

Universidad de las Ciencias Informáticas

Facultad 3



"Definición del proceso Desarrollo de Requisito del Modelo de Calidad Cubano para el Desarrollo de Aplicaciones Informáticas, para la disciplina Ingeniería de Requisitos en la Industria Cubana del Software"

Trabajo de diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autores: Alianis Hernández Trujillo

Raiko Alfonso Tápanes

Tutor: Ing. Kariné Ramos

Ciudad de la Habana, Junio de 2011

*"Solamente aquel que contribuye al futuro
tiene derecho a juzgar el pasado. "*

Friedrich Wilhelm Nietzsche

CONTACTO

Ing. Kariné Ramos Blanco

Graduada de Ingeniero en Ciencias Informáticas en 2007.

Pertenece al Centro de Calidad para soluciones tecnológicas (CALISOFT) en el grupo de Normalización y Métricas.

Dentro del Programa de Mejora que se está desarrollando actualmente en la Universidad cumple las siguientes funciones:

Es líder de los TWG (Grupos técnicos de trabajo) de Administración de la Configuración y Planeación de Proyecto y pertenece al TWG de Administración de Requisitos.

Pertenece al grupo EPG (Grupo de Ingeniero de Procesos).

Dedicatoria

Dedicado a Familiares, amigos, y a mi abuelito Jesús.

Raiko Alfonso Tápanes

Agradecimientos

A mi mamá, por ser la razón de mi existir, por no dejarme nunca solo, por su amor, sus noches de desvelo.

A mi hermana Roxana, el amorcito de mi vida, por los sueños que me regala a diario, por sus ocurrencias interminables.

A papá Juanito, por dejarme ser su hijo varón, por poner en mis venas la sangre de su espíritu.

A papá Osvaldo, por darme siempre un abrigo para mis sueños, por darme siempre los consejos que no pedía, pero tu sabías que yo necesitaba, por no divorciarte de mí.

A Niurys y Geydi, por su atención, su preocupación, por hacer de mi un hijo más en tu casa.

A mis abuelos, por sus rezos de cada día, por protegerme tanto.

A mi abuelo Jesús, que no puedo ver su sueño hecho realidad pero que estará siempre en cada logro de mi vida.

A mis amigos, a los que nunca me fallaron, a los que estuvieron en las buenas y en las malas, por su apoyo, por ser familia cuando estaba lejos de la mía.

A mi tutora, por sus regaños que tanto nos ayudaron, por su optimismo, por su tiempo, sus conocimientos, por confiar en nosotros.

A mi incansable compañera de tesis, por su trabajo, su fuerza, por su mano femenina para cada detalle, sin ti, este trabajo nunca hubiera sido lo que es.

Raiko Alfonso Tápanes

Dedicatoria

A mis padres y a mi hermano Eduardo que son las personas que nunca me han abandonado y realmente siempre están ahí para mí bajo cualquier circunstancia. Les dedico mi tesis como regalo por todo el esfuerzo que han hecho para que yo pudiera llegar hasta aquí. Los quiero.

Alianis Hernández Trujillo

Agradecimientos

A mis padres (Noemi y Héctor) por la educación que me dieron, por las muchas veces que por darme un gusto dejaron de hacer sus cosas, por siempre estar ahí a pesar de que a veces discutimos, jajaja. Les quiero.

A mi hermano Eduardo que lo adoro aunque nunca se lo digo, realmente es muy especial para mí con sus ocurrencias y sus cosas que a veces me dan ganas de ahorcarlo, jjajaj, gracias por ser mi hermano.

A mi novio que no sé cómo me soporta, jajaja le agradezco el siempre estar ahí para mí, le quiero muchísimo y espero que en un futuro cuando lea los agradecimientos de mi tesis esté todavía a mi lado, bebe I LOVE TO PIPI.

A mis suegros (Isabel y Máximo) que a diferencia de todos los demás suegros, los míos son buenísimos y me ayudan en todo lo que pueden, realmente los quiero mucho y son muy especiales para mí.

A Nancy por ser tan buena conmigo y a Isis por sus ocurrencias porque siempre está conmigo si se trata de vacilar a Robert Pattinson jajajaj (bebe no te pongas bravo)

A mis hermanos (Amaury y Doralys) y mis sobrinos (Daylin y Adrian), que por cuestiones de distancias y otras veces de vagancia, jajja hace alrededor de 4 años que nos conocemos más y realmente me gustaría compartir toda la vida con ellos porque son personas súper buenas, sobre todo mi hermana que es tan optimista, carismática, natural y bella.

A Mireya, Alina, Emilita y Anita, mis súper tías y a mi prima Yusleydi las quiero.

A mis amistades de la UCI, Yoanna, Yalaina, Leodany, Mayte, Marisabeth, Lexy, Nany, Daysi, que han estado conmigo estos años, siempre los tendré presente y los quiero.

A mi tutora, que por dios si no fuera por su persistencia este trabajo nunca se hubiese terminado, jajaj, es una gran profesional.

A mi compañero de tesis por no coger lucha conmigo, gracias...

Alianis Hernández Trujillo

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de la 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_____

Alianis Hernández Trujillo

Raiko Alfonso Tápanes

Firma del Autor

Firma del Autor

Kariné Ramos

Firma del Tutor

RESUMEN

El Modelo de Calidad Cubano para el Desarrollo de Aplicaciones Informáticas (MCCDAI) tiene como objetivo la mejora de los procesos para el desarrollo de productos y servicios. Dentro de estos se encuentra el proceso de Desarrollo de Requisitos, que es de vital importancia para que el producto final cumpla con las funcionalidades establecidas por el cliente, siendo este el éxito de cualquier empresa productora de software, darles a sus clientes lo que realmente desean.

Existen varios modelos de calidad del software a nivel mundial, que no satisfacen las necesidades de las empresas cubanas, haciendo hincapié en sus procesos de Desarrollo de Requisitos, la información que contienen es demasiado detallada y difícil de entender; la explicación de las prácticas no está clara y se requiere de consultorías para lograr una mejor comprensión, trayendo consigo mayor gasto y pérdida de tiempo. Por lo que se hace imprescindible definir un buen proceso de Desarrollo de Requisitos que tribute a la creación del MCCDAI y que establezca los cimientos sólidos con el fin de lograr un buen desarrollo del software.

En esta investigación se presenta una propuesta de la definición del proceso de Desarrollo de Requisitos que define las prácticas, los roles que las realizan y los artefactos que se generan; además, se plantean un grupo de técnicas y herramientas a utilizar para darle cumplimiento a las prácticas y una base de conocimientos, un repositorio de artefactos, que pueden ser reutilizables por los nuevos proyectos con desarrollos similares. Todo siendo validado por un grupo de expertos, mediante la utilización del método Delphi.

Índice

INTRODUCCIÓN	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	18
INTRODUCCIÓN	18
1.1 CMMI.....	18
1.2 MPS.BR.....	23
1.3 MOPROSOFT	26
1.4 ISO 90003	31
1.5 ISO 12207	32
1.6 ISO 15504	33
CONCLUSIONES DEL CAPÍTULO	35
CAPÍTULO 2: PROPUESTA DEL PROCESO DESARROLLO DE REQUISITOS	37
INTRODUCCIÓN	37
DESARROLLO	37
2.1 DESCRIPCIÓN GRÁFICA DEL PROCESO: DESARROLLO DE REQUISITOS.....	38
2.2 DESCRIPCIÓN TEXTUAL DEL PROCESO DESARROLLO DE REQUISITOS.....	39
2.3 PRODUCTOS DE TRABAJO	43
2.4 BASE DE CONOCIMIENTO.....	44
2.5 ROLES DEL PROCESO Y SUS RESPONSABILIDADES.....	45
2.6 TÉCNICAS.....	47
2.6.1 TÉCNICAS DE OBTENCIÓN DE REQUISITOS:.....	47
2.6.2 TÉCNICAS DE MODELADO.....	52
2.6.3 TÉCNICAS DE VALIDACIÓN:.....	55
2.6.4 TÉCNICAS DE PRIORIZACIÓN:	56
2.7 HERRAMIENTAS	57
2.7.1 RATIONAL ROSE: (González, y otros, 2010).....	58
2.7.2 MAGICDRAW: (Corporate and Sales Headquarters, 2011).....	60
2.7.3 VISUAL PARADIGM: (Headquarters, 2011).....	60
2.7.4 ENTERPRISE ARCHITECT (EA) (Systems, 2011).....	61
2.7.5 STAR UML (UML, 2011).....	62
2.8 MAPA DE COMPATIBILIDAD DE LAS ACTIVIDADES DEL PROCESO CON CMMI	63
2.9 MAPA DE COMPATIBILIDAD DE LAS ACTIVIDADES DEL PROCESO CON ISO 90 003.....	64
2.10 MAPA DE COMPATIBILIDAD DE LAS ACTIVIDADES DEL PROCESO CON ISO 12207.....	65
2.11 MAPA DE COMPATIBILIDAD DE LAS ACTIVIDADES DEL PROCESO CON ISO 15504.....	66
CONCLUSIONES DEL CAPÍTULO	69
CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA	70
INTRODUCCIÓN	70
3.1 VALIDACIÓN DE LA INVESTIGACIÓN	70
3.2 MÉTODO DELPHI	70
3.3 NIVEL DE COMPETENCIA DE LOS ESPECIALISTAS	70
3.4 EXPLOTACIÓN DE RESULTADOS.....	72
CONCLUSIONES DEL CAPÍTULO	77

CONCLUSIONES GENERALES	78
RECOMENDACIONES	79
BIBLIOGRAFÍA	80
ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 1: LISTAS DE NECESIDADES DEL CLIENTE	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 2: GLOSARIO DE TÉRMINOS	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 3: DESCRIPCIÓN DEL NEGOCIO	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 4: ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 5: MODELO DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 6: DOCUMENTO DE ANÁLISIS DE REQUISITOS	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 7: DOCUMENTO DE ARQUITECTURA DEL SOFTWARE	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 8: REGISTRO DE LOS RESULTADOS DE LA VALIDACIÓN	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 9: ENCUESTA REALIZADA A LOS EXPERTOS	¡ERROR! MARCADOR NO DEFINIDO.
GLOSARIO	¡ERROR! MARCADOR NO DEFINIDO.

INTRODUCCIÓN

El auge de las Tecnologías de la Información y las Comunicaciones (TIC) ha cambiado los paradigmas y estrategias reconocidos y establecidos por muchos años como válidos. Dentro de las TIC, la industria del software alcanza una posición relevante, por su característica de controlar o hacer accesible, en la mayoría de los casos, los adelantos electrónicos. Se han informatizado diversos sectores de la sociedad, pero a pesar de los adelantos alcanzados, todavía la comprensión de los requisitos de un problema está entre las tareas más difíciles que enfrenta un ingeniero de software, se tiene dificultades cuando se trata de obtener requisitos de los clientes y también al comprender la información que se adquiere. Con frecuencia se registran los requisitos de una manera desorganizada y se invierte muy poco tiempo en verificar lo que se registra (Pressman, 2005). En fin no se establecen correctamente los cimientos para un buen desarrollo del software.

Los problemas antes mencionados representan un reto, y cuando se combinan la imagen es desalentadora incluso para los profesionales del software más expertos, una posible solución a este problema es la Ingeniería de Requisitos que proporciona un enfoque sólido para abordar dichos desafíos.

“Un error o laguna en la recogida de los requisitos va a tener un impacto en el costo y tiempo de desarrollo del proyecto mucho mayor que otro que se produzca en fases posteriores de desarrollo, prueba o implantación” (Hofmann, y otros, 2001).

“El aumento de la fracción del presupuesto total dedicado a los requisitos en un grupo de proyectos de la NASA llevó a una disminución sustancial de la extralimitación de los costos y horarios” (Hooks, y otros, 2001).

