesos o acelerarlos.

Al existir tantas demandas en el campo, se comenzaron a realizar investigaciones en la rama del software y del hardware. Con el tiempo los costos se redujeron y el software se convirtió en un negocio rentable. Al haber tanto interés muchas personas empezaron a desarrollar y así surgen las primeras grandes empresas de software. Esto trajo consigo un problema natural en el proceso: al haber tantos desarrolladores en distintos países y para distintas aplicaciones empezó a haber diversidad de estilos además la calidad del producto final variaba mucho entre producto y producto. En este marco se hizo necesario un estándar que permitiera a los consumidores de software saber si el producto que estaban recibiendo era de calidad y si cumplía ciertos requisitos de funcionalidad.

Capítulo 1 Fundamentación Teórica

El software juega un papel muy importante en muchas cosas que se hacen, desde el software que se usa para controlar un avión o un coche, o el software que se usa en la medicina y en la educación, hasta el software de gestión de bancos y empresas. Desde los años 90 la calidad ha sido integrada en el proceso del desarrollo de software con mayor o menor intensidad según la criticidad del producto a obtener.

La historia del software está llena de proyectos fracasados por incumplimiento de plazos y presupuestos, con funcionalidades que no cubren las necesidades de los usuarios. La calidad y la cantidad de información suministrada, así como la claridad con la que se especifiquen los nuevos productos son claves del éxito. El hombre siempre ha tenido interés en obtener resultados satisfactorios, en todo momento se ha buscado la calidad y los bajos costos, sin embargo las estrategias para alcanzarlos se han modificado continuamente debido a las condiciones cambiantes de la sociedad.

El aseguramiento de calidad del software, no es sólo una moda hoy en nuestros días, es una necesidad. También es necesario el uso de prácticas que lleven hacia productos terminados con calidad. Así mismo, se requiere de procesos que ayuden a verificar, validar y controlar que dichas prácticas sean realizadas correctamente. El aseguramiento y mejora de la calidad del software es reconocido cada vez más como un problema fundamental, o quizá como el problema fundamental en cuanto a la ingeniería de software en los años de 1990. (1)

1.3 Situación actual

Uno de los problemas que se afrontan actualmente en la esfera de la computación es la calidad del software. Hoy en día, las empresas que producen software tienen que velar necesariamente por la calidad de su producción debido entre otras cosas a la competitividad que existe en esta rama y a la velocidad de respuesta que es necesario brindar reduciendo los tiempos de desarrollo del mismo para aumentar la productividad y la satisfacción del cliente.

La sociedad moderna depende mucho del software, pues estos facilitan el trabajo de muchos sistemas, ya sea en la salud, la educación, empresas, etc. Todos parecen necesitar más y mejores software más

Capítulo 1 Fundamentación Teórica

rápido y más barato. Muchos proyectos de desarrollo son actualmente tan grandes y complejos que un pequeño grupo de brillantes especialistas no pueden manejarlos solos.

El software actualmente controla la mayoría de los negocios, gobiernos y sistemas militares.

- ❖ Las fábricas son controladas por software.
- ❖ Los productos más avanzados son controlados por software.
- ❖ Las finanzas, la administración y las operaciones de negocios se ejecutan por software.

El costo, cronograma y calidad del software es actualmente el interés crítico de los negocios.

Los proyectos de software, grandes y pequeños, fallan por 4 razones:

- ❖ Los acuerdos del proyecto son irreales.
- ❖ Los grandes proyectos son difíciles de controlar.
- ❖ Los problemas de calidad se vuelven peores con el tamaño del proyecto.
- ❖ Para ser efectivos, los equipos necesitan liderazgo y asesoramiento.

La implantación de sistemas de calidad aporta gran número de beneficios a las compañías que apuestan por esta estrategia. No sólo reducen sus costes de manera razonable, sino que además incrementan sus ingresos gracias al mayor grado de satisfacción de sus clientes y en una mejora de la motivación de sus empleados.

El aseguramiento de la calidad del software es un aspecto muy importante que se debe tener en cuenta durante el desarrollo del software, pero en ocasiones se trata con retraso y con insuficiencia de prioridad y recursos. Si se aplican las actividades de aseguramiento de la calidad del proceso de

Capítulo 1 Fundamentación Teórica

desarrollo del software desde etapas tempranas se reduce la cantidad de trabajo repetitivo a realizar y esto supondrá costes bajos, mejorará el tiempo de llegada al mercado y la satisfacción del cliente. (2)

1.4 Calidad del software en Cuba

El desarrollo de la Industria Nacional de Software es una tarea de gran importancia para Cuba debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país. A pesar de ello, los resultados alcanzados no cubren las expectativas, pues la productividad es baja, el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida. Los proyectos no están en el tiempo requerido.

Hoy en día la industria del software no ha acabado de salir de la fase artesanal debido a la desorganización y la falta de planificación. La disciplina ingeniería del software es relativamente reciente y sus conceptos más importantes están aun inmaduros.

Las organizaciones que tienen poco conocimiento tienden a realizar los procesos de software improvisados y en caso de que se hayan especificado no se siguen rigurosamente. Los planes y presupuestos exceden sistemáticamente, al no estar basados en estimaciones realistas.

1.5 ¿Qué es Calidad?

Existen varias definiciones y concepciones de la calidad, algunas de estas son las siguientes:

- ❖ Un grado predecible de uniformidad y fiabilidad a bajo costo y adecuado a las necesidades del mercado. (W.Eduards Deming (1900-1993))
- ❖ Conjunto total de las características del producto de marketing, ingeniería, fabricación y mantenimiento a través del cual el producto en uso satisface las expectativas del cliente. (Armand Feigenbaum)
- ❖ Dar al cliente aquello que espera. (Trifus)
- ❖ Hacer las cosas bien a la primera. (Price)
- ❖ Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas. (ISO 8402 o UNE 66001-92 o UNE – EN- ISO 9000)

Capítulo 1 Fundamentación Teórica

- ❖ Cumplir con las necesidades y expectativas del cliente, lo que conlleva medir la satisfacción del cliente. Grado en que un conjunto de características inherentes cumple con unos requisitos. (Norma ISO 9000-2000)
- ❖ El American Heritage Dictionary, define la calidad como “una característica o atributo de algo”. Como un atributo de un elemento, la calidad se refiere a las características mensurables – cosas que se pueden comparar con estándares conocidos como longitud, color, propiedades eléctricas, maleabilidad, etc. Sin embargo, el software en su gran extensión, como entidad intelectual, es más difícil de caracterizar que los objetos físicos. No obstante sí existen medidas de características de un programa, tales como: complejidad ciclomática, cohesión, número de puntos de función, líneas de código, y muchas otras.

Se puede llegar a la conclusión que la calidad no es más que satisfacer todas las necesidades del cliente.

1.5.1 ¿Qué es Calidad de Software?

Las definiciones de la calidad del software son varias, a continuación se mencionan alguna de ellas:

- ❖ “Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas” (IEEE 729-83).
- ❖ De acuerdo a la definición del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Std. 610-1990) “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.
- ❖ Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario. (Pressman, 1998)

La calidad del software es importante y para esto se debe:

- ❖ Definir explícitamente lo que significa calidad del software.

Capítulo 1 Fundamentación Teórica

- ❖ Crear un conjunto de actividades que ayuden a garantizar que todo producto de la ingeniería del software presenta una alta calidad.
- ❖ Llevar a cabo actividades de garantía de calidad en cada proyecto de software.
- ❖ Utilizar métricas para desarrollar estrategias que mejoren el proceso del software y, como consecuencia, mejoren la calidad del producto final.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

La definición de calidad en el software propuesta por la organización internacional de estándares (ISO/IEC DEC 9126) es la siguiente: la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas.

Se puede decir que el software tiene calidad si cumple o excede las expectativas del cliente en cuanto a:

1. Funcionalidad (que sirva un propósito).
2. Ejecución (que sea práctico).
3. Confiabilidad (que haga lo que debe).
4. Disponibilidad (que funcione bajo cualquier circunstancia).

Capítulo 1 Fundamentación Teórica

5. Apoyo, a un costo menor o igual al que el cliente está dispuesto a pagar.

Después de haber leído varios conceptos de distintos gurús se puede llegar a la conclusión que la calidad del software no es más que desarrollar un producto o servicio, en este caso un software con la menor cantidad de defectos posibles y que el cliente quede satisfecho con este, luego de haber cumplido con sus expectativas y necesidades.

1.6 Aseguramiento de la Calidad del Software.

El Aseguramiento de la Calidad o SQA por sus siglas en inglés (Software Quality Assurance) es una actividad que se aplica en los proyectos durante todo el proceso de desarrollo del software.

También se puede decir que no es más que las acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio cumpla con los objetivos de la calidad.

El aseguramiento de la calidad del software tiene como finalidad principal determinar si las necesidades de los usuarios están siendo satisfechas adecuadamente. Para determinar si las necesidades de los usuarios están siendo satisfechas, se deben de evaluar tres áreas:

- ❖ **Objetivos:** Los objetivos de la organización son primero, luego vienen los requerimientos del usuario. Los objetivos de cualquier usuario deben de estar en armonía con los objetivos de la organización.
- ❖ **Métodos:** Deben de utilizarse métodos que contengan u observen las políticas, procedimientos y estándares de la organización.
- ❖ **Ejecución:** Optimización del uso de hardware y software al implementar los productos de software.

Para evaluar las áreas expuestas con anterioridad, es necesario que se cuente con un equipo de aseguramiento de calidad que sea efectivo y que tenga un impacto dentro del desarrollo y prueba del producto de software final.

El aseguramiento de la calidad aborda principalmente:

- ❖ Un enfoque de gestión de la calidad.

Capítulo 1 *Fundamentación Teórica*

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

1.6.1 Control de la Calidad

El control de la calidad es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados. (3)

El principal objetivo que se quiere lograr con el control de la calidad consiste en detectar y corregir los errores que surgen durante el proceso de desarrollo de un producto de software, así como garantizar que el producto final responda a las expectativas del cliente.

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición. Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Por ejemplo, John Wiley define métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad.

1.6.2 La Garantía de la Calidad

La garantía de la calidad consiste en la auditoría y las funciones de información de la gestión. El objetivo de la garantía de la calidad es proporcionar la gestión para informar los datos necesarios sobre la calidad del producto, por lo que se va adquiriendo una visión más profunda y segura de que la calidad del producto está cumpliendo sus objetivos. (3)

Capítulo 1 Fundamentación Teórica

También se puede decir que es un conjunto de actividades de planificación, estimación y supervisión de las actividades de desarrollo, que se realizan de forma independiente al equipo de desarrollo, de tal forma que los productos software resultante cumplen los requisitos establecidos.

1.6.3 Gestión de la Calidad

La gestión de la calidad es un conjunto de actividades y recursos necesarios para definir e implantar un sistema de calidad, responsabilizándose de su control, aseguramiento y mejora continua.

Existe una variedad de modelos para la gestión de la calidad del software, a continuación se presentan algunos de ellos:

- ❖ Modelo CMMI
- ❖ Norma ISO/IEC 12007
- ❖ Metrica3
- ❖ ISO 15504

1.7 Plan de Aseguramiento de la Calidad del Software

Para asegurar la calidad de un software hay que proponerse una serie de actividades, entre las cuales se encuentra el establecimiento de un plan de aseguramiento de la calidad.

El plan de aseguramiento de la calidad de un proyecto sirve como guía para el desarrollo e institución de las actividades de aseguramiento de la calidad. Un plan de aseguramiento debe contener por lo general:

- ❖ Objetivos de la calidad del proyecto y su enfoque para su consecución.
- ❖ Documentación referenciada en el plan.
- ❖ Gestión del aseguramiento de la calidad.

Capítulo 1 Fundamentación Teórica

- ❖ Documentación de desarrollo y de control o gestión.
- ❖ Estándares, normas o prácticas que hay que cumplir.
- ❖ Actividades de revisión y auditorías.
- ❖ Gestión de la configuración del software.
- ❖ Informes de problemas
- ❖ Pruebas
- ❖ Herramientas, técnicas y métodos de apoyo.

1.8 Estudio de planes existentes

En el desarrollo de este trabajo se ha realizado un estudio de otros planes existentes, viendo los elementos que se deben tener en cuenta a la hora de realizar una propuesta de un plan de aseguramiento de la calidad y así conformar la propuesta.

1.8.1 Plan de Aseguramiento de la Calidad propuesto por el proyecto Akademos

En el plan propuesto para el aseguramiento de la calidad del proyecto Akademos resaltan las siguientes actividades:

Revisión de las actividades de ingeniería del software

Estas revisiones son del tipo Revisión Técnica Formal (RTF).

Durante el proceso de revisión se decide:

- ❖ Si se acepta el artefacto sin posteriores revisiones.
- ❖ Si se rechaza el artefacto debido a los errores encontrados y es propuesto para una próxima revisión.
- ❖ Si se acepta el artefacto provisionalmente y sin revisiones, pues se encontraron errores menores que serán corregidos.

Capítulo 1 Fundamentación Teórica

Una vez tomada la decisión, se debe establecer la fecha de la próxima revisión, esto queda reflejado en el Acta de Revisión.

Auditorías a los procesos

Política de la auditoría:

1. Se ejecuta el proceso de auditoría a cada uno de los procesos definidos como auditables en el Plan SQA. Estas auditorías se materializan con RTF y verificaciones del cumplimiento de las normas procedimentadas para los procesos.
2. El Auditor emite las conclusiones del proceso de auditoría detallando si el proceso es aprobado o no, en caso de no ser aprobado debe indicarse la fecha de realización de la próxima auditoría.
3. El Auditor debe documentar las desviaciones encontradas en el Acta de Revisión (plantilla), después de finalizada la auditoría, e informar al líder del proyecto.
4. El Auditor actualiza el Registro de Auditorías (plantilla) para de esta forma darles seguimiento.

Medición

Se aplican métricas durante el levantamiento de los requisitos en las RTF

Proceso de Monitoreo y control del proyecto

En esta actividad se lleva un control de la revisión de estado del proyecto, la revisión de estado de los recursos y la revisión de estado de los conocimientos y habilidades del personal.

Auditoría al proceso de Aseguramiento de la calidad

El líder de proyecto y una comisión ejecuta la auditoría al proceso de aseguramiento de la calidad según las normas procedimentadas.

Capítulo 1 Fundamentación Teórica

1.8.2 Plan de Aseguramiento de la Calidad propuesto por la IEEE 730-1998

Propósito

Describe brevemente el propósito del plan.

Alcance

Se refiere al proyecto que estará involucrado el plan.

Definiciones, Acrónimos y Abreviaturas

Definiciones, acrónimos y abreviaturas utilizadas en el documento.

Resumen

Se realiza un resumen de los principales aspectos del plan.

Objetivos de Calidad

Se incluyen los requerimientos del proyecto que están alineados con los requerimientos de calidad.

Organización

Se describe la estructura de la organización. Se especifican cada uno de los responsables de la calidad.

Tareas y responsabilidades

Se proponen las actividades a realizar durante todo el proceso de aseguramiento de la calidad.

Capítulo 1 Fundamentación Teórica

Documentación

Lista de los documentos utilizados por el Plan de Calidad.

Métricas

En esta sección se describe todo el proceso de métricas que se realizará como producto del monitoreo del trabajo.

Estándares y guías

Lista de los estándares y guías utilizados por el Plan de Calidad.