Costo y Horario propasados en algunos proyectos de la NASA

Porcentaje del presupuesto gastado en los requisitos	Número de Proyectos	Costo promedio de saturación del proyecto
--	---------------------	---

Menos de 5	5	125%
5 a 10	7	83%
Más de 10	6	30%

Dedicar tiempo y recursos adecuados a los requisitos puede acelerar el desarrollo de varias maneras, como por ejemplo, reduce en el costo promedio del proyecto y el tiempo de ejecución del mismo. Además reduce la posibilidad de que los usuarios rechacen el nuevo sistema en la entrega. También, hacer hincapié en el desarrollo de los requisitos es más barato que confiar en las pruebas beta para encontrar problemas en los mismos.

El establecimiento de modelos de calidad, que abordan el tema de los requisitos como parte importante, cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Internacionalmente Cuba tiene una escasa imagen como país productor de software. De acuerdo a Riera (2004) no fue hasta la 10ª Convención y Feria Internacional Informática 2004, con la asistencia de alrededor de 1 600 delegados de 37 países, que se comenzó a crear cierta imagen para el mercado externo, a través de la presentación internacional de Incusoft (Industria Cubana del Software), que fue creada con el objetivo de aunar los esfuerzos individuales que han venido realizando diversas instituciones del país en este campo para alcanzar una fortaleza que permita incursionar, con más efectividad, en los mercados extranjeros. Algunas empresas del sector del software en Cuba son: Desoft, Avante, Softel, (Vismar Santos Hernández, 2009) la Universidad de las Ciencias Informáticas (UCI), entre otras.

El surgimiento de la UCI es una prueba evidente del progreso de Cuba en el desarrollo de software, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación, garantizando la innovación continua genera y aporta valor a los productos

y servicios. Brinda los servicios de calidad de software, arquitectura y tecnología, servicios legales y diseño de comunicación visual. (UCI, 2011)

La UCI cuenta con 15 centros, de los cuales el Centro de Gobierno Electrónico (CEIGE), el Centro de Informática Médica (CESIM) y el Centro de Informática Industrial (CEDIN) optan por alcanzar una evaluación internacional del nivel 2 de Capability Maturity Model Integration (CMMI), según datos publicados en por el Centro rector de la calidad en la UCI, quien está llevando a cabo este Programa de Mejora (Calisoft, 2010)

En Cuba la Oficina Nacional de Normalización (ONN) tiene como misión el mejoramiento continuo de la producción y los servicios en función del crecimiento de la economía nacional y la elevación de la calidad de vida de la población. La ONN está representada por el organismo cubano Normas Cubanas (NC), entidad designada oficialmente como órgano nacional de certificación de la República de Cuba, encargada de representar al país en las actividades de normalización, metrología y calidad; así como programar, organizar y controlar la participación de la economía nacional en los trabajos de elaboración y adopción de normas de organismos, entre los que se encuentra la ISO. Según los datos de la ONN, ninguna empresa cubana productora de software está certificada por las NC referentes a la calidad del software. (Ministerio de Ciencia, Tecnología y Medio Ambiente, 2011)

Se escogió a la empresa Softel como muestra, para determinar si se regía por algún modelo de calidad del software al realizar el proceso de Ingeniería de Requisitos, pero dicha empresa se rige solo por los conocimientos de sus especialistas, tienen como ambición futura regirse y certificarse por la norma ISO 9001:2001 referente a la calidad, pero plantean que la misma no tiene definido los roles que realizan las actividades, tampoco genera los artefactos de cada actividad.

El estudio realizado sobre el proceso de Ingeniería de Requisitos de los modelos de calidad del software existentes, como: CMMI, MOPROSOFT, MPS.BR y las normas ISO referentes a la calidad del software, arroja como resultado que no se satisfacen las necesidades de las empresas cubanas, porque no son procesos completos, es decir, no contienen los roles que realizan las actividades, ni los artefactos generados, la explicación de las actividades es demasiado detallada y difíciles de entender y no están en un lenguaje de fácil comprensión por lo que se requiere de consultorías trayendo consigo mayor gasto monetario y de recursos, pérdida de tiempo y desorden al aplicar un proceso que no contiene todo lo necesario para realizarse.

Teniendo en cuenta las dificultades antes mencionadas, el presente trabajo tiene como **problema a resolver**: ¿Cómo mejorar el proceso de Ingeniería de Requisitos que contribuya a la creación del Modelo de Calidad Cubano para el Desarrollo de Aplicaciones Informáticas? Por lo que constituye el **objeto de estudio**: el proceso de Ingeniería de Requisitos de los modelos de Calidad de software, siendo el **campo de acción** el Desarrollo de Requisitos.

Para dar respuesta al problema antes planteado se tiene como **objetivo general**: definir un proceso de desarrollo de requisitos de calidad de software que tribute a la creación del Modelo de Calidad Cubano para el Desarrollo de Aplicaciones Informáticas (MCCDAI).

Siendo los **objetivos específicos**:

- Elaborar el marco Teórico de la investigación.
- Definir un proceso de Desarrollo de Requisitos.
- Validar el proceso de Desarrollo de Requisitos.

Para alcanzar el objetivo propuesto se dará cumplimiento a las siguientes **tareas** de investigación:

- Caracterización de los modelos existentes de calidad de software en el mundo actual del desarrollo de software.
- Comparación de los modelos existentes de calidad de software en el mundo actual del desarrollo de software.
- Selección de los mejores modelos existentes de calidad de software en el mundo actual del desarrollo de software.
- Definición de técnicas de apoyo a la realización de las actividades definidas en el proceso.
- Proposición de herramientas que den soporte a las actividades definidas en el proceso.
- Elaboración de la propuesta de los productos de trabajo esperados como resultado de las actividades del proceso.
- Definición de los productos de trabajo que deben quedar establecidos para formar parte de la Base de Conocimiento de la organización.

- Definición de los roles involucrados en el proceso y las responsabilidades dentro de este.
- Elaboración del diagrama de flujo para las actividades definidas en el proceso.
- Elaboración de un mapa para establecer la compatibilidad de las actividades del proceso con las normas ISO referentes a la calidad del software y CMMI.

La **idea a defender** de esta investigación queda planteada de la siguiente manera:

Definido un proceso de Desarrollo de Requisitos de calidad de software para la empresa cubana de software, se contribuye a crear el MCCDAI.

Para el desarrollo del presente trabajo se hace uso de los **métodos científicos** de investigación: método teórico y método empírico.

Métodos teóricos:

Analítico - Sintético: Utilizando este método se analiza la teoría, bibliografía, documentos, entre otros, para extraer los elementos más importantes relacionados con el Desarrollo de Requisitos en los modelos de calidad ya existentes.

Análisis histórico-lógico: Mediante este método se estudia la trayectoria histórica del proceso Desarrollo de Requisitos en los modelos de calidad de software existentes.

Inductivo-deductivo: Mediante formas de razonamiento se llega a conocimientos relacionados sobre el proceso Desarrollo de Requisitos, basado en las prácticas realizadas por este proceso y las propuestas que se puedan encontrar en el estudio del estado del arte.

Métodos empíricos:

Observación: Se utiliza la observación para la recogida de información referente a los modelos de calidad de software.

Encuesta: Se preparan encuestas, y mediante preguntas se recogen las principales actividades del proceso Desarrollo de Requisitos, y se les entrega a todo aquel que tenga conocimientos sobre el proceso.

El **contenido** del presente trabajo quedará estructurado de la siguiente manera:

- En el capítulo 1 se realiza un estudio del estado del arte de los principales modelos de calidad del software, específicamente en el proceso de desarrollo de requisitos.
- En el capítulo 2 se elabora la propuesta del proceso de Desarrollo de Requisitos.
- En el capítulo 3 se presenta la validación mediante el método Delphi.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En la actualidad existen una serie de modelos para medir la calidad del software, los más significativos son:

- CMMI (Integración de Modelos de Madurez de Capacidades o Capability Maturity Model Integration).
- MPS.BR (Mejora de Proceso del Software Brasileño).
- MoProSoft (Modelo de procesos para la Industria del Software en México).
- Normas ISO referentes a la calidad del software.
 - ✓ ISO 90 003.
 - ✓ ISO 15 504.
 - ✓ ISO 12 207.

Estos modelos obtienen, analizan, especifican y validan los requisitos de forma diferente; pero todos tienen como propósito, en el proceso de Desarrollo de Requisitos, producir y analizar los requisitos del cliente y del producto.

A continuación se mostrará cómo es el proceso de Desarrollo de Requisitos en los modelos antes mencionados.

1.1 CMMI

“Es un modelo de madurez de mejora de los procesos para el desarrollo de productos y servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento.” (Chrissis, y otros, 2009)

Consta de dos representaciones, la representación continua que se centra en la mejora de procesos sobre acciones a completar dentro de las áreas de proceso y la representación escalonada está

enfocada en la mejora de la madurez de los procesos que una organización quiere lograr. (Chrissis, y otros, 2009)

Representación continua

La visión continua de una organización mostrará la representación de nivel de capacidad de cada una de las áreas de proceso del modelo.

La representación continua tiene áreas de procesos que contienen prácticas, que se organizan de manera que soportan el crecimiento y la mejora de un área de proceso individual. Permite cierta libertad a la organización para que seleccione un área de proceso en específico, para mejorar a través de ella. También permite a las organizaciones mejorar varias áreas al mismo tiempo que conforman una categoría, las cuales están en distintos niveles. (Chrissis, y otros, 2009)

Representación escalonada

La representación escalonada está dividida en 5 niveles de madurez (Tabla 1), donde para completar un nivel es necesario implementar los procesos del siguiente nivel y los procesos de niveles superiores pueden no llevarse a cabo si no se tiene la disciplina obtenida en los niveles inferiores, cada nivel sirve como punto de referencia para conocer el grado de madurez total que posee una organización. Cada nivel de madurez tiene un conjunto de áreas de procesos que indican dónde una organización debería enfocar la mejora de su proceso. (Mary Beth Chrissis, 2009)

Nivel	Enfoque	Áreas de Procesos y categoría a la que pertenece
5. Optimizado	Mejora continua de proceso.	OID – Innovación y Despliegue Organizacional. CAR – Análisis Casual y Resolución.
4. Cuantitativamente Administrativo	Control cuantitativo del proceso.	OPP – Desempeño de Procesos Organizacionales. QPM – Administración Cuantitativa de Proyectos.

<p>3. Definido</p>	<p>Proceso de normalización.</p>	<p>RD - Desarrollo de Requisitos.</p> <p>TS - Solución Técnica.</p> <p>PI - Integración de Producto.</p> <p>VER – Verificación.</p> <p>VAL – Validación.</p> <p>OPF - Enfoque al Proceso Organizacional.</p> <p>OPD - Definición de Procesos.</p> <p>OT - Entrenamiento Organizacional.</p> <p>IPM - Administración Integrada de Proyecto.</p> <p>RSKM - Administración de Riesgos.</p> <p>DAR - Análisis de decisiones y soluciones.</p>
<p>2. Administrativo</p>	<p>Administración básica del proyecto.</p>	<p>REQM - Administración de Requisitos.</p> <p>PP - Planeación del Proyecto.</p> <p>PMC - Monitoreo y Control del Proyecto.</p> <p>SAM -Administración de Acuerdos con Proveedores.</p> <p>MA - Medición y Análisis.</p> <p>PPQA - Aseguramiento de la Calidad de Procesos y Productos.</p> <p>CM - Administración de la Configuración.</p>

1.Inicial	Proceso impredecible control reactivo.	
-----------	--	--

Tabla 1.1 Niveles de Madurez de la Representación Escalonada del modelo CMMI. (Mary Beth Chrissis, 2009)

Cada área de procesos agrupa un conjunto de prácticas relacionadas entre sí cuya ejecución permite alcanzar una serie de objetivos o metas. (Chrissis, y otros, 2009). (Figura 1)

La representación escalonada guía a la organización sobre las áreas de procesos que debe de ir abordando, las prácticas que debe implementar y los objetivos que debe alcanzar para ir consiguiendo los sucesivos niveles de madurez. (Chrissis, y otros, 2009).