Tareas de revisiones y auditorías

Describe brevemente cada tipo de revisión y auditoría que se llevará a cabo en el proyecto. Para cada tipo, identifique los artefactos del proyecto que serán el asunto de la revisión o auditoría. Estos pueden incluir Revisiones Técnicas y de Gestión Conjuntas entre Cliente y Desarrollador, Revisiones y Auditorías de Proceso, Auditorías de Cliente, Revisiones Internas, Técnicas y de Gestión.

Cronograma

Detalle aquí el cronograma para las revisiones y auditorías. Este debe incluir las revisiones y auditorías programadas en las fechas principales del proyecto, así como revisiones que son provocadas por la entrega de artefactos del proyecto.

Organización y Responsabilidades

Liste aquí los grupos específicos o individuos a ser involucrados en cada una de las actividades de revisión y de auditoría identificadas. Describa brevemente las tareas y responsabilidades de cada uno. También, liste cualquier agencia externa que se espera que apruebe o regule cualquier producto del proyecto.

Capítulo 1 Fundamentación Teórica

Resolución de problemas y actividades de corrección

Esta subsección describe los procedimientos para informar y manejar problemas identificados durante las revisiones y auditorías del proyecto. El Plan de Resolución de Problemas puede ser referenciado.

Herramientas, técnicas y metodologías utilizadas en el plan de Revisiones y Auditorías

Describa aquí las herramientas, técnicas o metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y de auditoría identificadas en este plan. Usted debe describir el proceso explícito a ser seguido para cada tipo de revisión o auditoría. Su organización puede tener un Manual de Procedimientos de Revisión y de Auditoría estándar que puede ser referenciado. Estas descripciones de los procedimientos también deben dirigir la recolección, almacenamiento y archivado de los Registros de Revisión del proyecto. Deben describirse las listas de chequeo a utilizar en cada revisión y los atributos de calidad que serán abordados en cada una de ellas.

Pruebas y Evaluación

Se hace referencia al plan de pruebas.

Herramientas, Técnicas y Metodologías

Lista de todas las herramientas, técnicas y metodologías utilizadas en las actividades del Plan de Calidad.

Resolución de Problema y Acción Correctiva

Esta sección referencia el Plan de Resolución de Problema.

Capítulo 1 Fundamentación Teórica

Gestión de Configuración

Referencia al Plan de Gestión de Configuración.

Registros de Calidad

Descripciones de varios registros de calidad que se mantendrán durante el proyecto, incluyendo cómo y dónde cada tipo de registro se guardará y por cuánto tiempo.

Entrenamiento

Listado de las actividades de entrenamiento necesario para que el equipo de proyecto ejecute las actividades del Plan de Aseguramiento de la Calidad.

1.9 Estándares

Durante el desarrollo de proceso de aseguramiento de la calidad y todo el ciclo de vida se utilizaron estándares.

ISO (International Standard Organization) ha aportado estándares para la industria del software. Algunos de los más importantes son:

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

También existen estándares para la administración de proyectos los cuales garantizan en gran medida obtener productos de software con calidad, algunos de ellos son los siguientes:

Capítulo 1 Fundamentación Teórica

- ❖ CMMI Capability Maturity Model Integration. Modelo que permite catalogar a las organizaciones con el nivel de capacidad de madurez de su proceso de desarrollo.
- ❖ TSP Team Software Process. Permite calificar el proceso de desarrollo que se lleva a cabo en los equipos de trabajo o desarrolladores.
- ❖ PSP Personal Software Process. Certifica a un individuo o desarrollador en el nivel de madurez de su proceso de desarrollo.
- ❖ SPICE Software Process Improvement and Capability Determination. Se conforma como el estándar emergente orientado a la mejora continua del proceso de desarrollo de software.
- ❖ Modelo de Boehm. Modelo de descomposición de características de calidad entre niveles antes de aplicar las métricas: usos principales, componentes intermedios y componentes primitivos.
- ❖ Paradigma GQM (goalquestionmetric): enfoque de medición para evaluar la calidad del software basado en la identificación de los objetivos a lograr.
- ❖ Modelo Gilb. Creación de una especificación de requisitos de calidad para cada proyecto que deben describir conjuntamente el usuario y el analista.

1.9.1 ISO 9001

La Norma ISO 9001 tiene como objetivo satisfacer al consumidor, es un método de trabajo, que se considera tan bueno, que es el mejor para mejorar la calidad y satisfacción de cara al consumidor. (4)

1.9.2 ¿Qué es CMMI?

El CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software. Este modelo surgió a partir de CMM (Modelo de Capacidad y Madurez) como resultado de la integración de varios modelos que definían la madurez en diferentes disciplinas, pero que dada la variedad constituían un problema para las empresas por lo que fue necesario agruparlos en el CMMI.

CMMI dirige su enfoque a la mejora de procesos en una organización, los estudia y mide la capacidad para construir un software de calidad, atendiendo a una escala de cinco niveles (inicial, repetible, definido, dirigido y optimizado). Para que cada organización pueda enfocar la mejora de sus procesos

Capítulo 1 *Fundamentación Teórica*

se especifica en cada nivel de madurez un conjunto de áreas de proceso, que se describen en términos de prácticas, estas son un conjunto de actividades que contribuyen a la implementación eficiente de un área de proceso, si se cumplen todas las prácticas y se satisfacen todas las áreas de proceso de un determinado nivel entonces la empresa podrá certificarse en ese nivel de madurez.

Inicial o Nivel 1 CMMI: este es el nivel donde están todas las empresas que no tienen procesos. Los presupuestos se disparan, no es posible entregar el proyecto en fechas, los empleados si tienen que quedar durante noches y fines de semana para terminar un proyecto. No hay control sobre el estado del proyecto, el desarrollo del proyecto es completamente opaco, no se sabe que pasará con él.

Nivel 2 CMMI: quiere decir que el éxito de los resultados obtenidos se puede repetir. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Nivel 3 CMMI: alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida, definida quiere decir que está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos.

Nivel 4 CMMI: los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Nivel 5 CMMI: los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

En CMMI como en otros modelos de Calidad se tiene en cuenta las mediciones como datos necesarios para el cálculo de métricas. Estas deben ser recepcionadas por cada desarrollador de software en cualquier proyecto, o en general por el equipo de trabajo. En el mundo de las tecnologías de la información se han definido procesos que apoyen el trabajo individual y en equipo para el desarrollo de cualquier proyecto de software. Un ejemplo de ello son los procesos definidos por Watts S. Humphrey conocidos como: Proceso Personal de Software y el Proceso de Software del Equipo.

Capítulo 1 Fundamentación Teórica

Se puede decir que el CMMI es un modelo aceptado y orientado al desarrollo de productos de software, por lo que es el modelo que se ha tenido en cuenta para llevar a cabo el aseguramiento de la calidad.

1.9.3 Proceso Personal de Software (PSP)

El PSP (Personal Software Process) es un proceso de auto-mejora que ayuda a controlar, administrar y mejorar la manera en que se trabaja. Es un marco de trabajo estructurado de formularios, guías y procedimientos para el desarrollo de software. Si es usado de forma apropiada, brinda los datos necesarios para cumplir los acuerdos y hace los elementos rutinarios de su trabajo más predecible y eficiente. (5)

PSP define un proceso de mejora que contiene una serie de pasos a seguir por el ingeniero de manera que pueda retroalimentarse teniendo en cuenta el objetivo que persigue y de que forma puede cambiar su proceso para lograr los resultados esperados. En este sentido toma un papel fundamental la medición, pues ayuda a diagnosticar el problema y una vez tomadas las medidas pertinentes, valorar la mejora de los resultados.

El PSP es un prerrequisito del planeamiento de la organización para producir el TSP. (6)

1.9.4 Proceso de Software del Equipo (TSP)

TSP es un conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo.

El TSP es un proceso para construir y guiar a los equipos y ayudarlos a realizar bien su trabajo mediante la aplicación de buenas prácticas de ingeniería de software en cada fase del ciclo de desarrollo y ayudándolos a controlar, administrar y mejorar los resultados del equipo. Se basa en los principios del PSP. (6). El TSP tiene entre sus objetivos ayudar a los miembros de un equipo a establecer relaciones de trabajo, determinar los roles, ponerse de acuerdo en las metas del equipo y las particulares de cada uno de sus miembros, ser cooperativos y lograr un trabajo personal

Capítulo 1 *Fundamentación Teórica*

disciplinado, de forma tal que puedan planificar y dar seguimiento a sus tareas sobre la base de producir productos con calidad, siempre orientados a alcanzar las metas del equipo.

1.9.5 ISO

La Organización Internacional para la Estandarización, mejor conocida como ISO, es la agencia especializada en estandarización, conformada por representantes de los cuerpos normalizadores, fue establecida oficialmente el 23 de febrero de 1947 con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional de bienes y servicios así como el desarrollo científico y tecnológico. Actualmente abarca los estándares nacionales de 91 países. En los Estados Unidos, la representación se llama The American National Standards Institute (ANSI).

1.10 Métricas de Software

Las métricas son un buen medio para entender, monitorizar, controlar y predecir el desarrollo de software. El glosario de términos de estándares de IEEE define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (2)

Para poder asegurar que un proceso o sus productos resultantes son de calidad o poder compararlos, es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha comparación.

- ❖ Las mediciones del software proporcionan un mecanismo para la evaluación objetiva.
- ❖ Pueden ser aplicadas sobre: el proceso, el proyecto o el producto final.
- ❖ Las mediciones permiten:
 - ❖ Caracterizar y comparar con evaluaciones futuras.
 - ❖ Evaluar y controlar el avance del proyecto.

Capítulo 1 Fundamentación Teórica

- ❖ Predecir, planificar y estimar en base a datos históricos.
- ❖ Mejorar la calidad del producto.
- ❖ **Medida:** indicación cuantitativa de extensión, cantidad, capacidad o tamaño de algunos atributos de un proceso o producto.
- ❖ **Medición:** acto que determina una medida.
- ❖ **Métrica:** es una medida cuantitativa del grado en que un sistema, componente, o proceso posee un atributo dado.
- ❖ **Indicador:** métrica o combinación de métricas que proporciona una visión profunda del proceso, del proyecto o del producto.

Existen diferentes clasificaciones para las métricas, a continuación se mencionan algunas de ellas:

- ❖ Métricas técnicas.
- ❖ Métricas de calidad.
- ❖ Métricas de productividad.
- ❖ Métricas orientadas a la persona.
- ❖ Métricas orientadas al tamaño.
- ❖ Métricas orientadas a la función.

Para el desarrollo de esta investigación la métrica más utilizada es la métrica de calidad, que es la encargada de proporcionar una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir, cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

Las métricas permiten que los desarrolladores y líderes del proyecto mejoren el proceso de software, planificación y evalúen la calidad del producto. Así de esta forma se logra que el cliente quede satisfecho.

Capítulo 1 *Fundamentación Teórica*

Se han propuesto cientos de métricas para el software, pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad. (3)

Ejiogu [Eji, 1991] define un conjunto de atributos que deberían acompañar a las métricas efectivas del software. Las métricas obtenidas y las medidas que conducen a ello deberían ser:

Simple y fáciles de calcular: debería ser relativamente fácil aprender a obtener la métrica y su cálculo no debería demandar un esfuerzo o cantidad de tiempo inusuales.

Empírica e intuitivamente persuasivas: las métricas deberían satisfacer las nociones intuitivas del ingeniero sobre el atributo del producto.

Consistentes y objetivas: las métricas deberían siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información de software.

Consistentes en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería emplear medidas que no conduzcan a extrañas combinaciones de unidades. Independiente del lenguaje de programación: las métricas deberían basarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa.

Un eficaz mecanismo para la realimentación de calidad: la métrica debería proporcionar, al desarrollador de software, información que le lleve a un producto final de mayor calidad.

1.11 Factores que determinan la calidad del software

Un software de alta calidad es una de las metas más importantes. La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según los clientes que las necesiten.

Capítulo 1 Fundamentación Teórica

Según Pressman los factores que afectan a la calidad del software se pueden categorizar en dos amplios grupos:

- ❖ Factores que se pueden medir directamente (por ejemplo, defectos).
- ❖ Factores que se pueden medir solo indirectamente (por ejemplo, facilidad de uso o de mantenimiento).

1.11.1 Factores de calidad

Factores de McCall

McCall y sus colegas propusieron una útil clasificación de factores que afectan la calidad del software. Estos factores de calidad se concentran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad de cambios y su adaptabilidad a nuevos entornos. Cada una de estas características tiene un conjunto de factores que las describen. Los factores de calidad descritos por McCall representan solo una de las muchas listas de comprobación sugeridas para la calidad del software. (3)

Características operativas:

- ❖ Corrección: Hasta donde satisface un programa su especificación y logra los objetivos expuestos por el cliente.
- ❖ Fiabilidad: Hasta donde se puede esperar que un programa lleve a cabo su función con la exactitud requerida.
- ❖ Eficiencia: La cantidad de recursos informáticos y de códigos necesarios para que un programa realice su función.
- ❖ Integridad: Hasta donde se puede controlar el acceso al software o a los datos por personas no autorizadas.
- ❖ Usabilidad (Facilidad de manejo): El esfuerzo necesario para aprender a operar con el sistema, preparar los datos de entrada e interpretar las salidas (resultados) de un programa.
- ❖ Facilidad de mantenimiento: El esfuerzo necesario para localizar y arreglar un error en un programa.

Capítulo 1 Fundamentación Teórica

- ❖ Flexibilidad: El esfuerzo necesario para modificar un programa que ya está en funcionamiento.
- ❖ Facilidad de pruebas: El esfuerzo necesario para probar un programa y asegurar de que realice correctamente su función.
- ❖ Adaptabilidad a nuevos entornos.
- ❖ Portabilidad: El esfuerzo necesario para transferir el programa de un entorno a otro entorno diferente.
- ❖ Reusabilidad (Capacidad de reutilización): Hasta donde se puede volver a emplear un programa.
- ❖ Interoperabilidad: El esfuerzo necesario para acoplar un sistema con otro.

Factores FURPS

Existe un conjunto de factores a los cuales se les ha dado el acrónimo de FURPS: funcionalidad, facilidad de uso, fiabilidad, rendimiento y capacidad de soporte. Estos factores pueden usarse para establecer métricas de calidad para todas las actividades del proceso del software. (3)

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

Capítulo 1 Fundamentación Teórica

Factores de Calidad ISO 9126

Este estándar identifica seis atributos clave de calidad, estos son los siguientes:

Funcionalidad: El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.

Confiabilidad: Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.

Usabilidad: Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.

Eficiencia: Grado en que el software hace óptimo el uso de los recursos del sistema. Viene reflejado por los siguientes subatributos: tiempo de uso y recursos utilizados.

Facilidad de mantenimiento: La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Portabilidad: La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

1.12 Verificación y validación

Se puede definir la verificación y la validación de los procesos de desarrollo de software como:

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

Capítulo 1 Fundamentación Teórica

Se pueden incluir las pruebas y procesos de revisión y auditoría dentro de las actividades de verificación y validación ligadas al control de la calidad.

Las revisiones según el estándar ANSI/IEEE Std 10281988 se define como: evaluación de un elemento para determinar diferencias con los resultados planeados y recomendar mejoras.

Durante el proceso de desarrollo del software se pueden llevar a cabo diferentes tipos de revisiones, algunas de estas son:

- ❖ Reuniones Informales.
- ❖ Presentación formal.
- ❖ Revisiones técnicas formales.
- ❖ Las revisiones Pass Around.

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

1.12.1 Pruebas

Las pruebas son actividades en las cuales un sistema o sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan, registran y se realiza una evaluación de algún aspecto.

Existen varios tipos de pruebas que pueden realizarse en el transcurso del desarrollo del proceso de software, algunas de estas son:

- ❖ Unitarias: Pretenden probar cada función en un archivo de programa simple (una clase en terminología de objetos).
- ❖ Integración: Pretenden comprobar la integración de los componentes, es decir, la comunicación a través de interfaces, acceso incoherente a estructuras de datos globales. Pueden realizarse de forma ascendente o descendente

Capítulo 1 Fundamentación Teórica

- ❖ Validación: Pretende comprobar que se satisfacen los requisitos.
- ❖ Sistema: Se centran en comprobar la recuperación, seguridad, resistencia, rendimiento.

También existen técnicas de pruebas:

Pruebas de caja blanca: Se centra en comprobar la interacción interna de los componentes del sistema.

Pruebas de caja negra: Se centra en los requisitos funcionales del software, y se lleva a cabo sobre la interfaz del software.

1.13 Gestión de configuración del software.

La Gestión de Configuración del Software (GCS) se puede definir como una disciplina cuya misión es controlar la evolución de un sistema software. (7)

Según Babich, “El arte de coordinar el desarrollo de software para minimizar la confusión, se denomina Gestión de Configuración. La Gestión de Configuración es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.”

¿Qué papel juega la Gestión de Configuración dentro de un Proyecto de Desarrollo de Software? El éxito de un proyecto depende de la correcta ejecución de 4 tipos de funciones:

- ❖ Gestión del Proyecto, que incluye fundamentalmente la estimación, planificación y seguimiento del proyecto, y la organización, dirección y gestión de recursos humanos.
- ❖ Desarrollo Técnico: Actividades de Ingeniería del Software a lo largo de todo el ciclo de vida del producto (Análisis, Diseño, Codificación).
- ❖ Sistema de Calidad, que incluye las actividades de Validación (¿Se está construyendo el producto correcto?), Verificación (¿Se está construyendo el producto correctamente?) y Pruebas (¿Funciona el código?) y las actividades de Garantía de Calidad (Medidas encaminadas a asegurar que el producto se construye con unos determinados niveles de calidad).

Capítulo 1 Fundamentación Teórica

- ❖ Sistema de Gestión de Configuración, aunque también se suele considerar a la Gestión de Configuración como una actividad de Garantía de Calidad.

El objetivo de las actividades de Gestión de Configuración del Software es:

- ❖ Establecer y mantener la integridad de los productos generados durante un proyecto de desarrollo de software y a lo largo de todo el ciclo de vida del producto.
- ❖ Evaluar y controlar los cambios sobre ellos, es decir, controlar la evolución del sistema software.
- ❖ Facilitar la visibilidad sobre el producto.

1.14 Conclusiones

En este capítulo se abordaron diferentes conceptos de expertos en la calidad del software, temas relacionados con la calidad, y se hizo un estudio de otros planes de aseguramiento de la calidad existentes, los cuales son de mucha importancia para el desarrollo de este Trabajo Diploma.

Capítulo 2 Propuesta de Solución

Capítulo 2 Propuesta de Solución

2.1 Introducción

En este capítulo se abordaran temas que darán paso al desarrollo del presente trabajo de diploma. Se hace la propuesta de un plan para asegurar la calidad del proyecto Servicios Comunitarios.

Para dar cumplimiento a los objetivos específicos, se realizó la siguiente actividad:

Propuesta de Plan de Aseguramiento de la Calidad del Software (PSQA) con el objetivo de proporcionar un mapa para institucionalizar la garantía de calidad del software, se utiliza como plantilla para actividades de aseguramiento y garantía de la calidad instituida para el proyecto.

2.2 Descripción del proyecto productivo Servicios Comunitarios.

El propósito del proyecto productivo Servicios Comunitarios es automatizar una serie de servicios que se prestan en la Universidad de las Ciencias Informáticas, los cuales son de gran importancia para todos los trabajadores, estudiantes y profesores de la misma. Algunos de estos servicios son la reservación de transporte y reservación de gas. Con la automatización de estos servicios se hace más fácil el control de los mismo, ya que para reservar transporte, por parte de los estudiantes o profesores, depende del módulo que sea, simplemente lo pueden hacer a través de una aplicación, al igual que con la reservación de la balita del gas, así se lleva un mejor control de los servicios, y es mas fácil para el cliente como para el que presta el servicio.

El proyecto está constituido por varios módulos:

- ❖ Sistema para la gestión de los procesos del gas.
- ❖ Transportación Nacional.
- ❖ Transportación para trabajadores internos.

Capítulo 2 Propuesta de Solución

- ❖ Transportación Estudiantil.
- ❖ Trompo UCI.
- ❖ Sistema financiero presupuestario de la dirección de los servicios generales.

2.3 Proceso de Aseguramiento de la Calidad del Software en el Proyecto Servicios Comunitarios.

El Proceso de Aseguramiento de la Calidad del Software en el proyecto Servicios Comunitarios comienza con la confección del Plan de Aseguramiento de la Calidad del Software. El propósito de la propuesta de este plan es describir paso a paso como se va a realizar el aseguramiento de la calidad de los productos, los artefactos y los procesos que se generan durante el desarrollo de cada uno de los módulos del proyecto Servicios Comunitarios.

2.4 ¿Qué objetivo se persigue con la propuesta del Plan de Aseguramiento de la Calidad del Software?

El objetivo que se persigue con la propuesta del Plan de Aseguramiento de la Calidad del Software (PSQA), es presentar un modelo para sistematizar y planificar todas las acciones necesarias para proveer la confianza adecuada según los requerimientos técnicos establecidos por el proyecto, garantizando la calidad del mismo. Con este plan no se pretende otra cosa que el aseguramiento del producto de software que se está desarrollando.

2.5 Necesidad del establecimiento de un Plan de Aseguramiento de la Calidad del Software.

Cuando se refiere a calidad se habla de obtener un producto que satisfaga las necesidades del cliente, y cumpla con sus requisitos, es decir todas las cosas buenas que le gustaría ver en el producto. Si se construye un producto y se asegura su calidad, el cliente queda satisfecho, para lograrlo hay que trazar una serie de actividades para el Aseguramiento de la Calidad del Software (SQA), como son las pruebas, uso de métricas, estándares, guías, metodologías, y una de las más importantes, el establecimiento de un Plan de Aseguramiento de la Calidad del Software(PSQA), que es de gran importancia y necesario pues en él se especifican los objetivos de calidad, se seleccionan

Capítulo 2 Propuesta de Solución

estrategias para asegurar que las metas se han logrado, y se detalla un plan de acción para llevar a cabo las estrategias. Mediante este plan se seleccionan y se coordinan bien las actividades para el Aseguramiento de la Calidad del Software.

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

El Plan de Aseguramiento de la Calidad del Software (PSQA), es un artefacto que provee una visualización clara de cómo debe ser seguro un producto, artefactos, y la calidad del proceso. Contiene el plan de revisiones y auditorías. Es mantenido durante todo el proceso de desarrollo del software.

El plan de aseguramiento de calidad sirve como guía para el desarrollo e institución de las actividades de aseguramiento de la calidad. Para realizar esta propuesta sirvió de guía el plan propuesto por la IEEE 730-1998, que a la vez es el que propuso la Dirección de Calidad del Software (DCS).

Los elementos que conformarán este plan son los siguientes:

- ❖ Introducción
- ❖ Objetivos de Calidad
- ❖ Gestión del aseguramiento de la calidad.
- ❖ Documentación de desarrollo y de control o gestión.
- ❖ Métricas
- ❖ Estándares y guías.
- ❖ Plan de revisión y auditorías.
- ❖ Pruebas y Evaluación
- ❖ Herramientas, técnicas y metodologías
- ❖ Resolución de Problema y Acción Correctiva.
- ❖ Gestión de la configuración del software.
- ❖ Registros de Calidad
- ❖ Entrenamiento

Este plan ha sido adaptado a las condiciones específicas del proyecto Servicios Comunitarios. Por tanto, su uso es particularmente para los miembros del proyecto.

Capítulo 2 Propuesta de Solución

Con la propuesta de este plan se tiene el propósito de describir como asegurar la calidad de los artefactos, y de los productos que se generan en el proyecto Servicios Comunitarios, para lograr la satisfacción del cliente.

2.6.1 Objetivos de la Calidad

Los objetivos de la calidad establecidos durante el desarrollo del proyecto Servicios Comunitarios y sus diferentes módulos están alineados con los requerimientos de calidad expuestos por la norma cubana ISO 9126.

- ❖ Asegurar que las funciones de los diferentes sistemas del proyecto Servicios Comunitarios satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas.
- ❖ Asegurar que el producto de software sea comprendido, aprendido, utilizado y atractivo para el usuario.
- ❖ Lograr que el software sea idóneo, y pueda usarse para las tareas y condiciones de uso particulares.
- ❖ Asegurar que el software sea amigable y atractivo para el usuario.
- ❖ Proporcionar un software con una ejecución y un desempeño apropiado.

- ❖ Garantizar que los diferentes sistemas cumplan con las necesidades de los usuarios, y sean comprensibles.

- ❖ Lograr que los usuarios se sientan satisfecho con la interfaz de usuario.

- ❖ Garantizar que el funcionamiento de los sistemas sea comprendido por futuros desarrolladores.

- ❖ Los sistemas pueden ser fácilmente modificados y extendidos en el futuro.

- ❖ Los sistemas deben permitir ser sólo usados por usuarios autorizados y restringir el uso basado en permisos.

Capítulo 2 Propuesta de Solución

- ❖ Garantizar que los sistemas puedan ser ejecutados en las disímiles plataformas existentes en la universidad.
- ❖ Lograr que el equipo de calidad cuente con el personal capacitado con el conocimiento y las habilidades necesarias para realizar las tareas y actividades encaminadas a lograr la calidad del proyecto.

2.6.2 Organización del equipo de Aseguramiento de la Calidad

El proyecto Servicios Comunitarios está compuesto por un conjunto de profesores y estudiantes de la facultad 1. El equipo de aseguramiento de la calidad debe estar conformado por los siguientes roles:

Administrador de la calidad: comienza su trabajo con la confección del plan de aseguramiento de la calidad de software, cerciorándose de que este se ajuste a los estándares y mecanismos de la calidad del software, además de monitorear las actividades referentes a la misma. Paralelamente y como soporte al plan de calidad realiza un plan de revisiones y auditorías, así como la realización de las mismas para controlar las actividades de ingeniería de software del proyecto, teniendo en cuenta las herramientas (Listas de Chequeo), técnicas (RTF) y metodologías (RUP) utilizadas. Otra de las actividades que complementan y apoyan el plan de aseguramiento de la calidad del software es la definición de las métricas de calidad, así como la coordinación y el control de cambios efectuados en el proyecto. Es el responsable del flujo de trabajo (FT) de Pruebas. Asegura que la aplicación producida se ajusta a las especificaciones y está razonablemente libre de errores. Proporciona una metodología para realizar las pruebas. Coordina las pruebas de calidad internas, las pruebas de aceptación del cliente y pilotos. Evalúa los resultados que se obtienen en las pruebas de calidad.

Diseñador de pruebas: Diseña los casos de pruebas a realizar. Evalúa y documenta el resultado de las pruebas realizadas al software.

Revisor técnico: Señala la necesidad de mejoras en el producto, confirma las partes del producto en las que no es necesaria una mejora, consiguiendo un trabajo de mayor calidad. Describe errores en la función, la lógica o la implementación de cualquier representación de los sistemas de información. Verifica que los sistemas bajo revisión alcancen sus requisitos. Garantiza que los sistemas han sido

Capítulo 2 Propuesta de Solución

representados de acuerdo con ciertos estándares predefinidos. Consigue un sistema desarrollado en forma uniforme. Hace que los proyectos sean más manejables.

Probador: Ejecuta las pruebas diseñadas y anota los resultados obtenidos.

Es de suma importancia que en cada proyecto existente haya un equipo con una buena preparación dedicada a asegurar la calidad del producto en cada una de sus fases, para que el producto final salga lo mejor posible.

2.6.2.1 Tareas y responsabilidades

Durante el proceso de Aseguramiento de la Calidad se realizarán unas series de tareas, las cuales se recomienda realizar a cada uno de los módulos del proyecto.

La siguiente tabla muestra las tareas propuestas.

Tareas de Aseguramiento de la Calidad	Responsable
Establecimiento del Plan de Aseguramiento de la Calidad.	Equipo de aseguramiento de la calidad
Documentar el sistema (Expediente de Proyecto).	Equipo de aseguramiento de la calidad
Revisión de la captura de requisitos.	Equipo de aseguramiento de la calidad

Capítulo 2 Propuesta de Solución

Revisión de los casos de usos del negocio.	Equipo de aseguramiento de la calidad.
Revisión de los casos de usos del sistema.	Equipo de aseguramiento de la calidad
Revisión del análisis.	Equipo de aseguramiento de la calidad.
Revisión del diseño.	Equipo de aseguramiento de la calidad.
Auditorías a las bases de datos	Equipo de aseguramiento de la calidad
Revisión a la implementación.	Equipo de aseguramiento de la calidad
Revisiones a la Arquitectura.	Equipo de aseguramiento de la calidad
Definir estrategia de prueba para el sistema	Equipo de aseguramiento de la calidad
Definir el plan de prueba	Equipo de aseguramiento de la calidad

Capítulo 2 Propuesta de Solución

Diseñar los casos de prueba.	Equipo de aseguramiento de la calidad
Pruebas de caja blanca	Equipo de aseguramiento de la calidad
Revisiones a la configuración del sistema	Equipo de aseguramiento de la calidad
Pruebas de caja negra	Equipo de aseguramiento de la calidad
Pruebas de Integración.	Equipo de aseguramiento de la calidad
Pruebas de Seguridad	Equipo de aseguramiento de la calidad
Pruebas de Liberación	La facultad

Tabla 1 Tareas y responsabilidades del equipo SQA

2.6.3 Métricas propuestas

Las métricas de software se pueden definir como: “La continua aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa significativa y oportuna, junto con el uso de esas técnicas para mejorar el proceso y sus productos

Capítulo 2 Propuesta de Solución

Existen varias métricas del software la cuales están relacionadas con el desarrollo del software como funcionalidad, complejidad, eficiencia.

Las métricas de calidad son las que proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que el sistema se adapte a los requisitos que pide el cliente.

Su objetivo es ayudar a mejorar las áreas dentro del proceso que requieran mejoras, así como la calidad y oportunidad de los productos que se realizan.

Para lograr el aseguramiento de la calidad del proyecto Servicios Comunitarios se propone utilizar una serie de métricas.

Las métricas externas de funcionalidad deben ser capaces de medir un atributo como es el comportamiento funcional del sistema en el cual el software está presente.

2.6.3.1 Métricas externas de funcionalidad

Dentro de las métricas externas de funcionalidad se propone utilizar en el proyecto Servicios Comunitarios para el aseguramiento de la calidad del mismo las métricas de idoneidad, las de precisión y las de madurez.