Figura 1.1 Representación escalonada del modelo CMMI. (Chrissis, y otros, 2009)

Ambas representaciones reconocen que las áreas de procesos se pueden agrupar en 4 categorías generales: Gestión de Proyectos, Gestión de Procesos, Ingeniería y Apoyo y 2 categorías opcionales: Desarrollo Integrado del Producto y del Servicio y Gestión de Compras.

El nivel 3 de CMMI contiene el Área de Proceso Desarrollo de Requisitos la cual consta de 3 objetivos específicos y 10 prácticas específicas (Tabla 2), que son muy importantes para lograr un buen desarrollo de requisitos (Caferino, 2006).

Esta área de proceso suele implementarse mediante la adopción de técnicas de relevamiento y análisis de requisitos, como por ejemplo prototipado, escenarios de operación, casos de uso, etc. Más allá de las técnicas elegidas es importante que se formalice:

- ¿Cómo se obtendrán los requisitos del usuario?
- ¿Cómo se refinarán dichos requisitos hasta obtener los del producto y los correspondientes a sus componentes?
- ¿Cómo se especificarán, analizarán y validarán los requisitos?

Objetivos Específicos	Prácticas Específicas
<p>SG1 Desarrollar Requisitos del cliente.</p> <p><i>Se relevan las necesidades, expectativas, restricciones e interfaces y se traducen en requisitos del cliente.</i></p>	<p>SP 1.1 Relevar Necesidades.</p> <p>SP 1.2 Desarrollar los Requisitos del cliente.</p>
<p>SG2 Desarrollar los Requisitos del Producto.</p> <p><i>Los requisitos del cliente son refinados y elaborados para obtener los requisitos del producto y sus componentes.</i></p>	<p>SP 2.1 Establecer Requisitos del Producto y sus Componentes.</p> <p>SP 2.2 Asignar Requisitos a los Componentes del Producto.</p> <p>SP 2.3 Identificar los Requisitos de Interfaz.</p>
<p>SG3 Analizar y Validar Requisitos.</p> <p><i>Los requisitos son analizados y validados y se desarrolla una definición de la funcionalidad requerida.</i></p>	<p>SP 3.1 Desarrollar Concepto de Operación y Escenarios.</p> <p>SP 3.2 Desarrollar una Definición de la Funcionalidad Requerida.</p> <p>SP 3.3 Analizar Requisitos.</p>

	<p>SP 3.4 Analizar Requisitos para Balancear Necesidades y Restricciones.</p> <p>SP 3.5 Validar requisitos.</p>
--	---

Tabla 1.2 Prácticas del proceso Desarrollo de Requisitos del modelo CMMI. (Caferino, 2006)

Según (Sanches,2007) gerente del grupo Delaware (empresa española de consultoría de procesos de negocio y servicios tecnológicos) plantea que CMMI requiere de un alto esfuerzo de implantación. (Chrissis, 2010) critica fuertemente al modelo CMMI por no ser muy específico en la definición de los procesos; guía a las organizaciones a definir y mejorar sus procesos, indica qué actividades han de realizar, pero no explica sobre cómo hacerlo.

Se concuerda con las críticas hechas al modelo CMMI, además presenta como desventajas que es excesivamente detallado, pudiendo llegar a ser difícil de entender. La descripción de las prácticas, definidas en este proceso, no está clara y esto trae como consecuencias que las organizaciones cubanas que deseen utilizar este modelo de calidad requieran de consultorías que pueden llegar a costar más de 70 mil dólares, por lo que es de mayor inversión que otros modelos.

1.2 MPS.BR

Tiene como meta definir y perfeccionar un modelo de mejora y evaluación de proceso de software, dando preferencia a las micro, pequeñas y medianas empresas, de modo que atiendan sus necesidades de negocio y que sea reconocido nacional e internacionalmente como un modelo aplicable a la industria de software. (SOFTEX, 2009)

El modelo MPS está dividido en tres (3) componentes (Figura 2): Modelo de Referencia (MR-MPS), Método de Evaluación (MA-MPS) y Modelo de Negocio (MN-MPS). Cada componente está descrito por medio de guías y/o de documentos del modelo MPS.

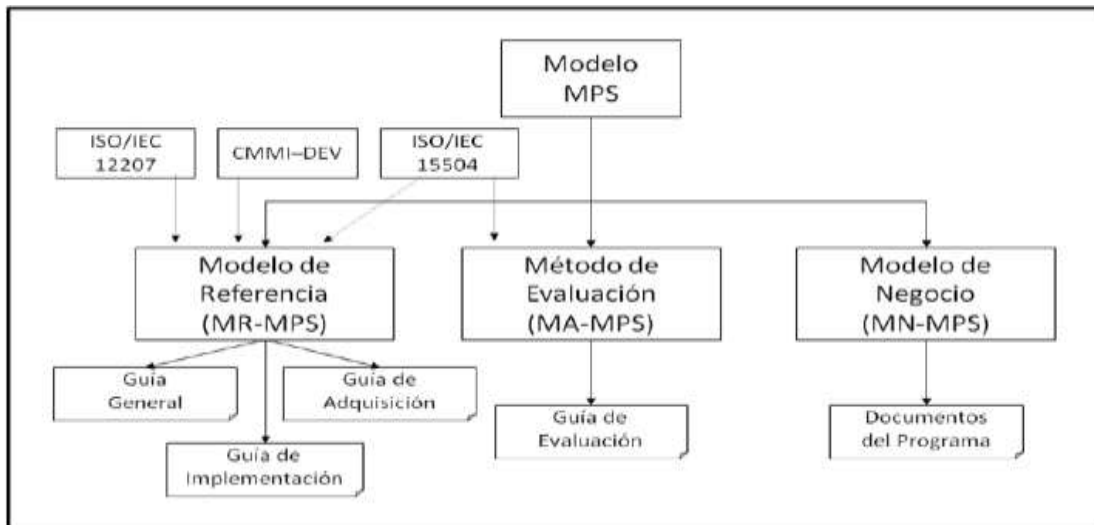


Figura 1.2 Estructura del modelo MPS.BR (SOFTEX, 2009)

El Modelo de Referencia MR-MPS define niveles de madurez que son una combinación entre procesos y su capacidad. El nivel de madurez en que se encuentra una organización permite la previsión de su desempeño futuro al ejecutar uno o más procesos. (SOFTEX, 2009)

Se definen 7 niveles de madurez (figura 3). Para cada uno de estos niveles se atribuye un perfil de procesos que indican dónde la organización debe colocar el esfuerzo de mejora. El progreso y el logro de un determinado nivel de madurez del MR-MPS se obtienen cuando se cumplen los propósitos y todos los resultados esperados de los respectivos procesos y los resultados esperados de los atributos de proceso establecidos para aquel nivel. (SOFTEX, 2009)

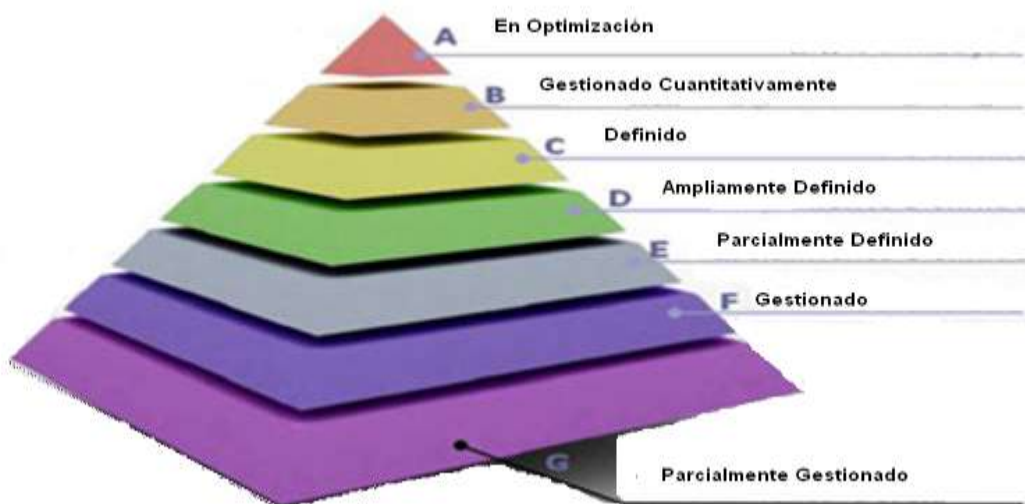


Figura 1.3 Niveles de madurez del modelo MPS.BR. (SOFTEX, 2009)

El pase hacia un nivel de madurez superior, los procesos anteriormente implementados deben pasar a ser ejecutados en el nivel de capacidad exigido en este nivel superior, es decir, los niveles son acumulativos, o sea, si la organización está en el nivel F, esta posee el nivel de capacidad del nivel F que incluye los atributos de proceso de los niveles G y F para todos los procesos relacionados en el nivel de madurez F (que también incluye los procesos del nivel G).

Los procesos están descritos en términos de propósitos y resultados. El propósito describe el objetivo general que debe ser logrado durante la ejecución del proceso. Los resultados esperados del proceso establecen los resultados que deben ser obtenidos con la efectiva implementación del proceso.

La capacidad del proceso es representada por un conjunto de atributos de proceso descrito en términos de resultados esperados. La capacidad del proceso expresa el grado de refinamiento e institucionalización con que el proceso es ejecutado en la organización/unidad organizacional. En el MR-MPS, a medida que la organización/unidad organizacional evoluciona en los niveles de madurez, debe lograr un mayor nivel de capacidad para desempeñar el proceso. (SOFTEX, 2009)

El nivel de madurez D está compuesto por los procesos de los niveles de madurez anteriores (G al E), agregándoles los procesos Desarrollo de Requisitos, Integración del Producto, Diseño y Construcción del Producto, Validación y Verificación. (SOFTEX, 2009)

El Proceso: Desarrollo de Requisitos – DRE tiene como propósito: definir los requisitos del cliente, del producto y de los componentes del producto. (SOFTEX, 2009)

Resultados esperados:

DRE 1. Las necesidades, expectativas y restricciones del cliente, tanto del producto como de sus interfaces, son identificadas.

DRE 2. Un conjunto definido de requisitos del cliente es especificado a partir de las necesidades, expectativas y restricciones identificadas.

DRE 3. Un conjunto de requisitos funcionales y no-funcionales, del producto y de los componentes del producto que describen la solución del problema a ser resuelto, es definido y mantenido a partir de los requisitos del cliente.

DRE 4. Los requisitos funcionales y no-funcionales de cada componente del producto son refinados, elaborados y atribuidos.

DRE 5. Interfaces internas y externas del producto y de cada componente del producto son definidas.

DRE 6. Conceptos operativos y escenarios son desarrollados.

DRE 7. Los requisitos son analizados, usando criterios definidos, para equilibrar las necesidades de los interesados con las restricciones existentes.

DRE 8. Los requisitos son validados.

El modelo brasileño MPS.BR tiene como deficiencia que no define quiénes son los que van a realizar las prácticas, tampoco establece los productos que se generan.

1.3 MOPROSOFT

El modelo de procesos para la industria del software en México fomenta la estandarización de su operación a través de la incorporación de sus mejores prácticas en gestión e ingeniería de software. (Oktaba, 2005)

El modelo está enfocado en procesos y considera los tres niveles básicos de la estructura de una organización que son: la Alta Dirección, Gestión y Operación (Figura 4). Los procesos se documentan en un patrón de procesos que se utiliza como esquema. El patrón de procesos puede ser utilizado para documentar e integrar otros procesos que no fueron contemplados en el modelo.