Métricas de idoneidad

Las métricas de idoneidad deben ser capaces de medir un atributo como es la ocurrencia de un funcionamiento insatisfactorio o la ocurrencia de una operación insatisfactoria.

Un funcionamiento u operación insatisfactoria puede ser:

- ❖ Funcionamiento u operación que no se desempeña de la forma especificada en el Manual de usuario o la especificación de requisitos.
- ❖ Funcionamiento u operación que no provee una salida aceptable o razonable al tomar en

Capítulo 2 Propuesta de Solución

consideración un objetivo específico de las tareas del usuario.

Existen varias métricas de idoneidad, ejemplo de estas son las que a continuación se describen:

1-Adecuación funcional.

La métrica se propone medir cuán adecuada es la función evaluada. El método de aplicación es el número de funciones idóneas para ejecutar funciones específicas en comparación con el número de funciones evaluadas.

Para el cálculo de esta métrica se usa la siguiente formula:

$$X = 1 - A/B$$

Donde A es el número de funciones en las cuales se detectaron problemas en la evaluación y B el número de funciones evaluadas. El valor obtenido debe estar entre $0 \leq X \leq 1$, a mayor cercanía al 1 más adecuada es la función.

Se propone el uso de esta métrica porque es de gran importancia conocer cuán adecuado es el sistema que se está implementando

2- Plenitud de la implementación funcional.

¿Cuán completa ha sido la implementación y su conformidad con la especificación de requisitos?

Para dar respuesta a esta interrogante se debe ejecutar las pruebas de caja negra funcionales de acuerdo con la especificación de requisitos, contando el número de funciones perdidas detectadas y comparando el resultado con el número de funciones descritas en la especificación de requisitos.

La fórmula que se debe utilizar es:

$$X = 1 - A/B$$

Capítulo 2 Propuesta de Solución

Donde A es el número de funciones perdidas detectadas en la evaluación y B es el número de funciones descritas en especificación de requisitos. El valor obtenido debe estar entre $0 \leq X \leq 1$, a mayor cercanía al 1 resultará mejor.

El uso de esta métrica se propone para saber si el sistema implementado cumple con los requisitos especificados.

Métricas de precisión

Las métricas externas de precisión deben ser capaces de medir un atributo como es la frecuencia con que los usuarios se encuentren con la ocurrencia de una falta de exactitud o de precisión, como pueden ser:

- ❖ Resultados incorrectos o imprecisos causados por datos inadecuados; por ejemplo, un dato con pocos dígitos significativos para un cálculo de precisión;
- ❖ Inconsistencia entre el procedimiento de operación actual y el descrito en el manual de operación;
- ❖ Diferencias entre el resultado actual y el razonablemente esperado producto de una tarea ejecutada durante la operación.

La exactitud esperada es una métrica de precisión que permite conocer si existen diferencias entre los resultados actuales y los razonablemente esperados. Se aplica ejecutando los casos de pruebas de entrada versus salida y comparando los resultados actuales y los razonablemente esperados, contando el número de casos encontrados con diferencias inaceptables en relación con los resultados razonablemente esperados.

La fórmula empleada es:

$$X = A/T$$

Donde A es el número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible y T es el tiempo de operación. El valor

Capítulo 2 Propuesta de Solución

debe estar en el rango $0 \leq X$ y a mayor cercanía al 0 resultará mejor.

Se propone el uso de esta métrica para saber si el sistema obtenido es el mismo que se esperaba, los datos de esta métrica externa deben obtenerse de información asequible externamente, de usuarios, clientes o serviciadores.

Métricas de madurez.

Las métricas externas de madurez deben ser capaces de medir un atributo como la exención de fallos totales en el software, causados por la ocurrencia de fallos existentes en el propio software.

1-Latencia estimada de la intensidad de fallos.

¿Cuántos problemas aún existen que pueden emerger como futuros fallos?

Cuente el número de fallos detectados durante el período definido de pruebas y pronostique el número potencial de futuros fallos utilizando un modelo de estimación de la fiabilidad.

Para esta métrica se utiliza la siguiente fórmula:

$$X = (ABS (A1- A2)) / B$$

Donde ABS es el valor absoluto, A1 el número total de fallos latentes predecibles en el producto de software, A2 el número total de fallos detectados realmente y B el tamaño del producto.

El resultado debe ser $0 \leq X$, en dependencia del estado de las pruebas, en etapas más avanzadas, mientras más pequeño, mejor.

2- Intensidad de fallos totales contra casos de prueba.

¿Cuántos fallos totales fueron detectados durante un período de pruebas definido?

Para dar respuesta a esta interrogante se cuenta el número de fallos totales detectados y el número de casos de pruebas.

Capítulo 2 Propuesta de Solución

Se aplica la siguiente fórmula:

$$X = A1 / A2$$

Donde A1 es el número total de fallos totales detectados y A2 el número de casos de pruebas ejecutados.

El resultado debe ser $0 \leq X$, en dependencia del estado de las pruebas, en etapas más avanzadas, mientras más pequeño, mejor.

3- Grado de solución ante fallos totales.

¿Cuántas condiciones de fallo total están resueltas?

Cuente el número de fallos totales que no se repitieron en determinado período de pruebas bajo condiciones similares. Mantenga un reporte de solución de problemas describiendo la situación de todos los fallos totales.

Utilice la siguiente fórmula:

$$X = A1 / A2$$

Donde A1 es el número de fallos totales solucionados y A2 el número total de problemas reales detectados.

El valor debe estar entre $0 \leq X \leq 1$, a mayor cercanía al 1 resultará mejor, cuanto más fallos totales estén resueltos.

4- Intensidad de fallos.

¿Cuántos fallos fueron detectados durante un período de pruebas definido?

Cuente el número de fallos detectados y compare su intensidad.

Capítulo 2 Propuesta de Solución

La fórmula utilizada es:

$$X = A / B$$

Donde A es el número total de fallos detectados y B el tamaño del producto.

5- Erradicación de fallos.

¿Cuántos fallos han sido corregidos?

Cuente el número de fallos resueltos durante el período de pruebas y compárelos con el número total de fallos detectados y el número total de fallos pronosticados.

Se emplean las siguientes fórmulas:

$$1) \quad X = A1 / A2$$

Donde A1 es el número de fallos solucionados y A2 el número total de fallos reales detectados.

El valor es $0 \leq X \leq 1$ y a mayor cercanía al 1 resultará mejor (cuanto menos fallos queden).

$$2) \quad Y = A1 / A3$$

Donde A3 es el número total de fallos latentes pronosticados.

El valor es $0 \leq Y$, a mayor cercanía al 0 resultará mejor (cuanto menos fallos queden).

2.6.3.2 Métricas de calidad para la especificación de requisitos

Las métricas de calidad para la especificación de requisitos que se proponen son:

1- Especificidad de los requerimientos: calcula la ausencia de ambigüedad, basado en la interpretación de los revisores para cada requisito.

$$Q1 = nui/nr$$

nui: número de requerimientos en los que los revisores tuvieron la misma interpretación.

Capítulo 2 Propuesta de Solución

nr: total de requerimientos, se calcula por la fórmula:

$nr = nf + rnf$, donde

nf: es el número de requerimientos funcionales

rnf: número de requerimientos no funcionales.

Esta métrica puede interpretarse según el valor de Q1, mientras más cerca de 1 esté, menor será la ambigüedad de la especificación.

2- Porcentaje de funciones necesarias especificadas para un sistema. $Q2 = nc / (nc + nnv)$

nc: número de requerimientos validados como correctos.

nnv: número de requerimientos sin validar aún. Esta métrica puede ser interpretada analizando el valor de Q2 que mide el porcentaje de funciones necesarias que se han especificado para un sistema, es decir, la cantidad de requerimientos que se han validado para que un sistema funcione correctamente.

Ahora bien, para que un sistema esté 100% acabado correctamente cada uno de los requerimientos tiene que ser validado, por lo que el valor de Q2 al terminar de aplicar esta métrica tiene que ser 1.

2.6.3.3 Otras métricas

También se propone utilizar las siguientes métricas:

1-La Eficacia de la Eliminación de Defectos (EED) es una medida de la habilidad de filtrar de las actividades de la garantía de calidad y de control, se utiliza de la siguiente forma:

$$EED = E / (E + D)$$

Donde E es el número de errores encontrados antes de la entrega final y D número de defectos encontrados después de la entrega. El objetivo es que EED adquiera el valor de 1, si esto sucede es

Capítulo 2 Propuesta de Solución

que no se han encontrado defectos en el software. Entre más errores se encuentren antes de la entrega mejor funcionarán las técnicas de garantía de calidad.

La EED también puede utilizarse para medir la habilidad de un equipo para encontrar errores antes de pasar a la siguiente actividad, entonces se define la siguiente fórmula:

$$EED (i) = E (i) / (E (i) + E (i + 1))$$

Donde:

E_i : errores detectados en la actividad i ,

$E (i+1)$: errores detectados en la actividad $i+1$ que no se detectaron y provienen de la actividad i , y el objetivo es obtener un valor de 1.

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

3-Defectos por unidad de tiempo (DE).

Los defectos por hora indican la efectividad del tiempo dedicado a las inspecciones. En la medida que el rendimiento incrementa, una disminución de los defectos por unidad de tiempo es natural.

Fórmula:

$$DE = 60 * DEE / ME$$

Donde DE son los defectos encontrados por hora, DEE los defectos eliminados en la etapa y ME los minutos dedicados a esa etapa.

Capítulo 2 Propuesta de Solución

2.6.4 Estándares

2.6.4.1 Organización Internacional de Normalización (ISO)

La Organización Internacional para la Estandarización (ISO), es un organismo de alcance mundial encargado de coordinar y unificar las normas nacionales, agrupando a 130 países. La misión de la ISO es promover el desarrollo de la estandarización y las actividades con ella relacionada en el mundo con la mira en facilitar el intercambio de servicios y bienes, y para promover la cooperación en la esfera de lo intelectual, científico, tecnológico y económico.

Las normas ISO son normas de carácter social al pretender satisfacer las necesidades del cliente, por medio de un producto de alta calidad, el cual se logrará al efectuar todo bien desde el inicio, se debe aclarar que las normas ISO no son de carácter restringido, ya que el objetivo de estas no es imponer reglas cerradas sino posibilitarle a las entidades mercantiles la expansión a mercados exigentes, lo que la ISO tampoco garantiza la estabilidad de la entidad en el mercado tan competitivo, ya que la garantía a la sobrevivencia solo se logrará si y solo si se complace al cliente.

ISO 9000

La familia ISO 9000 es el estándar de normas de calidad más internacional. Su surgimiento es la respuesta a una necesidad expresada por los mercados. Dado que la calidad de los productos es una variable cada vez de mayor importancia en los mercados, las empresas comenzaron a desarrollar distintos manuales con procedimientos y requerimientos para las partes interesadas, que especificaban los requisitos básicos que debían cumplir.

ISO 9000 presenta como sus principios:

- ❖ **Enfoque al cliente:** Las organizaciones dependen de sus clientes y por lo tanto deberían comprender las necesidades actuales y futuras de los clientes, satisfacer los requisitos de los clientes y esforzarse en exceder las expectativas de los clientes.
- ❖ **Liderazgo:** Los líderes establecen la unidad de propósito y la orientación de la organización. Ellos deberían crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.

Capítulo 2 Propuesta de Solución

- ❖ Participación del personal: El personal, a todos los niveles, es la esencia de una organización y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.
- ❖ Enfoque basado en procesos: Un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso.
- ❖ Enfoque de sistema para la gestión: Identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de una organización en el logro de sus objetivos.
- ❖ Mejora continua: La mejora continua del desempeño global de la organización debería ser un objetivo permanente de ésta.
- ❖ Enfoque basado en hechos para la toma de decisión: Las decisiones eficaces se basan en el análisis de los datos y la información.
- ❖ Relaciones mutuamente beneficiosas con el proveedor: Una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

ISO 9000-3

Este estándar tiene como objetivo crear una norma internacional de calidad. Para ello se basa en desarrollar prácticas de administración para garantizar el Aseguramiento y Administración de la Calidad.

Esta norma se divide en 4 guías fundamentales:

- ❖ ISO 9004 Elementos para la Administración y el Sistema de Calidad. Guía para el Sistema de Aseguramiento de la Calidad.
- ❖ ISO 9001 Sistemas de Calidad. Modelo de Aseguramiento de la Calidad en el diseño, desarrollo, producción, instalación y servicio.
- ❖ ISO 9002 Sistemas de Calidad. Modelo de Aseguramiento de la Calidad para la producción, instalación y servicio.

Capítulo 2 Propuesta de Solución

- ❖ ISO 9003 Sistemas de Calidad. Modelo de Aseguramiento de la Calidad para la inspección final y pruebas.

Además, existen otras normas y entre ellas las más relevantes son:

- ❖ ISO 9000-1 Guía para la selección de la norma a usar.
- ❖ ISO 8402 Recopilación de definiciones. Vocabulario.
- ❖ ISO 9000-3 Guía para la aplicación de ISO 9001 al desarrollo, suministro y mantenimiento.

De forma general esta norma tiene como requerimientos fundamentales:

- ❖ Escribir un manual de calidad, describiendo el Sistema de Calidad en alto nivel.
- ❖ Escribir documentos en forma de procedimientos que describan cómo debe hacerse el trabajo en la organización.
- ❖ Crear un sistema para controlar la distribución y reedición de documentos.
- ❖ Diseño e implantación de un sistema de acciones preventivas y correctivas para prevenir la ocurrencia de problemas.
- ❖ Identificar las necesidades en cuanto a entrenamiento en la organización.
- ❖ Determinar las medidas y equipos para realizar las pruebas.
- ❖ Capacitar al personal de la organización en la operación del Sistema de Calidad.
- ❖ Planificar y llevar a cabo auditorías de calidad internas.
- ❖ Tener en cuenta los requerimientos del estándar con los que no cumple la organización.

Esta es una de las normas más populares pero pueden surgir errores a la hora de adaptar esta norma a un proyecto por lo genérica que resulta ser.

Ventajas del modelo ISO 9000-3

- ❖ Es un factor competitivo para las empresas.
- ❖ Proporciona confianza a los clientes.
- ❖ Ahorra tiempo y dinero, evitando volver a certificar la calidad según los estándares locales o particulares de una empresa.

Capítulo 2 Propuesta de Solución

- ❖ Se ha adaptado a más de 90 países e implantado a todo tipo de organizaciones industriales y de servicios, tanto sector privado como público.
- ❖ Proporciona una cierta garantía de que las cosas se hacen tal como se han dicho que se han de hacer.

Desventajas del modelo ISO 9000-3

- ❖ Es costoso.
- ❖ Muchas veces se hace por obligación.
- ❖ Es cuestión de tiempo que deje de ser un factor competitivo.
- ❖ Hay diferencias de interpretación de las cláusulas del estándar.
- ❖ No es indicativa de la calidad de los productos, procesos o servicio.
- ❖ Hay mucha publicidad engañosa.

ISO/IEC 12207

La Norma ISO/IEC 12207 pertenece a la gran familia de ISO, cuya segunda edición fue elaborada en febrero de 2003 y emitida en junio de 2006 por la Dirección de Calidad conjuntamente con la versión en español de la Norma Internacional ISO/IEC 90003:2004.

Los procesos en esta Norma Internacional forman un conjunto amplio. Una organización, dependiendo de su propósito, puede seleccionar un subconjunto apropiado para cumplir ese propósito, está diseñada, por lo tanto, para ser ajustada a una organización, proyecto o aplicación particular.

Este modelo de calidad se aplica a la adquisición de sistemas, productos y servicios de software así como al suministro, desarrollo, operación y mantenimiento de productos de software. Están incluidos además aquellos aspectos de la definición del sistema necesarios para proporcionar el contexto de los productos y servicios de software.

Entre los contenidos que aborda comprende temas importantes como el Proceso de Aseguramiento de la Calidad, el cual se apoya en otros procesos tales como la Verificación, Validación, Revisión Conjunta, Auditoría y Solución de Problemas.

Capítulo 2 Propuesta de Solución

El proceso de Verificación determina si los productos de software de una actividad cumplen las condiciones o los requisitos impuestos sobre estos en las actividades precedentes.

El proceso de Validación determina si los requisitos y el sistema o el producto de software, tal como se han construido, cumplen con su uso específico previsto. La Validación se puede realizar en etapas tempranas del proyecto. Este proceso puede ser ejecutado como una parte de la actividad Apoyo a la Aceptación del Software.

La Revisión Conjunta evalúa el estado y los productos de una actividad de un proyecto, cuando sea apropiado. Las revisiones conjuntas se realizan tanto al nivel de la gestión del proyecto como técnico durante toda la vigencia del contrato.

La Auditoría es un proceso para determinar la conformidad con los requisitos, planes y contratos, cuando sea pertinente. Puede ser empleado por dos partes cualesquiera, donde la parte auditora audita los productos de software o actividades de la parte auditada.

Para solucionar los problemas es necesario analizarlos y resolverlos (incluyendo las no conformidades), cualquiera que sea su naturaleza u origen, que se descubran durante la ejecución de los procesos de desarrollo, operación, mantenimiento u otros procesos. El objetivo es proporcionar los medios oportunos, confiables y documentados para asegurar que todos los problemas descubiertos son analizados y resueltos así como que las tendencias de los mismos son reconocidas.

La norma ISO 9004:2000

La norma ISO 9004:2000 es un documento genérico que pretende ser utilizable como un medio para que el sistema de gestión de la calidad avance hacia la excelencia.

La norma ISO 9004, la cual está basada en ocho principios de gestión de la calidad, no fija requisitos sino que da directrices para la aplicación y uso de un sistema de gestión de la calidad para mejorar el desempeño total de la organización. Por ello no se aplica en certificación y ha sido redactada para ser utilizada por la alta Dirección.

El implementar la norma ISO 9004:2000 pretende alcanzar no sólo la satisfacción de los clientes de la organización, sino también de todas las partes interesadas, incluyendo al personal, a los propietarios, accionistas e inversionistas, proveedores y socios y la sociedad en su conjunto.

La Norma ISO 9001

Capítulo 2 Propuesta de Solución

La Norma ISO 9001 especifica los requisitos para los sistemas de gestión de la calidad aplicables a toda organización que necesite demostrar su capacidad para proporcionar productos que cumplan los requisitos de sus clientes y los reglamentarios que le sean de aplicación y su objetivo es aumentar la satisfacción del cliente.

Requisitos de la Norma ISO 9001:2000:

1-Sistema de gestión de la Calidad.

- ❖ Requisitos Generales.
- ❖ Generalidades.
- ❖ Manual de Calidad.
- ❖ Control de Documentos.
- ❖ Control de los Registros.

2- Responsabilidad de la Dirección.

- ❖ Compromiso de la Dirección.
- ❖ Enfoque al cliente.
- ❖ Política de Calidad.

3- Planificación.

- ❖ Objetivos de la Calidad.

Los requisitos del sistema de Gestión de la Calidad son complementarios a los requisitos del producto y se especifican en la norma ISO 9001:2000, son genéricos y aplicables a organizaciones de cualquier sector económico e industrial con independencia del producto que suministren y además hacen énfasis en el uso y aplicaciones técnicas del producto.

La Norma ISO 19011

Esta norma internacional proporciona orientación sobre los fundamentos de la auditoría, la gestión de los programas de auditoría, la conducción de auditorías de los sistemas de gestión de la calidad y ambientales, así como las calificaciones para los auditores de los sistemas de gestión de la calidad y ambientales.

Se refiere tanto a los Sistemas de Gestión de la Calidad como a los de Gestión Ambiental, mientras que la norma previa, ISO 10011 únicamente proporcionaba orientación sobre la auditoría de los sistemas de gestión de la calidad. Se aplica en auditorías internas y externas y sustituye a las normas ISO 10011-1, ISO 10011-2, ISO 10011-3 (Auditorías de Calidad).

Capítulo 2 Propuesta de Solución

NC-ISO/IEC 9126

La NC-ISO/IEC 9126 describe un modelo en dos partes para calidad de los productos de software: a) calidad interna y externa y b) calidad durante el uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en subcaracterísticas que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software.

La segunda parte del modelo especifica cuatro características de calidad durante el uso, pero no elabora dicho modelo a un nivel inferior al de las características. La calidad durante el uso es el efecto combinado para el usuario de las seis características de calidad del producto de software.

Esta parte de la NC ISO/IEC 9126 permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software. Por ejemplo puede ser utilizada por los programadores, los clientes, el personal de aseguramiento de la calidad y los evaluadores independientes, particularmente los responsables de especificar y evaluar la calidad de los productos de software. El modelo de calidad definido en esta parte de la NC ISO/IEC 9126 puede usarse para:

- ❖ Validar la integridad de la definición de los requisitos.
- ❖ Identificar los requisitos del software.
- ❖ Identificar los objetivos del diseño del software.
- ❖ Identificar los objetivos de ensayo del software.
- ❖ Identificar los criterios de aseguramiento de la calidad.
- ❖ Identificar los criterios de aceptación para un producto de software terminado.
- ❖ Esta parte de la NC ISO/IEC 9126 se puede utilizar junto con la ISO/IEC 9001 (que se relaciona con los procesos de gestión de la calidad) para garantizar:
 - ❖ un apoyo para establecer metas de calidad.
 - ❖ un apoyo para la revisión, verificación y validación del diseño.

Capítulo 2 Propuesta de Solución

2.6.4.2 Modelo de madurez de capacidad integrada.

El Modelo de Madurez y Capacidad Integrado (CMMI) del Instituto de Ingeniería de Software (SEI), es un marco de trabajo que describe cómo evolucionar de un proceso de desarrollo de software empírico e inmaduro a un proceso efectivo, maduro y disciplinado.

En agosto de 2006 aparece CMMI v1.2, con el propósito de proveer mejoras en costo, entregas en tiempo y calidad de proyectos en desarrollo de ingeniería.

Los cambios se producen para mejorar la calidad de los productos CMMI y la consistencia con que son aplicados. Entre las mejoras al modelo se desatacan:

- ❖ Ambas representaciones, escalonada y continua, ahora están juntas.
- ❖ Fueron eliminados los conceptos de «práctica avanzada» y «característica común».
- ❖ El objetivo genérico y las descripciones de las prácticas fueron movidas a la parte dos.
- ❖ Fueron agregadas ampliaciones con respecto a hardware y ejemplos.
- ❖ Todas las definiciones fueron consolidadas en el glosario.
- ❖ Las prácticas de Integración de productos y procesos en desarrollo (IPPD) fueron consolidadas y simplificadas. No hay más áreas de proceso separadas para IPPD.
- ❖ Fueron agregadas elaboraciones de prácticas genéricas a las prácticas genéricas de nivel 3.
- ❖ Se agregó una explicación sobre cómo las áreas de proceso apoyan la implementación de prácticas genéricas.
- ❖ Fue agregado un material para asegurar que los procesos estándares sean desplegados en los proyectos desde su inicio.

Esta nueva versión de CMMI cuenta con varias Áreas de Procesos las cuales se agrupan en 4 categorías:

Administración de procesos:

- ❖ Innovación y despliegue de organización.
- ❖ Definición de proceso de organización +IPPD.
- ❖ Foco de proceso de organización.
- ❖ Funcionamiento de proceso de organización.

Capítulo 2 Propuesta de Solución

- ❖ Entrenamiento de la organización.

Administración de proyecto:

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

Ingeniería:

- ❖ Integración del producto.
- ❖ Desarrollo de los requisitos.
- ❖ Administración de requisitos.
- ❖ Solución técnica.
- ❖ Validación.
- ❖ Verificación.

Soporte:

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

CMMI tiene dos representaciones, la escalonada y la continua. Ambas representaciones proporcionan maneras de poner la mejora en ejecución del proceso para alcanzar objetivos del negocio. Además proporcionan el mismo contenido esencial y utilizan los mismos componentes del modelo.

Representación continua:

Capítulo 2 Propuesta de Solución

Ofrece flexibilidad máxima al usar un modelo CMMI para la mejora de procesos. Una organización puede elegir mejorar el funcionamiento de un solo punto proceso-relacionado o puede trabajar en varias áreas que se alineen de cerca con los objetivos del negocio de la organización. Hay algunas limitaciones en las opciones de una organización debido a dependencias entre algunas áreas de proceso. La representación continua se refiere a seleccionar el área de proceso particular a mejorar y el nivel deseado de la capacidad para esa área de proceso. En este contexto, si un proceso es realizado o incompleto es importante. Por lo tanto, el incompleto conocido será el punto de partida continuo de la representación.

Representación escalonada:

Ofrece una manera sistemática, estructurada a mejorar el proceso. La realización de cada etapa se asegura de que una infraestructura de proceso adecuada se ha puesto como premisa para la etapa siguiente. Señala una orden para poner áreas de proceso en ejecución según los niveles de madurez que definen la trayectoria de la mejora para una organización del nivel inicial al nivel óptimo. La representación escalonada se refiere a la madurez total de la organización, los procesos individuales son realizados o incompletos.

Si no se sabe por dónde comenzar y qué proceso elegir para mejorar, la representación escalonada es una buena opción. Proporciona un sistema específico de procesos para mejorar en cada etapa.

Los niveles caracterizan la mejora de un estado mal definido a un estado que utilice la información cuantitativa para determinar y para manejar las mejoras que son necesarias para resolver los objetivos del negocio de una organización. Para alcanzar un nivel particular, una organización debe satisfacer todas las metas apropiadas del área o del sistema de proceso que son apuntadas para la mejora.

Los niveles se utilizan en CMMI para describir una trayectoria evolutiva recomendado para una organización que desee mejorar los procesos que utiliza para desarrollar y mantener los productos y servicios. Cuenta con dos representaciones para la mejora. Una permite a las organizaciones mejorar los procesos que corresponden a un área de proceso individual (o a áreas de proceso) seleccionadas por la organización y la otra permite mejorar un sistema de procesos relacionados incremental tratando los sistemas sucesivos de áreas de proceso.

Capítulo 2 Propuesta de Solución

En la representación continua utiliza niveles de la capacidad, mientras que la representación escalonada utiliza niveles de madurez. Los niveles de la capacidad y los niveles de la madurez proporcionan una manera de medir como las organizaciones pueden mejorar sus procesos. Sin embargo, el acercamiento asociado a la mejora de proceso es diferente.

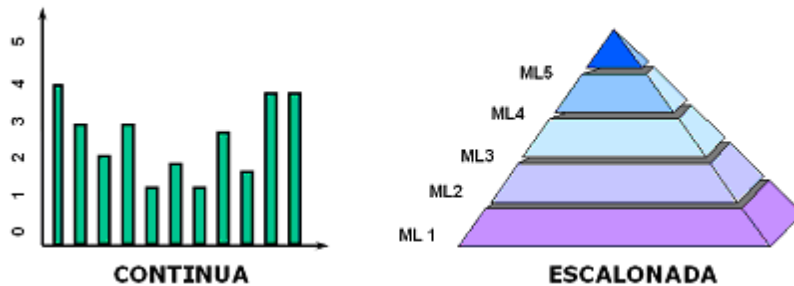


Figura 2.1 Representaciones de CMMI

Disciplinas que cubre el modelo CMMI

La intención de CMMI es proveer un CMM que cubra el desarrollo y mantenimiento de productos y servicios, así como un marco de trabajo extensible que ayude a agregar nuevas disciplinas. Actualmente hay cuatro disciplinas disponibles:

1. Ingeniería de Sistemas (SE). Cubre el desarrollo del total de los sistemas, incluyendo o no el software. Los ingenieros de sistemas se enfocan en la transformación de intereses, requerimientos y necesidades de los clientes en productos y el soporte de estos productos a través de su ciclo de vida. Cubre la construcción de un sistema con o sin software.
2. Ingeniería de Software (SW). Cubre el desarrollo de los sistemas de software. Los ingenieros de software se enfocan en la aplicación sistemática, disciplinada, y aproximaciones cuantificadas para el desarrollo, operación, y mantenimiento del software. Cubre la construcción de soluciones software.
3. Producto Integrado y Desarrollo de Procesos (IPPD). Es una aproximación sistemática que obtiene en forma temprana la colaboración de los stakeholders más importantes, a través del ciclo de vida del producto, para la satisfacción de los requerimientos y necesidades de los clientes. Hay que tener en

Capítulo 2 Propuesta de Solución

cuenta que si un proyecto o una organización elige IPPD, debe conjuntamente trabajar con otra disciplina adicional (aquellas relacionadas con sistemas de ingeniería). Cubre la relación a largo plazo con el cliente.

4. Relación con proveedores (Supplier Sourcing). A medida que el esfuerzo del trabajo se convierte en más complejo, los administradores de proyecto deben usar proveedores para realizar funciones o agregar modificaciones a productos que son necesarios para el proyecto. Bajo estas circunstancias, la disciplina de Supplier Sourcing se encarga de la adquisición de esos productos de manos de sus proveedores. Similar a IPPD, Supplier Sourcing debe aplicarse conjuntamente con otras áreas prácticas usadas para generar el producto. Cubre los procesos relacionados con la subcontratación de partes del sistema.

Beneficios de CMMI

CMMI es un producto que está a la vanguardia de la mejora de procesos ya que provee las últimas mejores prácticas del desarrollo y mantenimiento de productos y servicios. Los modelos de CMMI se mejoran sobre las mejores prácticas de los modelos previos, en varios caminos importantes. Las prácticas del modelo CMMI permiten a las organizaciones hacer lo siguiente:

1. Mayor manejo de conexiones explícitas y actividades de ingeniería para sus objetivos de negocio.
2. Expandir el alcance y la visibilidad dentro del ciclo de vida del producto y dentro de las actividades de ingeniería, para asegurar que los productos o servicios satisfacen las expectativas de los clientes.
3. Incorporar lecciones aprendidas desde las mejores prácticas de áreas adicionales (medidas, gestión de riesgo y gestión de proveedores).
4. Las organizaciones que lograron el nivel de madurez 4 o 5 usando SW-CMM generaron información al SEI sobre sus sucesos y dificultades. Esto dio lugar a implementar prácticas de madurez más robustas.
5. Dirigir funciones críticas de organización adicionales para sus productos y servicios.