En los procesos están definidos los roles responsables por la ejecución de las prácticas. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y capacitación para desempeñarlos. Se clasifican los roles en Grupo Directivo, Responsable de Proceso y otros roles involucrados; además se considera al Cliente y al Usuario como roles externos a la organización. (Oktaba, 2005)

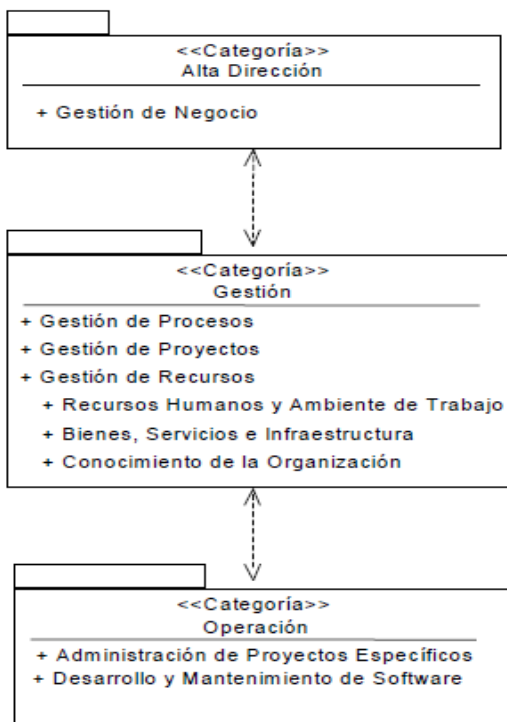


Figura 1.4 Estructura del modelo MOPROSOFT. (Oktaba, 2005)

La Categoría de Alta Dirección (DIR) proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos.

La Categoría de Gerencia (GER) categoría de procesos que aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por estos y comunica los resultados a la Categoría de Alta Dirección. (Oktaba, 2005)

La Categoría de Operación (OPE) es la categoría de procesos que aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y le entrega la información y los productos generados.

La Categoría de Operación contiene entre sus procesos el Desarrollo y Mantenimiento del Software, cuyo propósito es la realización sistemática de las actividades de análisis, diseño, construcción,

integración y pruebas de productos de software nuevo o modificado cumpliendo con los requisitos especificados. (Oktaba, 2005)

El proceso de Desarrollo y Mantenimiento de Software se compone de uno o más ciclos de desarrollo; donde cada ciclo está compuesto por diferentes fases entre las que se encuentra la fase de requisitos, que no es más que un conjunto de actividades, cuya finalidad es obtener la documentación de la Especificación de Requisitos y Plan de Pruebas de Sistema para conseguir un entendimiento común entre el cliente y el proyecto.

La Especificación de Requisitos se compone de una introducción y una descripción de requisitos y tiene como salida la descripción general del software y su uso en el ámbito de negocio del cliente. Esta descripción de requisitos está compuesta por:

Funcionales: Necesidades establecidas que debe de satisfacer el software cuando es usado en condiciones específicas; deben de ser adecuadas, exactas y seguras.

Interfaz con usuario: Definición de aquellas características de la Interfaz de usuario que permiten que el software sea fácil de entender, aprender, que genere satisfacción y con el cual el usuario pueda desempeñar su tarea eficientemente.

Confiabilidad: Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación.

Eficacia: Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos.

Mantenimiento: Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software.

Portabilidad: Descripción de las características del software que permitan su transcendencia de un ambiente a otro.

Restricciones de diseño y construcción: Necesidades impuestas por el cliente.

Legales y reglamentarios: Necesidades impuestas por leyes, reglamentos entre otros.

El analista, el cliente y el usuario junto al diseñador de interfaz de usuario son los encargados de documentar o modificar la Especificación de Requisitos. Tienen como metas: (Oktaba, 2005)

- Identificar y consultar fuentes de información para obtener nuevos requisitos.

- Analizar los requisitos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.
- Elaborar o modificar el prototipo de la interfaz con el usuario.
- Generar o actualizar la Especificación de Requisitos.

Luego siguen unas secuencias de pasos que componen la actividad Realización de la fase de Requisitos. (Tabla 2)

Abreviaturas de los roles: (Oktaba, 2005)

RE: Revisor.

AN: Analista.

DU: Diseñador de interfaz de usuario.

CL: Cliente.

RPU: Responsable de Pruebas.

RM: Responsable de Manuales.

RDM: Responsable de desarrollo y mantenimiento del software.

A2. Realización de la fase Requisitos.	
RDM AN	A2.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al <i>Plan de Desarrollo</i> actual.
AN CL US DU	A2.2. Documentar o modificar la <i>Especificación de Requisitos</i> . Identificar y consultar fuentes de información para obtener nuevos requisitos. Analizar los requisitos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio, del cliente o del proyecto. Elaborar o modificar el prototipo de interfaz con el usuario. Generar o actualizar la <i>Especificación de Requisitos</i> .
RE	A2.3. Verificar la <i>Especificación de Requisitos</i> .

AN CU	A2.4. Corregir los defectos encontrados en la <i>Especificación de Requisitos</i> con base en el <i>Reporte de Verificación</i> y obtener la aprobación de las correcciones.
CL US RPU	A2.5. Validar la <i>Especificación de Requisitos</i> .
AN DU	A2.6. Corregir los defectos encontrados en la <i>Especificación de Requisitos</i> con base en el <i>Reporte de Validación</i> y obtener la aprobación de las correcciones.
RPU AN	A2.7. Elaborar o modificar <i>Plan de Pruebas de Sistema</i> .
RE	A2.8. Verificar el <i>Plan de Pruebas de Sistema</i> .
RPU	A2.9. Corregir los defectos encontrados en el <i>Plan de Pruebas de Sistema</i> con base en el <i>Reporte de Verificación</i> y obtener la aprobación de las correcciones.
RM	A2.10. Documentar la versión preliminar del <i>Manual de Usuario</i> o modificar el manual existente.
RE	A2.11. Verificar el <i>Manual de Usuario</i> .
RM	A2.12. Corregir los defectos encontrados en el <i>Manual de Usuario</i> con base en el <i>Reporte de Verificación</i> y obtener la aprobación de las correcciones.
RDM	A2.13. Incorporar <i>Especificación de Requisitos</i> , <i>Plan de Pruebas de Sistema</i> y <i>Manual de Usuario</i> como líneas base a la <i>Configuración de Software</i> .
RDM	A2.14. Elaborar el <i>Reporte de Actividades</i> registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.

Tabla 1.3 Actividades de la fase Requisitos del modelo MOPROSOFT. (Oktaba, 2005)

Según la máster en Tecnología de la Información (Zúñiga , 2009) el modelo MoProSoft define actividades de manera muy general; requiere de CMMI para asegurar la calidad de los productos o procesos; además define los procesos por áreas de responsabilidad, sin embargo la secuencia de las actividades se da naturalmente entre áreas.

Las actividades referentes a los requisitos, se encuentran en la categoría de operación, en el proceso de Administración de Proyectos específicos se describen las acciones para administrar los requisitos, y en el de Desarrollo y Mantenimiento de Software las acciones para desarrollarlos.

En este modelo encontramos que en esta categoría de operación se unifican de una forma muy compacta las actividades de varias disciplinas como las relacionadas con pruebas, administración de la configuración, elementos técnicos de implementación y diseño, planificación y monitoreo y además la administración y desarrollo de requisitos en solo 2 procesos, esto dificulta la inserción paulatina de las actividades a estandarizar (madurez) en las organizaciones. Como parte de los procesos se detallan un conjunto de elementos importantes para la orientación de los proyectos como los roles involucrados en cada actividad, los indicadores para medir la ejecución del proceso, identifican los productos de trabajo resultado de las actividades, entre otros. Pero pasan por alto un elemento importante: una propuesta de técnicas que le permitan al proyecto realizar correctamente las actividades que las necesitan.

1.4 ISO 90003

Esta Norma Internacional, proporciona una guía a las organizaciones para la aplicación de la Norma ISO 9001:2000 a la adquisición, suministro, desarrollo, operación y mantenimiento del software. (ISO, 2006)

La ISO 90003 plantea una serie de requisitos relacionados con el producto para lograr una mejora en la realización del producto. (ISO, 2006)

1. Determinación de los requisitos relacionados con el producto:

La organización debe determinar:

- Los requisitos especificados por el cliente, incluyendo los requisitos para las actividades de entrega y las posteriores a la misma.
- Los requisitos no establecidos por el cliente pero necesarios para el uso especificado o para el uso previsto, cuando sea conocido.

- Los requisitos legales y reglamentarios relacionados con el producto.
 - Cualquier requisito adicional determinado por la organización.
2. Revisión de los requisitos relacionados con el producto.
- La organización debe revisar los requisitos relacionados con el producto. Esta revisión debe efectuarse antes de que la organización se comprometa a proporcionar un producto al cliente (por ejemplo: envío de ofertas, aceptación de contratos o pedidos, aceptación de cambios en los contratos o pedidos) y debe asegurarse que:
 - a) están definidos los requisitos del producto.
 - b) están resueltas las diferencias existentes entre los requisitos del contrato o pedido y los expresados previamente.
 - c) la organización tiene la capacidad para cumplir con los requisitos definidos.
 - Deben mantenerse registros de los resultados de la revisión y de las acciones originadas por la misma.
 - Cuando el cliente no proporcione una declaración documentada de los requisitos, la organización debe confirmar los requisitos del cliente antes de la aceptación.
 - Cuando se cambien los requisitos del producto, la organización debe asegurarse de que la documentación pertinente sea modificada y que el personal correspondiente sea consciente de los requisitos modificados.

3. Comunicación con el cliente.

La organización debe determinar e implementar disposiciones eficaces para la comunicación con los clientes, relativas a:

- a) La información sobre el producto.
- b) Las consultas, contratos o atención de pedidos, incluyendo las modificaciones.
- c) La retroalimentación del cliente, incluyendo sus quejas.

1.5 ISO 12207

Esta norma internacional tiene 3 amplias categorías: Primario, Soporte y Organizacional.

Dentro de los procesos primarios se encuentran: Adquisición, Suministro, Operación, Mantenimiento y Desarrollo. (JTC1, 1995)

El proceso Desarrollo consta de un conjunto de actividades entre las cuales se encuentra:

Análisis de los requisitos del sistema.

Análisis de los requisitos del software.

Análisis de los requisitos del sistema consta de las siguientes tareas: (JTC1, 1995)

- Se deberá analizar el uso específico previsto del sistema a ser desarrollado para especificar los requisitos del sistema, que deberá describir funciones y capacidades del sistema.
- Requisitos del negocio, organizativos y de usuario.
- Requisitos de seguridad física y de acceso.
- Requisitos de ingeniería de factores humanos, interfaces y requisitos de operación y mantenimiento.
- Limitaciones de diseño y requisitos de calificación. Se deberá documentar la especificación de requisitos del sistema.

Análisis de los requisitos del software: Para cada elemento software el desarrollador deberá establecer y documentar: (JTC1, 1995)

- Especificaciones funcionales y de capacidad, incluyendo prestaciones, características físicas y condiciones del entorno en donde el elemento software ha de funcionar.
- Interfaces externas al elemento software.
- Requisitos de calificación.
- Especificaciones de seguridad física.
- Especificaciones de seguridad de acceso.
- Especificaciones relacionadas con ingeniería de factores humanos (ergonomía).
- Definición de datos y requisitos de la base de datos.
- Requisitos de instalación y aceptación del producto software entregado, en el lugar o lugares de operación y mantenimiento.

Documentación de usuario.

Requisitos de operación y ejecución por parte del usuario.

Requisitos de mantenimiento por parte del usuario.