Capítulo 2 Propuesta de Solución

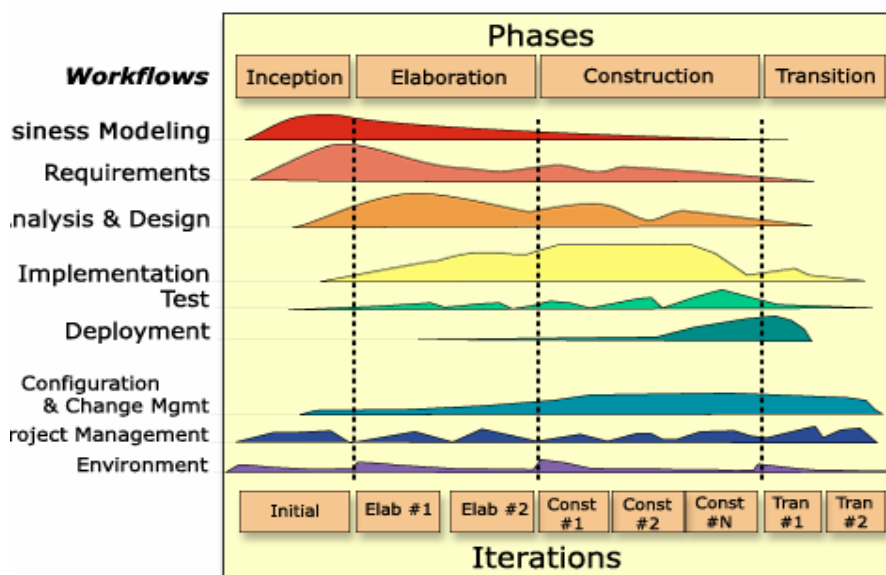
Proceso Unificado del Software (RUP)

El Proceso Unificado de Desarrollo de Software (RUP) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

RUP es un proceso de desarrollo de software que constituye una forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo. Su principal objetivo es la producción de software de calidad dentro de plazos y costos predecibles.

RUP ha agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos de apoyo, además divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante.

La figura (RUP en dos dimensiones) representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.



Capítulo 2 Propuesta de Solución

Figura 2.2 RUP en dos dimensiones

Los flujos de trabajo son los que a continuación se describen:

1. Modelación del negocio: Describe los procesos de negocio, identificando quienes participan y las actividades que requieren automatización.
2. Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
3. Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
4. Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
5. Prueba (Testeo): Es una actividad de control de la calidad que busca los defectos a lo largo del ciclo de vida de un producto.
6. Despliegue: Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
7. Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a utilización/actualización concurrente de elementos, control de versiones, etc.
8. Administración de proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Capítulo 2 Propuesta de Solución

9. Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El ciclo de vida de RUP se caracteriza por ser:

- ❖ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- ❖ Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- ❖ Iterativo e Incremental: Aunque se puede sugerir que los flujos de trabajo se desarrollan en cascada, la lectura de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

A lo largo del ciclo de vida, se realiza el manejo de la calidad, las mediciones y valoraciones del proceso así como del producto. La evaluación de calidad puede ocurrir cuando un evento mayor ocurre, como al final de una fase, o puede ocurrir cuando un artefacto se produce, como un camino directo al código.

Las fases de RUP son las siguientes:

Conceptualización (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Capítulo 2 Propuesta de Solución

Al final de la fase tiene lugar el primer hito, objetivos, se examinan los objetivos del ciclo de vida del proyecto y se decide si proceder con él o cancelarlo.

Criterios de evaluación:

- ❖ La concurrencia de Stakeholder en la definición del alcance y estimaciones del costo plasmadas en el plan.
- ❖ Los requisitos capturados son realmente los que se necesitan.
- ❖ Todos los riesgos se han identificado y una estrategia de la mitigación existe para cada uno.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Al culminar la fase tiene lugar el segundo hito importante del proyecto, Arquitectura, es donde se examinan los objetivos del sistema detallados y el alcance, la opción de arquitectura, y la resolución de los riesgos mayores.

Criterios de evaluación:

- ❖ La Visión del producto y requisitos son estables.
- ❖ La arquitectura es estable.
- ❖ Los acercamientos importantes que se utilizarán en prueba y la evaluación son probados.
- ❖ La prueba y la evaluación de prototipos ejecutables han demostrado que se han tratado y resuelto los elementos principales del riesgo.
- ❖ El plan de iteración para la fase de construcción está lo suficientemente detallado y da la fidelidad para permitir que el trabajo proceda.
- ❖ Los planes de la iteración para la fase de la construcción son apoyados por estimaciones creíbles.
- ❖ Todos los stakeholders convienen que la visión actual se puede resolver si el plan actual se ejecuta para desarrollar el sistema completo, en el contexto de la arquitectura actual.
- ❖ El gasto real del recurso contra gasto previsto es aceptable.

El proyecto puede ser abandonado o ser meditado considerablemente si no puede alcanzar este hito.

Capítulo 2 Propuesta de Solución

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.

Al inicio de esta fase, el producto está listo para ser entregado al equipo de la transición, se ha desarrollado toda la funcionalidad, además del software se ha desarrollado un manual de usuario, y hay una descripción del lanzamiento actual.

Criterios de evaluación:

El criterio de evaluación para la fase de construcción envuelve las respuestas a las siguientes preguntas:

- ❖ ¿Es este lanzamiento de producto estable y lo bastante maduro como para desplegarlo en la comunidad de usuario?
- ❖ ¿Todos los stakeholders están listos para la transición en la comunidad de usuario?
- ❖ ¿Son los gastos reales de recursos realmente aceptables?

La transición puede ser pospuesta por un lanzamiento si el proyecto no puede alcanzar este hito.

Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Al final de la fase tiene lugar el cuarto hito importante del proyecto. A este punto, decides si los objetivos fueron resueltos, y si comienzas otro ciclo de desarrollo. En algunos casos este hito puede coincidir con el final de la fase del inicio para el ciclo siguiente. El hito del lanzamiento de producto es el resultado del cliente que repasa y que acepta los ejecutables del proyecto.

Los criterios primarios de la evaluación para la fase de la transición implica la respuesta a la pregunta siguiente:

- ❖ ¿Están los usuarios satisfechos?

En el hito del lanzamiento del producto, el producto está en la producción y después de lanzado el ciclo del mantenimiento comienza. Esto puede implicar el comenzar un nuevo ciclo, o un cierto lanzamiento adicional del mantenimiento.

Capítulo 2 Propuesta de Solución

2.6.5 Proceso de revisiones y auditorías

Para llevar a cabo el proceso de Aseguramiento de la Calidad del Software se realizan una serie de actividades de gran importancia, una de estas actividades es el proceso de revisiones, con la cual se detectan defectos en etapas tempranas del desarrollo de los proyectos de software, lo que constituye una garantía del producto final.

Durante el desarrollo de revisiones y auditorías se llevan a cabo una serie de tareas. Existen diferentes tipos de revisiones que pueden llevarse a cabo: (3)

Reuniones Informales

Una reunión que tiene lugar alrededor de una máquina, u otro espacio en el cual se discuten problemas técnicos. Cualquier producto de trabajo puede ser objeto de este tipo de revisión.

Presentación formal

Una presentación formal de un diseño de software, una aplicación o cualquier producto de trabajo que se realiza bajo la audiencia de clientes, ejecutivos o personal técnico. Los productos de trabajo/artefactos que serán objeto de este tipo de revisión son los siguientes: Los diseños de arquitectura de información, diagramas de despliegue, diagramas de arquitectura, propuesta de soluciones tecnológicas, prototipos funcionales y no funcionales, configuración del sistema, presentación y análisis de informes.

Revisiones técnicas formales

Actividad desarrollada por los ingenieros del software que tiene como objetivo los siguientes:

- ❖ Descubrir errores en la función, la lógica o la implementación de cualquier representación de software.
- ❖ Verificar que el software bajo revisión alcanza los requisitos.

Capítulo 2 Propuesta de Solución

- ❖ Garantizar que el software ha sido representado de acuerdo a varios estándares predefinidos.
- ❖ Conseguir un software desarrollado de forma uniforme.
- ❖ Las revisiones técnicas formales tienen tres elementos fundamentales:
 - ❖ Informe escrito del estado del producto revisado.
 - ❖ La participación activa y abierta de todos los del grupo de revisión.
 - ❖ Total responsabilidad de todos los participantes en la calidad de la revisión.

Ventajas de las revisiones técnicas formales:

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

Los productos de trabajo/artefactos que serán objeto de este tipo de revisión son los siguientes: Documentación del sistema (informes, registros de calidad, expediente de proyecto, modelos, diagramas), productos desarrollados y la gestión de configuración del software.

Las revisiones Pass Around

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

Capítulo 2 Propuesta de Solución

2.6.5.1 Plan de revisiones y auditorías.

El plan de revisiones y auditorías es la actividad más importante dentro del plan de aseguramiento de la calidad.

El procedimiento para introducir las revisiones en el proceso de desarrollo de software es:

- ❖ En caso de que no esté conformado el equipo de Aseguramiento de Calidad del Software, conformarlo y definir el administrador de calidad, el cual se encargaría de las funciones correspondientes a este grupo.
- ❖ Definir roles adecuados para el grupo de desarrollo del proyecto, entre los que deben estar: Líder del proyecto, Desarrolladores y Administrador de Calidad.
- ❖ Ejecutar revisión del producto terminado que ha desarrollado el proyecto.
- ❖ Definir e introducir métricas que permitan analizar la efectividad de las revisiones.
- ❖ Introducción de herramientas automatizadas para planificación y seguimiento de las revisiones.

Revisiones a utilizar en el proceso de revisión y auditoría del proyecto Servicios Comunitarios

Las diferentes revisiones que se le realizan al software sirven para purificar las actividades de ingeniería del software, que suceden como resultado del análisis, el diseño, y la implementación.

- ❖ Revisión de los Requerimientos: Asegurar que el sistema tenga condiciones o capacidades para satisfacer las exigencias expuestas por el cliente (el artefacto revisado es la lista de requerimientos).
- ❖ Revisión de la Arquitectura: Garantizar un diseño preliminar (el artefacto revisado es la arquitectura).
- ❖ Revisión del análisis y diseño: Asegurar que el modelo se acerque lo más posible a la implementación para tener una idea más clara de lo que se quiere en el sistema (los artefactos revisados son modelo de casos de uso del sistema, actores que se relacionan con el modelo, clases del diseño, diagramas de secuencia, base de datos.)
- ❖ Revisión del proceso.
- ❖ Revisión Administrativa: Revisión de aprobación del proyecto, revisión de la planificación del proyecto y revisión del plan de iteración.

Capítulo 2 Propuesta de Solución

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

Las auditorías realizadas son:

- ❖ Auditoría de la Configuración Funcional: Para verificar que todos los requerimientos han sido cumplidos, donde los artefactos son la capa de presentación, capa de negocio y capa de acceso a datos.
- ❖ Auditoría de la Configuración Física: Para verificar que el software y su documentación están completos y listos para entregar.
- ❖ Auditoría del proceso.

Para realizar las revisiones previstas en el plan se tienen definido un proceso general aplicable a todos los tipos de revisiones definidas:

Proceso:

- ❖ Realizar las revisiones de acuerdo al plan de proyecto.
- ❖ Definir el producto a revisar.
- ❖ Definir día y hora para la revisión.
- ❖ Determinar que es necesario y quien debe hacerlo.
- ❖ Definir el tipo de revisión a realizar.
- ❖ Identificar a las personas que deben participar e invitarlas indicándole su responsabilidad en la revisión.
- ❖ Definir que se debe hacer durante la revisión y quien debe hacerlo.
- ❖ Definir cuando puede finalizar la revisión.
- ❖ Identificar y guardar los resultados que deben conservarse de la revisión.

En la siguiente tabla se hace una breve descripción del proceso de revisiones y auditorías propuesto para el proyecto Servicios Comunitarios.

Estas revisiones se deben realizar a cada iteración que se haga en el proyecto.

Capítulo 2 Propuesta de Solución

Tipo de revisión	Objetivos	Descripción (Iteración)	Fase del Proyecto	Responsable
Revisión Administrativa	<ul style="list-style-type: none"> • Revisión de aprobación del proyecto. • Revisión del plan de iteración 	Inicio del proyecto	Inicio	<ul style="list-style-type: none"> • Líder del proyecto. • Administrador de calidad
Revisión post-mortem	❖ Revisión de la aceptación de la iteración.	Inicio de la iteración 1	Inicio	❖ Administrador de calidad
Revisión del Modelo del Negocio	Artefacto: Diagrama de casos de uso del negocio	Inicio de la iteración 1	Inicio	❖ Administrador de calidad
Revisión de los requerimientos	Todos los módulos	Iteración 1	Inicio	❖ Administrador de calidad
Revisión de la Arquitectura	Garantizar diseño	Iteración 1	Elaboración	❖ Administrador de calidad

Capítulo 2 Propuesta de Solución

	preliminar Artefacto: Arquitectura			
Revisión del análisis y diseño	Todos los módulos	Iteración 1	Elaboración	❖ Administrador de calidad
Auditoría configuración funcional	❖ Capa de presentación. ❖ Capa de negocio. ❖ Capa de acceso a datos.	Iteración 1	Construcción	❖ Administrador de calidad
Auditoría de la Configuración Física	❖ Verificación de toda la documentación del software. ❖ Verificación del producto Final.		Transición	❖ Administrador de calidad

Tabla 2 Revisiones y auditorías por fase.

En la siguiente tabla se muestra la organización y responsabilidades dentro del plan de revisiones y auditorías.

Capítulo 2 Propuesta de Solución

Tipo de revisión	Personal involucrado en las actividades de revisión y auditorias.	Descripción de las tareas y responsabilidades de los involucrados.
Revisión administrativa	Líder del proyecto	<p>Inicia el proyecto y planifica su desarrollo.</p> <p>Define los roles y la organización del proyecto.</p> <p>Controla cada iteración y fase.</p> <p>Organiza y orienta las actividades para el desarrollo del proyecto.</p>
Revisión de los Requerimientos	Analista del sistema. Arquitecto.	<p>Prioriza los casos de uso.</p> <p>Realiza el análisis de la arquitectura y prueba su viabilidad.</p> <p>Estructura el modelo de análisis, diseño, implementación y despliegue.</p>

Capítulo 2 Propuesta de Solución

Revisión post-mortem	Líder del proyecto Analista del sistema	Desarrollar el documento visión. Desarrollar el Plan de Administración de Requerimientos. Buscar Actores y Casos de Uso. Estructurar el modelo de Casos de Uso del sistema.
Revisión de análisis y diseño.	Diseñador de interfaz de usuario. Diseñador de la de base de datos. Revisor técnico	Construir el diseño de la base de datos. Revisión técnica a los artefactos generados durante el proceso.

Capítulo 2 Propuesta de Solución

	<p>Analista del sistema</p> <p>Arquitecto del software</p>	<p>Implementar y ejecutar pruebas al sistema.</p> <p>Analizar las fallas para encontrar los errores y solicitar los cambios.</p>
<p>Auditoría de la configuración funcional.</p>	<p>Desarrolladores.</p>	<p>Implementar los elementos diseñados.</p> <p>Desarrollar y ejecutar las pruebas de implementación sobre las clases programadas.</p>
<p>Auditoría de la configuración física.</p>	<p>Analista del sistema</p> <p>Arquitecto del software</p> <p>Líder del proyecto</p>	<p>Verificar que todos los documentos realizados en el proyecto estén completos y listos para entregar</p>

Capítulo 2 Propuesta de Solución

Tabla 3 Organización y responsabilidades dentro del plan de revisiones y auditorías.