1.6 ISO 15504

La ISO 15 504 determina la capacidad de mejora del proceso del software. Tiene una arquitectura basada en dos dimensiones: de proceso y de capacidad de proceso. Dentro de los procesos primarios

se encuentra el Grupo de Procesos de Ingeniería, el cual está compuesto por 12 procesos y sus actividades. (/N3302, 2005)

Los procesos son:

- ENG.1** Obtención de requisitos.
- ENG.2** Análisis de los requisitos del sistema.
- ENG.3** Diseño arquitectónico del sistema.
- ENG.4** Análisis de los requisitos del software.
- ENG.5** Diseño del software.
- ENG.6** Construcción del software.
- ENG.7** Integración del software.
- ENG.8** Pruebas del software.
- ENG.9** Integración del software.
- ENG.10** Pruebas del sistema.
- ENG.11** Instalación del software.
- ENG.12** Mantenimiento del software y sistema.

Los procesos del 1 al 4 contienen actividades que pertenecen al proceso de Desarrollo de Requisitos, las cuales son: (/N3302, 2005)

- ENG.1** Obtención de requisitos.
 - **BP1** Obtener requisitos y pedidos del cliente.
 - **BP2** Comprender las expectativas del cliente.
 - **BP3** Estar de acuerdo con los requisitos.
 - **BP4** Establecer la línea base de los requisitos del cliente.
 - **BP5** Gestionar los cambios de los requisitos del cliente.
 - **BP6** Establecer mecanismos para consultas con el cliente.

- ENG.2** Análisis de los requisitos del sistema.
 - **BP1** Establecer los requisitos del sistema.
 - **BP2** Optimizar la solución del proyecto.
 - **BP3** Analizar los requisitos del sistema.
 - **BP4** Evaluar y actualizar los requisitos del sistema.
 - **BP5** Asegurar consistencia.

- **BP6** Comunicar los requisitos del sistema.

ENG.3 Diseño arquitectónico del sistema.

- **BP1** Describir la arquitectura del sistema.
- **BP2** Asignar requisitos.
- **BP3** Definir interfaces.
- **BP4** Verificar la arquitectura del sistema.
- **BP5** Evaluar alternativas para la arquitectura del sistema.
- **BP6** Asegurar la consistencia.
- **BP7** Comunicar el diseño de la arquitectura del sistema.

ENG.4 Análisis de los requisitos del software.

- **BP1** Especificar los requisitos del software.
- **BP2** Determinar el impacto del ambiente operacional.
- **BP3** Desarrollar un criterio para la prueba del software.
- **BP4** Asegurar consistencia.
- **BP5** Evaluar y actualizar los requisitos del software.
- **BP6** Comunicar los requisitos del software.

Según (López, 2009) las normas ISO referentes a la calidad requieren de un gran esfuerzo y tiempo para lograr sus objetivos, también el modelo origina cierta burocracia y se necesitan demasiados recursos, por lo que resulta costoso.

Se coincide con las críticas mencionadas anteriormente, pero también hay que mencionar que las normas ISO no tienen bien definido el proceso de Desarrollo de requisitos, al no plantear qué roles realizan las actividades, ni los artefactos que generan las mismas, en fin no definen bien cómo realizar las actividades del proceso.

CONCLUSIONES DEL CAPÍTULO

El estudio realizado permitió conocer los modelos más relevantes a nivel mundial, destinados a establecer calidad en el software, se mantuvo el enfoque en estudiar el proceso de Desarrollo de Requisitos de estos modelos; usando bibliografía especializada en el tema.

Se pudo observar que cada modelo tiene su propia forma de ejecutar el proceso de Desarrollo de Requisitos pero, de forma general, todos presentan un punto en común, y es que desarrollan los requisitos del cliente y del producto, aunque sea por distintas vías.

Se puede concluir que el proceso Desarrollo de Requisitos de los modelos estudiados no está completo desde el punto de vista que no hay un proceso que tenga definido, dentro de sí mismo, los roles, las actividades con sus explicaciones, los artefactos que genera cada actividad y las técnicas y herramientas necesarias que sirvan de ayuda al realizar las actividades, esto puede provocar atraso y desorden a la hora de aplicar esos procesos en las empresas cubanas.

CAPÍTULO 2: PROPUESTA DEL PROCESO DESARROLLO DE REQUISITOS

INTRODUCCIÓN

El proceso de Desarrollo de Requisitos contiene mecanismos apropiados para comprender lo que el cliente desea, analizar sus necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades y validar la especificación de requisitos. (Pressman, 2005) Realizar este proceso bien, implica la obtención de un software con calidad.

En el presente capítulo se realiza una propuesta del proceso de Desarrollo de Requisitos mediante una descripción gráfica y textual. Se proponen técnicas y herramientas que sirven de apoyo en la realización de las actividades del proceso además de una base de conocimientos.

DESARROLLO

El proceso Desarrollo de Requisito que se define a continuación asegura el cumplimiento de las buenas prácticas de los modelos de calidad de software estudiados y se pretende mejorar las deficiencias de cada modelo.

El proceso está integrado por:

- Roles: son los responsables de realizar las actividades propuestas en el proceso.
- Actividades: son las que guían el proceso y contienen entradas y salidas.
- Entradas: son los artefactos generados por las actividades realizadas con anterioridad.
- Salidas: son los artefactos que genera la actividad en ejecución.
- Control: son las técnicas, herramientas y planillas que se utilizan para realizar la actividad.

En el capítulo se realiza una descripción gráfica que define el proceso y una descripción textual que detalla cómo realizar las actividades del proceso. El proceso cuenta con un conjunto de técnicas y herramientas opcionales que serán de ayuda al equipo de desarrollo para la realización de las actividades propuestas.

El proceso asegura una compatibilidad con modelos como CMMI y las normas ISO para garantizar que las empresas que implementen nuestro proceso puedan, si están interesadas en algún momento, en obtener una evaluación CMMI o una certificación ISO con muy poco esfuerzo.

2.1 Descripción Gráfica del Proceso: Desarrollo de Requisitos.

Proceso de Desarrollo de Requisitos				
Criterios de Entrada		Se da inicio al proyecto		
Criterios de Salida		Requisitos desarrollados		
Roles	Entrada	Control	Actividades	Salida
			Inicio	
-Cliente. -Usuario. -Analista.		-Plantilla Lista de necesidades del cliente. -Técnicas de obtención de requisitos	1. Obtener necesidades, expectativas y restricciones del cliente.	-Glosario de Términos. -Lista de necesidades del cliente.
			2. ¿Son suficientes? (SI) [NO]	
-Analista.	-Lista de Necesidades del cliente.	-Plantilla Descripción del Negocio. -Técnicas de modelado	3. Descripción del negocio.	-Descripción del Negocio.
-Analista.	-Descripción del Negocio.	-Plantilla Especificación de requisitos del software.	4. Elaborar y refinar los requisitos.	-Especificación de requisitos del software.
-Analista.	-Lista de Necesidades del cliente. -Especificación de requisitos del software. -Descripción del Negocio.	-Plantilla Documento de análisis de requisitos. -Técnicas de priorización de requisitos	5. Analizar los requisitos.	-Documento de análisis de requisitos. -Especificación de requisitos del software(actualizado)
-Analista.	-Especificación de Requisitos del Software. -Descripción del Negocio.	-Plantilla Modelo del Sistema. -Técnicas de modelado	6. Establecer los conceptos operativos y los escenarios.	-Modelo del Sistema.
-Analista.	-Modelo del sistema. -Especificación de Requisitos del Software.		7. Análisis de los conceptos operativos y escenarios.	-Documento de análisis de requisitos. (actualizado)
-Arquitecto de Software.			8. Identificar los requisitos de la interfaz.	-Especificación de requisitos del software.(actualizado)
-Analista. -Arquitecto de Software. -Jefe de proyecto.	-Modelo del Sistema. -Especificación de Requisitos del Software.	-Plantilla Documento de Arquitectura de Software.	9. Distribución de los requisitos según la funcionalidad requerida.	-Documento de Arquitectura de Software.
-Jefe de Proyecto -Analista. -Cliente	-Especificación de Requisitos del Software. -Documento de Arquitectura de Software. -Modelo del Sistema	-Plantilla Registro de los Resultados de la validación. -Plantilla Acta de Aceptación -Técnicas de validación	10. Validar los requisitos.	-Registro de los Resultados de la validación. -Acta de Aceptación (de la validación)
			11. ¿Aceptada la validación? (SI)	
			Fin	

Tabla 2.1 Descripción gráfica del proceso: Desarrollo de Requisitos.

2.2 Descripción Textual del Proceso Desarrollo de Requisitos.

Proceso Desarrollo de Requisitos		
Criterios de Entrada	Se da inicio al proyecto.	
Criterios de Salida	Requisitos desarrollados.	
N.	Descripción	Salida
1	<p>El analista identifica a los interesados (son los que se benefician directa o indirectamente con el sistema en desarrollo).</p> <p>Los interesados pueden tener diferentes puntos de vista y es necesario categorizar toda la información, identificando áreas en común y áreas en conflicto o inconsistencia.</p> <p>El analista, mediante la utilización de técnicas, obtiene de los interesados (cliente y usuarios) las necesidades, expectativas, restricciones y los requisitos de las interfaces externas.</p> <p>Se lleva una lista de los conceptos asociados al negocio que son utilizados normalmente y que deben de ser del dominio del equipo de desarrolladores.</p> <p>Las técnicas propuestas a utilizar son: tormenta de ideas, entrevista, observación y encuesta. (Para más detalle, ver acápite Técnicas de obtención de requisitos).</p>	<ul style="list-style-type: none"> - Listas de necesidades del cliente.(ver Anexo1) - Glosario de términos. (ver Anexo2)
2	Si se tiene la suficiente información para describir las necesidades, expectativas, restricciones y las interfaces externas, se desarrolla la actividad 3.	

	En caso contrario se vuelve a realizar la actividad 1.	
3	<p>El analista elabora los escenarios del negocio para que se organicen y presenten los detalles importantes que se vinculan con el sistema informático a construir, reforzando la idea de que sea el propio negocio lo que determine los requisitos.</p> <p>Se recomienda como técnicas para la realización de escenarios del negocio: casos de uso del negocio, diagrama de casos de usos del negocio, diagrama de actividades y el diagrama de clases del negocio (Para más detalle, ver acápite técnicas de modelado).</p>	- Descripción del negocio. (ver Anexo3)
4	<p>El analista traduce las necesidades, expectativas y restricciones del cliente, tanto del producto como de sus interfaces, en requisitos del cliente documentados. Para esto clasifica los requisitos del cliente en requisitos funcionales y no funcionales, y describe dichos requisitos.</p> <p>Para que los requisitos estén bien documentados es necesario resolver conflictos entre los proveedores de los requisitos, por lo que se recomienda utilizar la estrategia establecida por Pressman: (Para más detalle, ver acápite Técnicas de obtención de requisitos)</p> <p>El analista refina los requisitos utilizando técnicas de modelado que son refinadas continuamente para obtener nuevos requisitos.</p> <p>Se recomienda como técnicas de modelado para refinar los requisitos: casos de uso, diagrama de secuencia, diagrama de actividades, entre otros (Para más detalle,</p>	- Especificación de requisitos del software. (ver Anexo4)

	ver acápite técnicas de modelado).	
5	<p>El analista analiza las listas de necesidades del cliente para eliminar conflictos y las organiza por temas. Analiza los requisitos para determinar si satisfacen los objetivos de los clientes y asegurarse de que sean completos, factibles, realizables y verificables.</p> <p>Identifica los requisitos claves que tienen una fuerte influencia en el coste, calendario, funcionalidad, riesgo o rendimiento, dándoles una priorización que se agrega en el documento Especificación de Requisitos del Software.</p> <p>Se recomienda para identificar los requisitos claves el uso de técnicas de priorización (Para más detalle, ver acápite técnicas de priorización).</p>	<ul style="list-style-type: none"> - Documento de análisis de requisitos. (ver Anexo6) - Especificación de requisitos del software.(actualizado)
6	<p>El analista desarrolla los escenarios y los conceptos operativos de cada escenario para definir qué es lo que debe de hacer el sistema.</p> <p>Se recomienda como apoyo para modelar los escenarios del sistema los diagramas de casos de uso del sistema (Para más detalle, ver acápite técnicas de modelado).</p> <p>Los conceptos operativos y el desarrollo de escenarios es un proceso iterativo, por lo que las revisiones deberían de mantenerse periódicamente para asegurar que están de acuerdo con los requisitos.</p>	<ul style="list-style-type: none"> - Modelo del sistema. (ver Anexo5)
7	<p>Los conceptos operativos y los escenarios se realizan para determinar cómo va a funcionar el sistema y es necesario analizarlos a fondo.</p>	<ul style="list-style-type: none"> - Documento de análisis de requisitos.

	<p>El analista analiza los conceptos operativos y los escenarios para refinar las necesidades, restricciones y las interfaces externas del cliente, y para descubrir nuevos requisitos.</p> <p>Analiza el equilibrio entre las necesidades y las restricciones de las partes interesadas.</p> <p>Se recomienda para la realización de este análisis el uso de modelos, simulaciones o prototipos.</p> <p>Realiza y documenta una evaluación de los riesgos sobre los requisitos, para luego examinar su impacto. Estos riesgos estarán documentados en el documento de análisis de requisitos.</p>	
8	<p>El arquitecto de software identifica las interfaces internas del producto; luego desarrolla los requisitos para las interfaces identificadas, ejemplo de requisitos de interfaz interna se encuentran: las librerías, lenguajes de programación, framework, servidores, entre otros.</p>	- Especificación de requisitos del software.(actualizado)
9	<p>El arquitecto de software en conjunto con el analista y el jefe del proyecto dividen los requisitos en grupos (p.ej., funcionalidad similar, rendimiento o acoplamiento), para facilitar y enfocar la funcionalidad del producto.</p> <p>Realizan una descripción de lo que se pretende que haga el producto, puede incluir acciones, secuencias, entradas, salidas u otra información que comunique la</p>	- Documento de arquitectura del software.

	manera en la cual el producto será usado.	
10	<p>El analista, en conjunto con el cliente, valida los requisitos para asegurar que el producto cumpla con las funcionalidades establecidas.</p> <p>Se recomienda como técnicas para validar: Listas de chequeo, auditorías, prototipos y Revisiones. (Para más detalle, ver acápite técnicas de validación).</p> <p>El cliente, si está de acuerdo con los resultados obtenidos en el proceso de validación, procede a firmar el documento Acta de Aceptación (de la validación).</p>	<ul style="list-style-type: none"> - Registro de los resultados de la validación. (ver Anexo7) - Acta de Aceptación (de la validación)
11	<p>Si el cliente firma el documento Acta de Aceptación, entonces culmina el proceso de Desarrollo de Requisitos.</p> <p>Si el cliente no firma es porque el producto no está preparado para funcionar en el entorno previsto y se vuelve a la actividad no.4 "Elaborar y refinar los requisitos".</p>	

Tabla 2.2 Descripción textual del proceso: Desarrollo de Requisitos.

2.3 Productos de Trabajo

Glosario de términos: Lista de concepto asociados al negocio que son comúnmente usados y que deben ser del dominio del equipo de desarrollo para poder modelar el negocio y dar una solución a la problemática encontrada.

Listas de necesidades del cliente: Contiene las necesidades, expectativas y restricciones del cliente, tanto del producto como de sus interfaces externas.

Descripción del negocio: Brinda una visión general de los procesos que existen en un determinado negocio, permitiéndonos comprender a qué se dedica el mismo; describe el ambiente que rodea al producto y las condiciones que deben de estar dadas al iniciar los procesos y al concluir cada uno.

Especificación de requisitos del software: Captura los requisitos de software para el sistema; establecidos en requisitos funcionales, su prioridad y complejidad y las reglas o restricciones que deben de tenerse en cuenta o cumplirse; y en una detallada descripción de los requisitos no funcionales.

Documento de análisis de requisitos: Contiene los defectos de los requisitos y los cambios propuestos a los requisitos para resolver esos defectos; esto pudiera producir nuevos requisitos que quedaran plasmados en este documento. También tendrá una evaluación de los riesgos relacionados con los requisitos y el impacto de esos riesgos.

Modelo del sistema: Define qué es lo que debe de hacer el sistema, mediante la descripción de un flujo de eventos.

Documento de arquitectura del software: Descripción de lo que se pretende que haga el producto, representado por módulos.

Registro de los resultados de la validación: Se registrarán los productos a ser validados y los métodos utilizados para validar, se identifican los problemas y se comparan los resultados de la validación con el rendimiento real previsto para el producto. Este documento asegura el producto cumple o no con las funcionalidades establecidas por el cliente.

Acta de Aceptación: Documento donde queda plasmado que los requisitos cumplen con las funcionalidades establecidas por el usuario y funciona en el entorno previsto, requiere de la firma del cliente.

2.4 Base de Conocimiento

La base de conocimiento es un repositorio de artefactos que pueden ser re-utilizables para realizar un producto nuevo. Cada proyecto deja en la base de conocimiento los artefactos que crean puedan ser

reutilizables en otros software. A continuación mostramos los artefactos que se consideran sean reutilizables:

Glosario de términos: Documento reutilizable por otros proyectos (en caso de que los términos del nuevo proyecto sean similares a los de la base de conocimientos.), le facilita la comprensión al equipo de desarrollo de los conceptos comúnmente utilizados por los asociados al negocio.

Especificación de los requisitos del software: Documento reutilizable por otros proyectos porque al estar establecidos, clasificados (en funcionales y no funcionales) y priorizados los requisitos del cliente se pudiera ahorrar en tiempo la culminación de un proyecto determinado con características similares.

Documento de análisis de requisitos: Documento reutilizable que ayudaría a corregir los defectos de los requisitos de un proyecto; además de que obtendría una evaluación de los riesgos relacionados con los requisitos y el impacto de los riesgos, en caso de que los requisitos del nuevo proyecto sean similares.

2.5 Roles del proceso y sus responsabilidades.

Rol	Responsabilidades	Habilidades
Analista.	Captura, documenta y prioriza los requisitos del cliente.	Razonar lógicamente. Identificar conexiones entre ideas. Ver los aspectos de un problema. Ir a las fuentes originales de la información básica. Obtener el núcleo del problema. Conocimientos sobre técnicas de priorización.

		Buena ortografía.
	Describe el negocio y el funcionamiento del sistema.	Interpretar toda la información con precisión. Conocimiento y habilidad en las diferentes técnicas de modelación de escenarios.
	Analiza los requisitos.	Habilidades en la utilización de técnicas de verificación.
	Valida los requisitos y documenta los resultados de la validación.	Habilidades en la utilización de técnicas de validación.
	Aprueba las técnicas que serán usadas durante la desarrollo de requisitos.	Conocimiento de las técnicas utilizadas en el proceso de Desarrollo de Requisitos.
Arquitecto de software.	Distribuye los requisitos según la funcionalidad requerida.	Conocimientos sobre la vista lógica de la arquitectura.
	Identifica los requisitos de interfaz interna y externa.	Capacidad para crear un entorno que motive al cliente.
Jefe de Proyecto.	Participa en la distribución de requisitos según la funcionalidad requerida en módulos o subsistemas.	Debe de ser un líder. Tomar decisiones en el momento preciso.
	Participa en la validación de los requisitos.	Tener conocimientos sobre todas las fases del ciclo de vida del producto.
Cliente	Participa en las actividades de obtención de requisitos.	Hablar clara y articuladamente. Desplegar un buen vocabulario.
	Participa en la validación de los requisitos.	Tener conocimiento del problema en cuestión.

Tabla 2.3 Roles del proceso y sus habilidades.

2.6 Técnicas

Las técnicas serán de ayuda al equipo de desarrollo para la realización de las actividades propuestas durante el proceso Desarrollo de Requisitos.

Si el equipo encargado de la realización del proceso cuenta con un conjunto de técnicas que hayan utilizado con anterioridad y tengan dominio de las mismas, entonces podrían utilizarlas durante la realización de las actividades propuestas en el proceso.

A continuación se describirán un conjunto de técnicas que se han puesto en práctica en proyectos productivos de la UCI y que están referenciadas por bibliografía calificada, estas técnicas podrían servirle de guía al equipo de desarrollo si así lo desea.

2.6.1 TÉCNICAS DE OBTENCIÓN DE REQUISITOS:

Los analistas utilizan técnicas con el fin de recopilar los datos sobre una situación existente. Cada técnica tiene ventajas y desventajas por lo que generalmente se utilizan dos o tres para complementar el trabajo de cada una y ayudar a asegurar una investigación completa. A continuación se explicarán algunas de ellas.

Tormenta de ideas: Extraído de (Jeffrey Baumgartner, 2007)

El objetivo al realizar una tormenta de ideas, es “hacer un listado” de todo lo que se nos ocurre que esté relacionado con el tema elegido y que ayudará al momento de comenzar a escribir.

Pautas para realizar esta técnica:

- Elige un tema.
- Establece un tiempo (10 minutos como máximo).
- Durante ese tiempo escribe frases o palabras que creas relacionadas con el tema elegido.
- Lo ideal es, durante el tiempo que escribes, no corregir nada, sino crear ideas.
- Transcurrido el tiempo, lee la lista y trata de reorganizar lo escrito, de modo que tengan sentido y se relacionen con el tema seleccionado.
- Si lo has logrado, comienza a trabajar.

Tormenta de ideas como técnica de grupo:

Es imprescindible la participación espontánea de todos. Esta técnica generará muchas ideas y también soluciones a algún problema determinado. Además se aprovecha la capacidad de todos y no de unos pocos. Genera un sentido de responsabilidad compartido. Para que la lluvia de idea como técnica de grupo sea eficaz, hay que resaltar algunas prioridades:

- Hacer hincapié en la cantidad y no en la calidad de las ideas que se van a expresar.
- Evitar la crítica de las ideas presentadas.
- Dar a conocer las ideas en crudo sin elaborarlas con anticipación.
- Aceptar todas las ideas y no rechazar ninguna por más disparatada que parezca.
- Un análisis posterior, clasificará, seleccionará, cuantificará y explotará las ideas expuestas para lograr la solución buscada.
- Es conveniente generar un ambiente relajado, libre de calificaciones a medida que se dan a conocer las ideas. Esto es muy necesario, para no inhibir la creatividad de los participantes.

Otra forma de llevar a cabo esta técnica grupal, es en forma anónima. Se plantea el problema o tema y se coloca una especie de buzón en donde se colocarán las ideas, esto permitirá reflexionar sobre la idea y guardar el anonimato, lo que otorgará mayor libertad de expresión. Luego se sacan las ideas y se trabajan en grupo.

Observación: Extraído de (Técnicas Recolección de Datos, 2010)

¡Ver es creer! Observar las operaciones le proporciona al analista hechos que no podría obtener de otra forma.

Recopilación de datos mediante la observación:

Leer en relación con una actividad del negocio le proporciona al analista una dimensión de las actividades del sistema. Entrevistar personal, ya sea directamente o a través de cuestionarios, también le ayuda y le dice algo más. Ninguno de los dos métodos da una información completa.

La observación proporciona información de primera mano en relación con la forma en que se llevan a cabo las actividades. Las preguntas sobre el uso de documentos, la manera en la que se realizan las tareas y si ocurren los pasos específicos como se pre-establecieron, pueden contestarse rápidamente si se observan las operaciones.