Resolución de problemas y actividades de corrección

El proceso para informar y manejar problemas en las revisiones sería el siguiente:

- ❖ Una vez concluida la revisión el revisor debe emitir un documento con las no conformidades encontradas, de existir alguna.
- ❖ Este documento debe ser enviado al responsable del producto o artefacto en revisión.
- ❖ El responsable se encargará de arreglar las no conformidades encontradas.
- ❖ Una vez resueltas las no conformidades se debe emitir un documento de respuesta a no conformidades para el líder del proyecto quien decidirá si es necesaria una nueva revisión.
- ❖ Los documentos deben almacenarse como registros de calidad.

2.6.6 Herramientas, Técnicas y Metodologías

Para la realización de este plan se utilizaron una serie de herramientas, técnicas y metodologías:

Estándares y Normas de la IEEE, ISO.

Plantillas del expediente de proyecto establecido por la DCS para registrar la documentación del proyecto.

Para la realización de las revisiones se utilizaron las revisiones técnicas formales mencionadas en secciones anteriores y las listas de chequeo.

Desarrollo de casos de pruebas.

2.6.7 Gestión de Configuración

El proceso de gestión de configuración se aplica durante todo el ciclo de vida del sistema para controlar las modificaciones realizadas en cualquier módulo del proyecto y las versiones de los elementos que componen el sistema.

2.6.8 Pruebas y evaluaciones

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. La creciente percepción del software como un elemento del sistema y la importancia de los costes asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien planificadas.

Capítulo 2 Propuesta de Solución

La prueba de software permite al desarrollador determinar si el producto generado satisface las especificaciones establecidas. Así mismo, una prueba de software permite detectar la presencia de errores que pudieran generar salidas o comportamientos inapropiados durante su ejecución.

Las pruebas de software son una parte del proceso de aseguramiento de calidad; realizar pruebas a un sistema de información no significa necesariamente que el proceso de desarrollo esté asegurado y tampoco que de manera directa esté mejorando. Pero implementar un proceso de pruebas de software y más aún sostenerlo en el tiempo, es un buen inicio para más adelante aumentar el alcance y con base en las reflexiones al interior del equipo y de los hallazgos registrados en su producto, realizar un mejoramiento del proceso de desarrollo basado en los lineamientos del aseguramiento de calidad de software.

Como guía para la aplicación de pruebas existe el plan de pruebas. En dicho plan se describe de forma específica cómo el equipo de aseguramiento de calidad realizará todas sus pruebas.

Las pruebas recomendadas a aplicar en el proyecto Servicios Comunitarios son las que se mencionan a continuación:

Técnicas:

Pruebas de Caja Blanca

Pruebas de Caja Negra.

Tipos de pruebas:

Pruebas de Integración.

Pruebas de Sistema

Pruebas de Aceptación

Prueba de caja blanca: Durante la aplicación de esta técnica se trabaja con el código del proyecto, este es separado por segmentos y se le asigna un número a cada instrucción con el objetivo de crear un grafo con estos números. Cada sentencia tiene una representación, con esta prueba se realiza la revisión del código detectando errores lógicos dentro de este. Una vez hecho esto se calcula la complejidad ciclomática del grafo formado con el objetivo de definir la cantidad de casos de pruebas que se van a hacer.

Con este método se determina cuáles son los casos de prueba a partir del código fuente del software y se utilizan las especificaciones para determinar el resultado esperado del caso. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y ciclos. Se puede

Capítulo 2 Propuesta de Solución

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

Prueba de Caja negra: Las técnicas de pruebas de caja negra consisten en seleccionar los casos de prueba basándose en la funcionalidad esperada del mismo, es decir, en la especificación de lo que debe hacer el programa. El objetivo de estas pruebas es comprobar si existen errores en la funcionalidad del software, es decir se comprueban las interfaces del mismo para detectar defectos de función en ellas. Antes de realizar un programa conviene tener por escrito qué es lo que debe hacer el mismo (documento de especificación software), las pruebas de caja negra deben emplear esta documentación para seleccionar los casos de prueba. La prueba de todos los casos posibles es inviable, por lo que, al igual que para las pruebas de caja blanca, se debe realizar una selección de los casos que se van a utilizar.

Pruebas de Integración: Comprueban la correcta unión de los componentes del sistema entre sí a través de sus interfaces, y si cumplen con la funcionalidad establecida. Tienen por objetivo verificar el conjunto funcionamiento de dos o mas módulos, si bien se deben poner en práctica desde la creación de dos módulos que interactúen entre si, en el supuesto caso que se necesiten mas de dos módulos para efectuar las pruebas, deberán generarse simples emuladores de módulos que entreguen datos esperados para la prueba individual de cada uno.

Las pruebas de integración se centran en probar la coherencia semántica entre los diferentes módulos, tanto de semántica estática (se importan los módulos adecuados; se llama correctamente a los procedimientos proporcionados por cada módulo), como de semántica dinámica (un módulo recibe de otro lo que esperaba). Normalmente estas pruebas se van realizando por etapas, englobando progresivamente más y más módulos en cada prueba.

Las pruebas de integración se pueden empezar en cuanto se tengan unos pocos módulos, aunque no terminarán hasta disponer de la totalidad.

Pruebas de Sistema: Prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción. Uno de los objetivos de la fase de pruebas del sistema es verificar que el comportamiento externo del sistema software satisface los requisitos establecidos por los clientes y futuros usuarios del mismo. A medida que aumenta la complejidad de los sistemas de software y aumenta la demanda de calidad, se hacen necesarios procesos y métodos que permitan obtener buenos conjuntos de pruebas del sistema. Este trabajo describe los modelos necesarios para generar de manera sistemática un conjunto de pruebas que permitan verificar la implementación de los requisitos funcionales de un sistema de software.

Capítulo 2 Propuesta de Solución

Pruebas de Aceptación: Verifican que el sistema cumple con todos los requisitos indicados y permite que los usuarios del sistema den el visto bueno definitivo. Con este tipo de prueba se comprobará en la práctica que el sistema posee todas las funcionalidades exigidas en el cuestionario de especificaciones técnicas tanto en los que respecta a las características generales como en lo relativo a cada uno de los grupos de funciones y utilidades. Para ello se realizarán pruebas reales de funcionamiento sobre todas y cada una de las características funcionales exigidas.

Las pruebas a los productos de software se le pueden aplicar a lo largo del ciclo de desarrollo de los mismos, las pruebas anteriormente mostradas, aplicables al proyecto Servicios Comunitarios, también se ponen en práctica en diferentes fases de desarrollo.

La prueba de caja blanca, se aplica en la fase de Construcción dentro del flujo de trabajo de Implementación, porque esta prueba consiste en medir el código del sistema y es en esa fase donde se puede poner en práctica dicha prueba, la prueba de caja negra se aplica dentro de la fase de transición, cuando el software está terminado ya, porque esta prueba es para medir la funcionalidad del producto, apoyándose en las diferentes interfaces del mismo, esta prueba es muy eficiente, porque la interfaz es lo primero que el usuario ve y prueba del sistema, y a través de ella se pueden corregir los diferentes errores en la funcionalidad del producto.

La prueba de Integración se hace cuando algunos módulos estén terminados, es decir, se puede aplicar en cualquier fase de desarrollo, esta prueba solo depende de la terminación de 2 módulos o más para poder ponerse en práctica, con ella se infiere la integración de los diferentes módulos que pueda tener la aplicación en específico, lo cual es muy importante porque el vínculo entre módulos es lo que le da, en conjunto con la prueba de caja negra, la total funcionalidad al sistema.

Las pruebas anteriormente explicadas tienen una particularidad, que son pruebas destinadas a comprobar la usabilidad del software.

2.6.9 Entrenamiento

Es de gran importancia realizar una serie de actividades de entrenamiento para que el equipo de aseguramiento de la calidad del proyecto ejecute las actividades propuestas en el Plan de Aseguramiento de la Calidad, motivo por el cual se proponen las siguientes:

Capítulo 2 Propuesta de Solución

- ❖ Curso de capacitación sobre los temas de aseguramiento de la calidad. El propósito que se tiene con este curso es que el equipo obtenga mayores conocimientos del tema y así lograr una mejor preparación del equipo.
- ❖ Estudio de las normas internacionales de calidad, con el objetivo de elevar los conocimientos del equipo.
- ❖ Estudio de las diferentes métricas y su empleo.
- ❖ Investigación detallada de los roles de RUP. El objetivo de esta tarea es identificar según cada rol las tareas que se deben realizar, para hacer una mejor distribución del trabajo.
- ❖ Curso de capacitación para la confección de las listas de chequeos. Se procura que se vayan creando las herramientas necesarias para la realización de este tipo de trabajo.
- ❖ Curso de capacitación para la realización de revisiones y auditorías, y la confección de listas de chequeos.
- ❖ Curso de capacitación para el diseño de casos de prueba La meta de esta actividad es proporcionar los conocimientos indispensables para realizar de forma satisfactoria las pruebas a los diferentes módulos.

2.7 Conclusiones

En el presente capítulo se hizo la propuesta del plan para el aseguramiento de la calidad, se explicó paso a paso las actividades necesarias para asegurar la calidad de los diferentes sistemas que se desarrollan en el proyecto Servicios Comunitarios, desde el inicio hasta el final.

Capítulo 3 Evaluación de la Propuesta

Capítulo 3 Evaluación de la Propuesta

3.1 Introducción

Para la validación de la propuesta presentada en el Capítulo 2, se utilizó Criterio de un Panel de Expertos, y para guiarse se utilizaron las técnicas que propone el Método Delphi. Este panel está conformado por especialistas en la calidad del software y el aseguramiento de la calidad.

Una Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

En esta técnica se realiza una selección del grupo de expertos que participará en el proceso de evaluación, teniendo en cuenta que ningún experto conoce la identidad y las respuestas individuales de los otros que componen el grupo. Se debe hacer una buena elección de los expertos, para que el resultado sea el mejor.

En el presente capítulo se hará un breve descripción de cómo se llevara a cabo el proceso de elección de los expertos y los resultados obtenidos.

PROCESO DELPHI

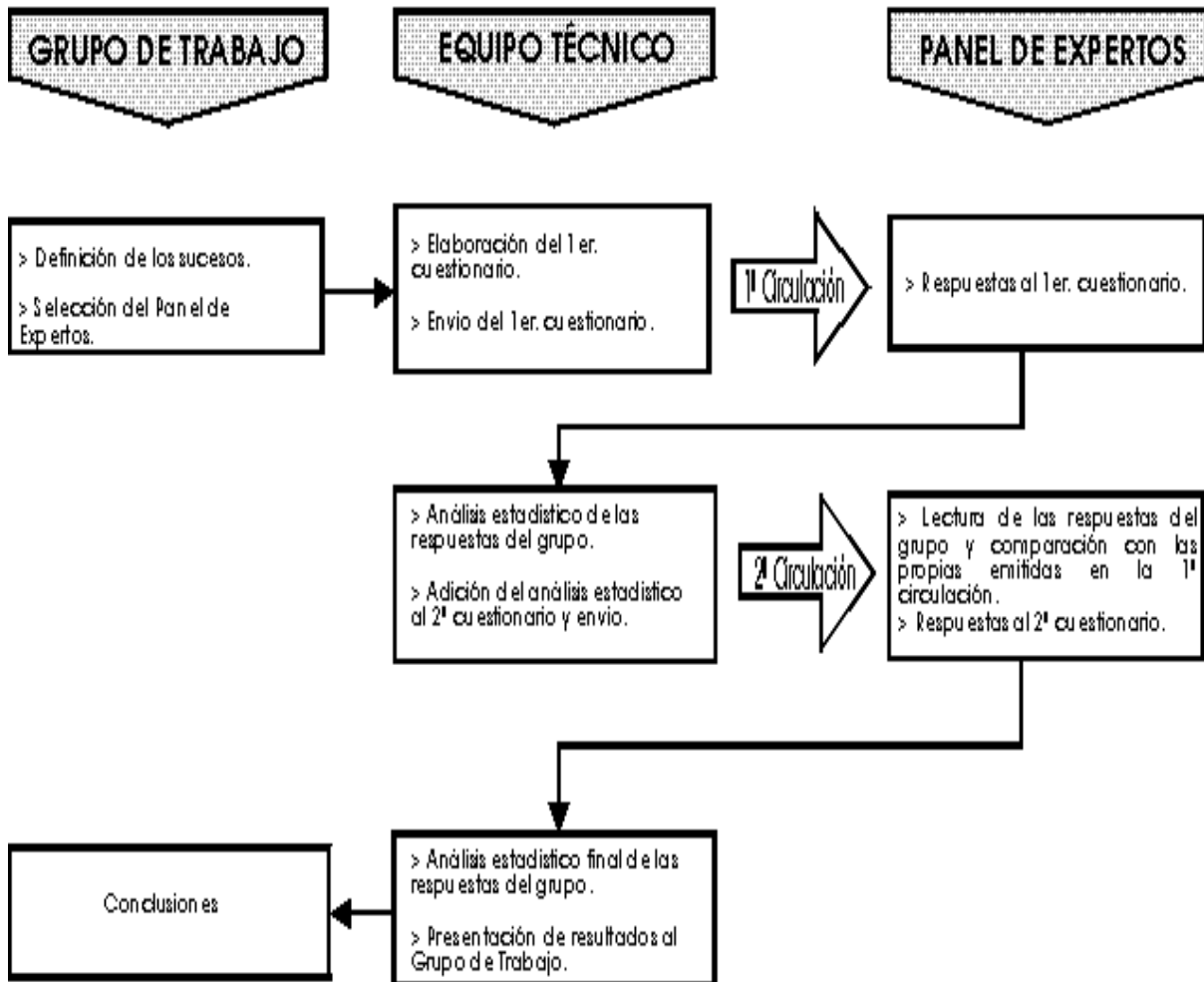


Figura 3.1 Método Delphi

Capítulo 3 Evaluación de la Propuesta

3.2 Elección de los expertos

Se entiende por experto a un especialista en materia. Persona que posee una gran experiencia en el tema y es capaz de ofrecer valoraciones conclusivas del tema y hacer recomendaciones al respecto.

Para el procedimiento de la elección de los expertos se debe llevar a cabo los siguientes pasos:

3.2.1 Determinar la cantidad de expertos

Para la determinación de la cantidad de expertos, no existe una norma generalizada que determine un número óptimo. Se recomienda como mínimo de siete expertos y un máximo de 30.

En este trabajo se decidió contar con un número de siete expertos para la confección del panel, teniendo en cuenta el nivel de complejidad y profundidad del contenido.

3.2.2 Confeccionar el listado de expertos

La confección del listado de expertos se realizó teniendo en cuenta las posibilidades reales de participación de los candidatos, pues todos son profesionales de la UCI y tienen experiencia en la docencia y en el proceso productivo de la Universidad, poseen además un elevado conocimiento en temas relacionados con el proceso a evaluar:

Aseguramiento de la Calidad del Software.

Plan de Aseguramiento de la Calidad del Software.

Métricas

Pruebas

Capítulo 3 Evaluación de la Propuesta

Proceso de revisiones.

3.2.3 Confirmar la participación de los candidatos

Una vez confirmado el listado se invitó personalmente a cada experto para participar en la evaluación. Al ser confirmada su participación, se estableció el listado final de los expertos, informando a cada especialista su inclusión en el proceso a evaluar y las instrucciones necesarias para contestar las preguntas. De esta forma culmina el proceso de selección, logrando la participación de los expertos escogidos.