Cuándo observar:

La observación es muy útil cuando el analista necesita ver de primera mano cómo se manejan los documentos, cómo se llevan a cabo los procesos y si ocurren los pasos especificados. Saber qué buscar y cómo guiar su significado, también requiere de experiencia. Los observadores con experiencia captan quién utiliza los documentos y si encuentran dificultades; también están alertas para detectar documentos o registros que no se utilizan.

Cuestionario: Extraído de (Técnicas Recolección de Datos, 2010)

Los cuestionarios proporcionan una alternativa muy útil para las entrevistas.

Obtención de datos mediante cuestionarios:

Para los analistas los cuestionarios pueden ser la única forma posible de relacionarse con un gran número de personas para conocer varios aspectos del sistema, pero no es posible observar las expresiones o relaciones de quienes responden a los cuestionarios.

Selección de formas para cuestionarios:

Existen dos formas de cuestionarios para recabar datos: cuestionario abierto y cerrados, y se aplican dependiendo de si los analistas conocen de antemano todas las posibles respuestas de las preguntas y pueden incluirlos. Con frecuencia se utilizan ambas formas en los estudios de sistemas.

Los cuestionarios abiertos se aplican cuando se quieren conocer los sentimientos, opiniones y experiencias generales; son útiles al explorar el problema básico. Los cuestionarios cerrados limitan las respuestas posibles del interrogado; por medio de un cuidadoso estilo en la pregunta, el analista puede controlar el marco de referencia. Los cuestionarios cerrados son el mejor método para obtener información sobre los hechos porque fuerza a los individuos para que tomen una posición y forma de opinión sobre los aspectos importantes.

Etapas en el desarrollo de un cuestionario:

Los cuestionarios bien llevan tiempo y mucho trabajo. La primera consideración se encuentra en determinar el objetivo del cuestionario. ¿Qué datos quiere conocer el analista a través de su uso? El analista define cómo utilizar los cuestionarios, a fin de obtener los hechos, al considerar la estructura más útil para el estudio y la más sencilla de entender por parte de los interrogados. Lleva tiempo desarrollar preguntas bien elaboradas y deben siempre probarse y modificarse, si es necesario, antes de que imprima una forma final y se distribuya.

Encuesta: Extraído de (Stanton , et al., 2009)

La encuesta, una de las técnicas de investigación social más difundidas, se basa en las declaraciones orales o escritas de una muestra de la población con el objeto de obtener información. Se puede basar en aspectos objetivos (hechos, hábitos de conducta, características personales) o subjetivos (opiniones o actitudes).

Tipos de encuesta:

En la encuesta personal se entrevista por separado a cada uno de los individuos que constituyen la muestra. El entrevistador formula las preguntas y anota las respuestas en el cuestionario.

En la encuesta telefónica el contacto entrevistador-entrevistado se realiza a través del teléfono. Este tipo de encuesta da acceso a grupos de población que con otro tipo de encuesta quedarían fuera de la muestra. Puede realizarse, además asistida por ordenador con lo que se facilita el procesamiento de datos.

En la encuesta auto-administrada el propio encuestado lee el cuestionario y anota las respuestas. Puede no estar acompañado del entrevistador (encuesta por correo). Este tipo de encuesta facilita al entrevistado cierta sensación de privacidad y le da más tiempo para meditar las respuestas.

Fases de una encuesta

Formulación del problema: en esta fase se trata de marcar los objetivos que persigue la encuesta.

Diseño de la muestra: en este momento hay que decidir qué sector de la población será el entrevistado (niños, estudiantes, jubilados, mujeres, usuarios de un gimnasio, fumadores...)

Selección del tipo de encuesta: decidirse por un tipo de encuesta u otro dependerá de muchos factores, especialmente relacionados con la disponibilidad de recursos (económicos, materiales y humanos) y de tiempo que se tenga para realizar la investigación.

Diseño del cuestionario: el cuestionario es un listado de preguntas que puede formularse de distintas maneras. A cada pregunta se le ha de asignar un valor para facilitar la codificación o recuento de resultados.

Pre-test: antes de realizar la encuesta es conveniente probar la eficacia y corrección del cuestionario revisando si las preguntas se entienden, si responden al tipo de pregunta elegida (abierta, cerrada), si están bien codificadas, si siguen una secuencia lógica, si las instrucciones del cuestionario son correctas.

Realización de la encuesta: esta fase es propiamente trabajo de campo que va a realizar el entrevistador. Esta es la persona encargada de localizar a los entrevistados, convencerles de que

participen en la investigación, leer las preguntas, tomar nota de las respuestas y comprobar que la respuesta se ajuste a los objetivos de la pregunta.

Codificación de las preguntas: para esta fase suele prepararse una plantilla en el que se hace el recuento de los datos obtenidos en el muestreo y que servirán para analizar los resultados de la entrevista.

Análisis de datos: a la vista de los resultados, hay que realizar un detallado análisis de las respuestas y características de la muestra para llegar a conclusiones.

Redacción de un informe: el proceso finaliza con la redacción de un informe que tienen por objeto difundir los datos obtenidos a través de la técnica de la encuesta. En el informe deben constar los objetivos de que se ha partido, el desarrollo del proceso y las conclusiones obtenidas. Asimismo se adjuntarán como anexos los cuestionarios y las plantillas de codificación.

Estrategia recomendada por: (Pressman, 2005)

Orientar a cada interesado hacer tres listas:

- La primera contiene los objetos que son parte del ambiente que rodea al sistema, otros objetos que el sistema producirá y los objetos que utiliza para realizar sus funciones.
- La segunda contiene los servicios (procesos o funciones), que manipulan o interactúan con los objetos.
- La tercera contiene las restricciones (costo, tamaño, reglas del negocio) y los criterios de rendimiento (velocidad y exactitud).

Luego se orienta una reunión, en la que se selecciona un moderador para que la controle; cada participante expresa sus listas, creando así tres listas combinadas, en las que se eliminan los asuntos redundantes. El moderador coordina el debate; las listas combinadas se reducen, se incrementan o se replantea para desarrollar de manera apropiada el producto o sistema que se desarrollará.

Los integrantes de la reunión se dividen en equipos, trabajan para desarrollar mini especificaciones, una explicación concisa de los elementos de la lista. Cada equipo presenta sus mini especificaciones para comentarlas, se realizan las anulaciones, adiciones y elaboraciones posteriores. Cada integrante del equipo hace una lista de criterios de validación para el producto o sistema y la presenta al equipo, realizando finalmente una lista consensada de criterios de validación entre todos los integrantes de la reunión.

2.6.2 TÉCNICAS DE MODELADO.

Casos de uso: Extraído de (Wesley, et al., 1999)

Un caso de uso representa a un proceso, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores.

Los casos de uso contienen nombre y una descripción breve pero clara y debe de ser fácil de comprender. Los casos de uso se involucran con al menos un actor; son iniciados por actores, e interactúan con actores para realizar las actividades y envía resultados.



Figura 2.1 Representación del Actor y Caso de Uso (Aprendizaje, 2011)

Un caso de uso representa a un proceso, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores. Cada caso de uso trae consigo una descripción textual.

El sentido de la flecha entre el caso de uso y el actor depende de quién inicie la comunicación.

En la estructuración del modelo se consideran 3 tipos de relaciones entre los casos de uso.

- Relación de inclusión.
- Relación de extensión
- Generalización/especialización entre actores y entre casos de uso.

Se puede particionar el diagrama de casos de uso en sub-diagramas más pequeños agrupando casos de usos de acuerdo a criterios establecidos; de manera que se definirían paquetes y estos a su vez podrían relacionarse entre sí. Un paquete es un mecanismo de propósito general para organizar en grupos los elementos.

Un actor se comunica con un caso de uso por múltiples razones:

- Para invocar un caso de uso. Una instancia de un actor siempre invoca una instancia de un caso de uso.
- Para solicitar algún dato almacenado en el sistema, el cual el caso de uso obtiene y presenta al actor.
- Para cambiar el dato almacenado en el sistema mediante el uso de un diálogo con el sistema.

- Para reportar que algo especial ha ocurrido alrededor del sistema y que dicho sistema debe cuidar.

Los casos de uso de comunican con los actores por muchas razones:

- Si algo especial ha ocurrido en el sistema, un actor puede necesitar conocerlo.
- Un caso de uso puede necesitar pedirle a un actor que lo ayude a tomar una decisión si se tienen varias opciones.

Diagrama de Actividades: Extraído de (OMG (Object Management Group), 2000).

Es un grafo (grafo de actividades) que contiene estados en que puede hallarse una actividad. Un estado de actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo y espera la terminación de su cómputo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de actividad dentro del grafo. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente.

Un diagrama de actividades puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas en una organización. La concurrencia se presenta con frecuencia a partir de la agregación, en la cual cada objeto tiene su propio hilo concurrente. Las actividades concurrentes se pueden realizar simultáneamente o en cualquier orden.

Un grafo de actividades es como un organigrama tradicional, excepto que permite el control de concurrencia además de control secuencial: una gran diferencia.

Los diagramas de actividades permiten muchas libertades, por lo que se puede incluir un alto nivel de detalle; pero para lograr una buena realización del diagrama es necesario centrarse en las necesidades del cliente y no moverse hacia el espacio de la solución. En la medida de lo posible se debe utilizar un diagrama de actividades por cada casos de uso y si los caso de uso se modifican los diagramas también deben de modificarse.

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. El diagrama de actividades permite una administración adecuada de la complejidad del proceso que se está modelando mediante el anidamiento de diagramas de actividades como descripción de la secuencia de sub-actividades que se ejecutan dentro de una actividad.

Diagrama de clases: Extraído de (Hitschfeld, 2010)

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Diagrama de Secuencia: Extraído de (Booch, et al., 2008)

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo de la página. La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.

Se muestra un mensaje como una flecha desde la línea de vida de un objeto a la del otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo.

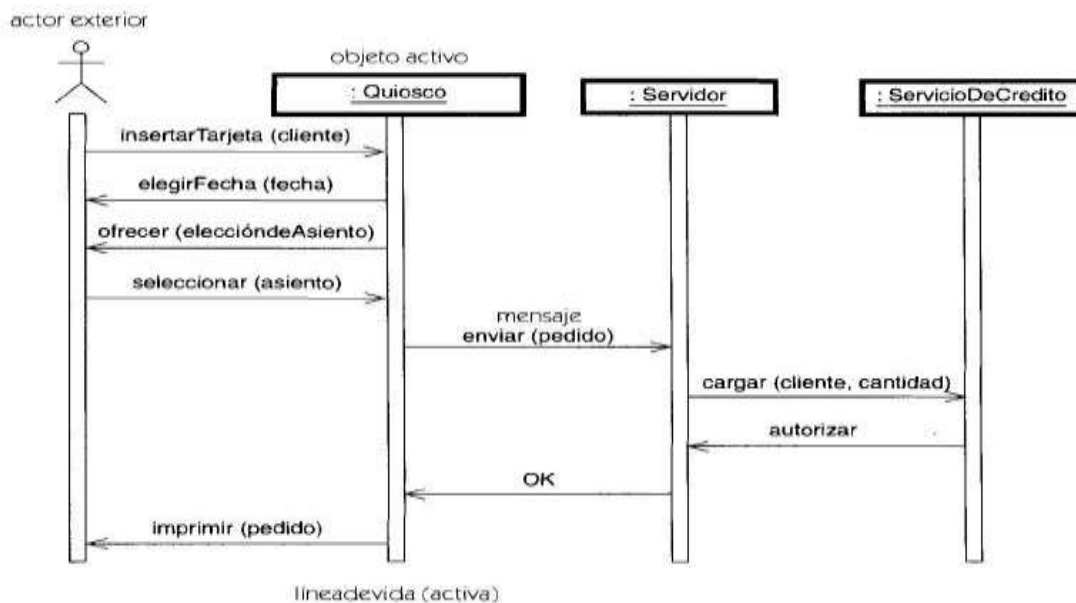


Figura 2.3 Representación del Diagrama de Secuencia: Extraído de (Booch, et al., 2008)

2.6.3 TÉCNICAS DE VALIDACIÓN:

La validación de requisitos es una actividad muy importante, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo.