3.3 Elaboración de los cuestionarios

Para la elaboración del cuestionario se tuvieron en cuenta los objetivos que debería cumplir el procedimiento propuesto para su implantación en el proyecto Servicios Comunitarios.

La encuesta establece una serie de preguntas de enfoque investigativo, que permitieron ver la posibilidad real de que pueda ser aplicada la propuesta, según las características actuales del proyecto productivo Servicios Comunitarios. (Ver anexo 1)

3.4 Resultados de la evaluación

Después de procesada la encuesta se determinó que todos los expertos estuvieron de acuerdo que es importante la elaboración de un plan para el aseguramiento de la calidad de los proyectos. El por qué de esa respuesta, prevaleció en que todo plan permite organizar y controlar el proceso, y velar por la calidad del producto a desarrollar desde sus inicios.

Todos los expertos consideraron que el plan propuesto para el proyecto Servicios Comunitarios puede tener efectividad en el aseguramiento de la calidad del mismo, pues recoge los principales aspectos a tener en cuenta para la obtención de un producto con calidad y propone realizar pruebas durante todo el ciclo de vida del software y otras actividades para el control de la calidad, como las revisiones y auditorías.

Capítulo 3 Evaluación de la Propuesta

Las actividades propuestas en el plan son las necesarias para realizar el aseguramiento de la calidad, pues son las que se tienen en cuenta en todo los procesos de control de la calidad por las principales empresas que norman esta actividad, garantizan el éxito del proyecto y el buen desarrollo del sistema a lo largo de todo el ciclo de vida, además cubren todas las fases y son de varios tipos.

Los elementos expuestos a favor de la propuesta fueron los siguientes:

- ❖ El plan tiene una organización bien determinada de cada uno de los aspectos y propone de forma clara cómo y quién desarrolla cada una de las actividades.
- ❖ Logra organización en el control de la calidad.
- ❖ Involucra a todos los miembros del proyecto en el aseguramiento de la calidad.
- ❖ Cumple con las características presentes en el plan propuesto por la DCS y aborda todos los elementos presentes en el mismo.
- ❖ Propone métricas y revisiones a lo largo de todo el desarrollo del software.

Se recomendó ampliar el conjunto de herramientas propuestas en aras de automatizar procesos y actividades como puede ser el desarrollo de pruebas. El plan se debe poner en práctica dentro del proyecto Servicios Comunitarios.

En la tabla que se muestra a continuación se exponen las actividades propuestas en el PSQA, y los criterios de los expertos, con un 1 si consideran que es de gran importancia realizar esa actividad para el aseguramiento de la calidad y con un 0 en las que creen menos importantes.

Capítulo 3 Evaluación de la Propuesta

Principales actividades propuestas en el plan para el SQA	Exp	Exp	Exp	Exp	Exp	Exp	Exp	Total
	1	2	3	4	5	6	7	
Uso de diferentes métricas	1	1	1	0	1	0	1	5
Pruebas propuestas a desarrollar	1	1	1	1	1	1	1	7
Revisiones y auditorías propuestas	1	1	1	1	1	1	1	7
Actividades de entrenamiento para el equipo de SQA	0	1	1	0	1	1	1	5

Tabla 4 Prioridad de las actividades

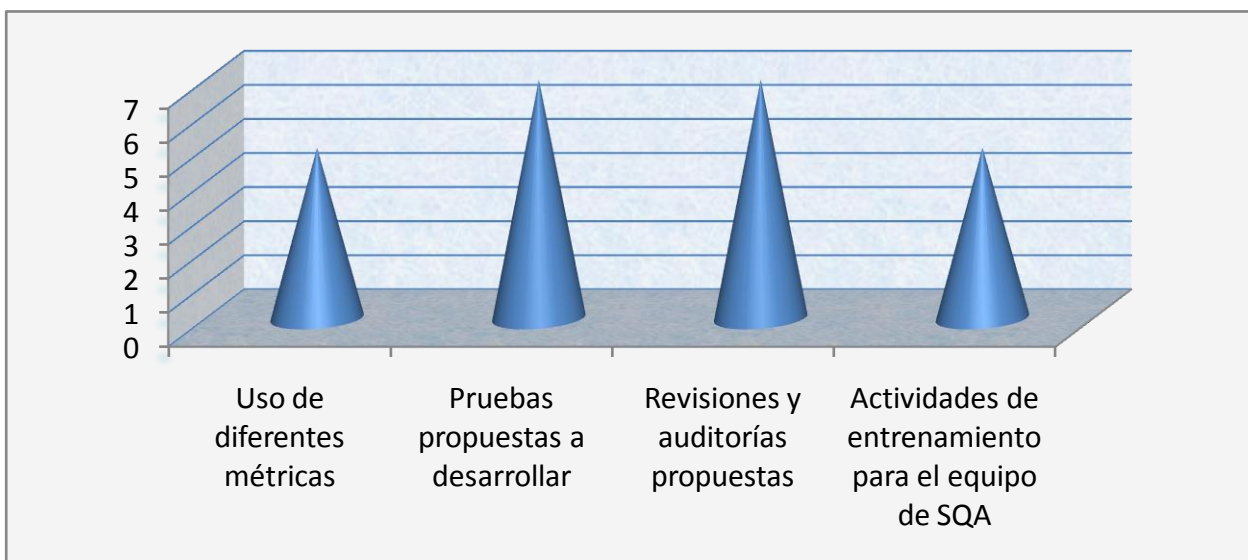


Figura 3.2 Coincidencia de los expertos en actividades más importantes.

Capítulo 3 Evaluación de la Propuesta

Como se puede apreciar los expertos coincidieron que las actividades propuestas de mayor importancia son las revisiones y auditorías y la aplicación de pruebas, pero no por eso las otras dejan de ser menos importantes.

3.5 Conclusiones

Durante el transcurso de la evaluación de la propuesta, mediante los expertos encuestados, se pudo llegar a la conclusión que el plan puede ser aplicado dentro del proyecto Servicios Comunitarios, pues propone las actividades más importantes para llevar a cabo el aseguramiento de la calidad.

Conclusiones

Después de elaborada y evaluada la propuesta de Plan para el Aseguramiento de la Calidad del Software del proyecto Servicios Comunitarios, se llegaron a las siguientes conclusiones:

- ❖ El plan propuesto es adaptable, objetivo y con amplias posibilidades de aplicación.
- ❖ El plan propuesto es efectivo porque propone una serie de actividades que son las más importantes para llevar a cabo el aseguramiento de la calidad.
- ❖ Todos los expertos encuestados coincidieron que las actividades propuestas son las necesarias para realizar el aseguramiento de la calidad.
- ❖ El proceso de Aseguramiento de la Calidad del Software representa un factor importante dentro del proceso de desarrollo de un proyecto asegurando que este cumpla con todos los requisitos y expectativas del cliente.
- ❖ El Plan de Aseguramiento de la Calidad es de suma importancia para lograr la calidad del producto final.
- ❖ Con la puesta en práctica de la propuesta realizada en este trabajo se logrará el Aseguramiento de la Calidad del proyecto Servicios Comunitarios.

Recomendaciones

- ❖ Se le recomienda al equipo de aseguramiento de la calidad del proyecto Servicios Comunitarios dar seguimiento a todas las actividades del plan propuesto durante todo el ciclo de vida del proyecto.

BIBLIOGRAFÍA

Bibliografía Referenciada

1. *MEJORES PRÁCTICAS PARA EL ESTABLECIMIENTO Y ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE*. Vega Lebrún, Carlos, Rivera Prieto, Laura Susana and García Santillán, Arturo. España : s.n., 2008.
2. Neuland Agüero, Dennis. <http://calidadsoft.prod.uci.cu/articulos/articulo-SQA.pdf>. [Online] 2008.
3. S. Pressman, Roger. *"Ingeniería del Software. Un enfoque práctico"*.
4. ISO 9001 - Norma de Calidad (Gestión de la Calidad o Excelencia). http://www.buscarportal.com/articulos/iso_9001_gestion_calidad.html. [Online] 2008.
5. *Conferencia # 2 de Gestión de Software. Introducción al Proceso de Software Personal PSP0 - Línea base del proceso personal.* . 2008.
6. SCALONE, L. F. *ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE*. Tutor: Martínez, D. R. G. Tesis de maestría, UNIVERSIDAD TECNOLÓGICA NACIONAL, FACULTAD REGIONAL BUENOS AIRES, 2006.
7. Antonio, Angelica. LA GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE. http://is.ls.fi.upm.es/udis/docencia/plani/G_Configuracion.pdf. [Online] 2008.

Bibliografía Consultada

1. Aseguramiento de la Calidad, Metodología MÉTRICA Versión 3, Ministerio de Administraciones Públicas.
2. Brito Reyes, Irina, Napal Torres, Irina. La pruebas de software, su aplicación al Config.CASE, 2003
3. calidadsoft.prod.uci.cu. [En línea] <http://calidadsoft.prod.uci.cu/2008>
4. CMMI for Development, Version 1.2, 2006.
5. COLLADO, M. (2003). "Pruebas de software. Técnicas de prueba del software. Estrategia de prueba del software." Consultado: 17 de febrero de 2007, 2007.Disponible en: <http://lmi.ls.fi.upm.es/ftp/ed2/0203/Apuntes/pruebas.ppt>
6. Franco Navarro, José Ángel. **Entorno Unificado para la Gestión de Configuración de Software**. Tesis para optar por el Título de Máster en Informática Aplicada.2006
7. Granja Alvarez, Juan Carlos. Controles y Métricas, Técnicas del Software. 2007.
8. Grosso, Luis Alberto. Calidad de Producto de Software, taller de Calidad de Software, 2006.
9. <http://es.wikipedia.org/wiki/CMMI> 2008
10. http://www.12manage.com/methods_helmer_delphi_method_es.html 2008
11. <http://www.gtlic.ssr.upm.es/encuestas/delphi.htm> 2008
INGENIERO INFORMÁTICO, 2002.
12. ISO/IEC 90003: 2006, DIRECTIVAS PARA LA APLICACIÓN DE LA NC ISO 9001:2001 AL SOFTWARE DE COMPUTACIÓN (ISO/IEC 90003:2004, IDT), 2006.
13. ISO/IEC 9126, Software engineering –Product Quality, Parte 3: Internal Metrics.2003.
14. ISO/IEC 9126-1: 2005, INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO— PARTE 1: MODELO DE LA CALIDAD (ISO/IEC 9126-1:2001, IDT), 2005.
15. ISO/IEC 9126-1: 2005, INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD (ISO/IEC 9126-1:2001, IDT), 2005.
16. ISO-IEC 12119: 2005, TECNOLOGÍA DE LA INFORMACIÓN—PAQUETES DE ISO 12119: 1994, IDT, 2005.

Bibliografía

17. Marcelo Rizzi, Francisco, Tesis de Magister en Ingeniería del Software “Sistema Experto Asistente de Requerimientos” 2001.
18. Monsalve, Luis. Calidad de los Productos Software, <http://www.inf.udec.cl/revista/edicion1/lmonsalve>
19. Pérez Estévez, Isabel. MÉTRICAS PARA EL CONTROL DE PROYECTOS DE SOFTWARE, TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
20. Personal Software Process for Engineers: Part II, Software Quality, 2006.
21. Procedimiento Proceso de Pruebas al Producto de Software, Laboratorio de Liberación (CALISOFT).
22. René Houle, Fundación Ciutat de Viladecans, Aseguramiento de la calidad y acreditación, ISO 9000:2000. Una herramienta de competitividad en la Nueva Economía
23. Rodríguez, Moises. Kemis: Entorno para la Medición de la Calidad del Software, España, 2007.
24. SOFTWARE—REQUISITOS DE CALIDAD Y ENSAYOS / PRUEBAS
25. USAOLA, D. M. P. Mantenimiento Avanzado de Sistemas de Información. Pruebas del Software [Sitio Web]. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/masi/doc/lec/parte5/polo-apuntesp5.pdf.2008>

ANEXOS

Anexo 1

Cuestionario realizado a los expertos

Encuesta

Usted ha sido seleccionado por su conocimiento en calidad de software y los resultados alcanzados en su labor profesional, como experto para evaluar la propuesta realizada en este trabajo investigativo.

1-¿Considera usted que es importante la elaboración de un plan para lograr el Aseguramiento de la Calidad del Software?

Si _____ No _____ ¿Por qué?

2- ¿Considera usted que la aplicación del plan propuesto en el proyecto Servicios Comunitarios pueda tener efectividad en el Aseguramiento de la Calidad del Software?

Si _____ No _____ ¿Por qué?

3- ¿Cree usted que las actividades propuestas en el plan son las necesarias para realizar el Aseguramiento de la Calidad?

Si _____ No _____ ¿Por qué?

Si cree preciso proponer o eliminar alguna menciónela o explíquela brevemente.

4-¿Cuáles elementos usted expondría a favor de la propuesta y cuales en contra?

5- Realice algún comentario o aporte sobre la propuesta que es por su parte objeto de evaluación.

Le agradecemos por su valiosa colaboración y estamos seguros que sus sugerencias contribuirán a perfeccionar el proceso propuesto.

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

Anexo 2

Caracterización de los Expertos

Experto	Graduado de	Años vinculados a la UCI	Experiencia en el tema
1	Ingeniero en Ciencias Informáticas	1	Especialista de la Dirección de Calidad
2	Ingeniero en Ciencias Informáticas	1	Ingeniería de Software.
3	Ingeniero en Ciencias Informáticas	1	Especialista de la Dirección de Calidad(Laboratorio de liberaciones)
4	Ingeniero Informático	3	Especialista de la Dirección de Calidad(Grupo de aseguramiento de la calidad)
5	Ingeniero Informático	4	Especialista de la Dirección de Calidad
6	Ingeniero Informático	3	Ingeniería de Software, miembro del proyecto Calidad

Anexos

			de la facultad 1.
7	Ingeniero en Ciencias Informáticas	1	Ingeniería de Software, miembro del proyecto Calidad de la facultad 1.

Glosario de Términos

GLOSARIO DE TÉRMINOS

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.

Aseguramiento o garantía de la calidad: Todas aquellas acciones planificadas y sistemáticas que proporcionan una confianza adecuada en que un producto o servicio cumpla determinados requisitos de calidad.

Auditoría: Inspección 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.

Calidad del software: Es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Calidad: Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer una necesidad explícita o implícita.

CMMI: Modelo de capacidad de madurez integrada.

CU: casos de uso

DCS: Dirección de Calidad del Software.

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

IEC: Comisión Electrotécnica Internacional.

IEEE: (Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos) Asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas

Glosario de Términos

tecnologías, como ingenieros eléctricos, ingenieros en electrónica, ingenieros en sistemas e ingenieros en telecomunicación.

IPPD: Integración de productos y procesos en desarrollo.

ISO: Organización Internacional de Normalización.

Modelos de calidad: Son un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Los modelos de calidad te dicen QUÉ hacer, no CÓMO hacerlo, porque depende de las metodologías que uses y de tus objetivos del negocio.

PSP: Proceso Personal de Software (Personal Software Process).

PSQA: Plan de Aseguramiento de la Calidad del Software.

Releases: liberaciones de un producto de software.

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

RTF: Revisiones Técnicas Formales.

SQA: Aseguramiento de la calidad del software.

SW: Software.

Stakeholders: Representa los intereses del grupo de personas que necesita el producto (Clientes).

TSP: Proceso de Software en Equipo (Team Software Process).