Reviews (opiniones): Extraído de (Cáceres, 2010)

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.

Revisiones de requisitos: Extraído de (Sommerville, 2005)

Las revisiones de requisitos es un proceso manual que involucra a personas de la organización del cliente y al contratista. Ellos verifican el documento de requisitos en cuanto a anomalías y omisiones. Antes de la revisión de requisitos, el equipo de desarrollo debe <conducir> al cliente a través de los requisitos del sistema, explicándole las implicaciones de cada requisito. El equipo de revisión debe verificar cada requisito para la consistencia además de verificar los requisitos como un todo para la completitud. Los revisores también pueden probar la:

- Verificabilidad: ¿Puede probarse el requisito de modo realista?
- Comprensibilidad: ¿Las personas que adquieren el sistema o los usuarios finales comprenden correctamente los requisitos?
- Rastreabilidad: ¿Está claramente establecido el origen del requisito? Puede tener que volver a la fuente del requisito para evaluar el impacto del cambio.
- Adaptabilidad: ¿Es adaptable el requisito? es decir, ¿puede cambiarse el requisito sin causar efectos de gran escala en los otros requisitos del sistema?

Los conflictos, contradicciones, errores y omisiones en los requisitos deben de ser señalados por los revisores.

Prototipos: Extraído de (UDIS, 2009)

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución. Puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.

Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. En ingeniería de requisitos, los prototipos más comunes son los siguientes:

- **Mock-ups.** Se trata de pantallas, típicamente dibujadas a mano en papel, que representan un aspecto concreto del sistema.
- **Storyboards.** Son una evolución de los mock-ups, ya que además del interfaz, se muestra la secuencia de acciones, o escenarios⁹, que se deben realizar con el programa.
- **Maquetas.** Una maqueta es una versión simplificada del sistema software deseado. Típicamente, una maqueta representa únicamente el interfaz del sistema y, opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones. Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema, de tal modo que además, del interfaz, el software pudiera ofrecer algunos resultados reales. Ello es lo que se conoce como “prototipo funcional”. Las maquetas acostumbran a realizarse mediante programas gráficos de diseño, tales como Visual Basic o PowerBuilder, por citar únicamente dos ejemplos.

Generación de casos de prueba: Extraído de (Sommerville, 2005)

Esta técnica consiste en probar los requisitos; si las pruebas se conciben como parte del proceso de validación, a menudo se revelan los problemas en los requisitos. Si una prueba es difícil o imposible de diseñar, normalmente significa que los requisitos serán difíciles de implementar y deberían ser considerados nuevamente. Desarrollar pruebas para los requisitos del cliente antes de que se escriba el código es una parte fundamental de la programación externa.

2.6.4 TÉCNICAS DE PRIORIZACIÓN:

La priorización es un paso crucial para tomar buenas decisiones en la planificación del proyecto porque varios aspectos de la funcionalidad son considerados, tales como riesgos, costos, etc.

Las técnicas de priorización se clasifican en:

- **Métodos cuantitativos:** Están basados en la asignación de valores cuantitativos a diferentes aspectos de los requisitos. Se llega a decisiones de priorización más rápidamente.

- Enfoque de negociación: Da prioridad a los requisitos logrando un acuerdo entre los diferentes involucrados. Está basado en medidas subjetivas y es más útil cuando las variables de decisión (importancia, costo, riesgo, consecuencias, etc.) están fuertemente relacionadas.

Las técnicas de priorización pueden ser combinadas para la obtención de mejores resultados.

Proceso de Análisis Jerárquico (AHP del inglés Analytical Hierarchy Process): Extraído de (Hurtado, et al., 2010)

El proceso Análisis Jerárquico está diseñado para resolver problemas complejos de criterios múltiples. Requiere que, quien toma las decisiones proporcione evaluaciones subjetivas respecto a la importancia relativa de cada uno de los criterios y que, después, especifique su preferencia con respecto a cada una de las alternativas de decisión y para cada criterio. El resultado es una jerarquización con prioridades que muestran la preferencia global para cada una de las alternativas de decisión.

En un ambiente de certidumbre, el AHP proporciona la posibilidad de incluir datos cuantitativos relativos a las alternativas de decisión. La ventaja consiste en que adicionalmente permite incorporar aspectos cualitativos que suelen quedarse fuera del análisis debido a su complejidad para ser medidos, pero que pueden ser relevantes en algunos casos

Es poco aplicable cuando el número de requisitos es muy elevado.

Asignación Numérica (Agrupamiento): Extraído de (Telecomunicaciones, 2009)

Técnica de priorización más común. Se basa en el agrupamiento de requisitos por sus diferentes niveles de prioridad (Ej.: críticos, estándares, opcionales).

2.7 Herramientas

Las herramientas servirán de apoyo al realizar algunas de las actividades propuestas en el proceso Desarrollo de Requisitos, las mismas deberán cumplir con un conjunto de requisitos necesarios para realizar un buen modelado visual.

Requisitos que deben de cumplir las herramientas:

Crear y visualizar diagramas UML (Lenguaje Unificado de Modelado).

Permitir diversos modelados como son:

- Diagrama de Casos de Uso.
- Diagrama de Actividades.

- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Clases.
- Diagrama de Componentes.
- La capacidad de generar código a partir de los diagramas y viceversa.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- Capacidad de crear, modificar y eliminar los componentes de un diagrama.

Si el equipo de desarrollo cuenta con una herramienta que cumpla con los requisitos antes mencionados, entonces esta puede ser utilizada para realizar las actividades propuestas en el proceso. A continuación se darán a conocer algunas herramientas que cumplen con estas características para que el equipo de desarrollo tenga conocimientos de las mismas, en caso de que desee utilizarlas o no.

2.7.1 RATIONAL ROSE: Extraído de (González, y otros, 2010)

Es una herramienta software para el Modelado Visual mediante UML de sistemas de software, tiene la capacidad de crear, ver, modificar y manipular los componentes de un modelo; además de especificar, analizar y diseñar el sistema antes de codificarlo.

Presenta como características principales el chequeo de la sintaxis UML, genera documentación automáticamente, genera códigos a partir de modelos y crea modelos a partir de código.

Presenta como interfaces de usuario:

- La interfaz Browser que brinda las opciones de:
 - ✓ Navegar por los elementos de las vistas de Rose.
 - ✓ Añadir, borrar, renombrar y mover elementos a los modelos.
 - ✓ Abrir las especificaciones de algún elemento.
 - ✓ Agrupar en paquetes los elementos de los modelos.
 - ✓ Añadir y abrir diagramas.
- La interfaz Document Window que brinda la opción de:
 - ✓ Añadir documentación a un elemento del modelo.
- La interfaz Diagram Window que brinda la opción de:
 - ✓ Crear y visualizar diagramas UML.
- La interfaz Toolbars brinda la opción de:

- ✓ Acceder a los comandos.
- La interfaz Log brinda la opción de:
 - ✓ Ver los resultados de los comandos.

Presenta 4 vistas:

- Vista de Casos de uso, contiene:
 - ✓ Actores.
 - ✓ Casos de uso.
 - ✓ Documentación de casos de uso.
 - ✓ Diagrama de casos de uso.
 - ✓ Diagrama de actividad.
 - ✓ Diagrama de secuencia.
 - ✓ Diagrama de colaboración.
 - ✓ Paquetes.
 - ✓ Actores del negocio.
 - ✓ Trabajadores del negocio.
 - ✓ Casos de uso del negocio.
 - ✓ Diagrama de casos de uso del negocio.
- Vista lógica, contiene:
 - ✓ Diagrama de clases.
 - ✓ Asociaciones.
 - ✓ Interfaces.
 - ✓ Diagrama de secuencias.
 - ✓ Diagrama de colaboración.
 - ✓ Paquetes.
- Vista de componentes, contiene:
 - ✓ Componentes.
 - ✓ Interfaces.
 - ✓ Diagramas de componentes.
 - ✓ Paquetes.

- Vista de despliegue, contiene:
 - ✓ Procesos.
 - ✓ Procesadores.
 - ✓ Conectores.
 - ✓ Dispositivos.
 - ✓ Diagrama de implementación(despliegue (deployment))

2.7.2 MAGICDRAW: Extraído de (Corporate and Sales Headquarters, 2011)

Es una herramienta de modelado con completas características UML. Tiene una interfaz elegante e intuitiva, la mayor parte de las opciones accesibles con un solo click. Presenta ayudas en el diseño con autocompletación y corrección automática en tiempo real. Permite una posible derivación de modelos UML a través de códigos fuente escritos anteriormente. Facilidad y rapidez para el cambio del dominio del modelado. Disponible para un gran número de plataformas y sistemas operativos. Permite diversos modelados entre los que se encuentran: diagrama de caso de uso, de actividades, de secuencia, etc.

2.7.3 VISUAL PARADIGM: Extraído de (Headquarters, 2011)

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. Incluye localización en castellano.

Presenta como desventaja que muestra otras funcionalidades no disponibles como gancho para las versiones de pago; además que no permite su uso en proyectos comerciales e incluye marca de agua recordando este hecho.

- Tiene como características:
 - ✓ Soporte de UML versión 2.1.
 - ✓ Diagramas de Procesos de Negocio
 - ✓ Modelado colaborativo con CVS y Subversion.

- ✓ Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- ✓ Ingeniería de ida y vuelta.
- ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- ✓ Ingeniería inversa Java, C++, Esquemas XML, XML,.NET exe/dll, CORBA IDL.
- ✓ Generación de código - Modelo a código, diagrama a código.
- ✓ Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- ✓ Diagramas EJB - Visualización de sistemas EJB.
- ✓ Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- ✓ Diagramas de flujo de datos.
- ✓ Soporte ORM - Generación de objetos Java desde la base de datos.
- ✓ Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- ✓ Generador de informes para generación de documentación.
- ✓ Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- ✓ Importación y exportación de ficheros XMI.
- ✓ Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- ✓ Editor de figuras.

2.7.4 ENTERPRISE ARCHITECT (EA) Extraído de (Systems, 2011)

EA soporta la generación y la ingeniería reversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi; permite tanto modelar en UML 2.0, como rastrear la información importante de proyecto con artefactos de diseño; ayuda a manejar la complejidad con instrumentos para rastrear dependencias.

Entre sus características principales se encuentra:

- Última especificación UML 2.3.
- Exportación e importación XMI 2.1.

- Reportes HTML y RTF.
- Transformaciones MDA.
- Soporte para Perfiles y Tecnologías.
- Pruebas, gestión de recursos y mantenimiento.

Excelencia en Modelado Visual

Ingeniería inversa de código en más de 10 lenguajes.

Importación de esquemas de bases de datos.

Visualizar código XSD y WSDL.

Importar binarios .NET y Java.

Sencilla y escalable

Para usuarios individuales y grandes equipos.

Soporte para repositorios DBMS.

Carga y uso rápido incluso en grandes modelos.

Modelado en equipo

Archivos compartidos o modelos de repositorio.

Control de versiones con cualquier herramienta SCC compatible.

Seguridad basada en roles incorporada.

2.7.5 STAR UML Extraído de (UML, 2011)

Es un software libre muy personalizable, pero que presenta problemas con la ingeniería inversa si hay caracteres locales, como por ejemplo, las tildes; además que no presenta versión para Linux.

Entre sus características principales están:

- Soporte completo al diseño UML mediante el uso de:
 - Diagrama de casos de uso.
 - Diagrama de clase.
 - Diagrama de secuencia.
 - Diagrama de colaboración.
 - Diagrama de estados.
 - Diagrama de actividad.

- Diagrama de componentes.
- Diagrama de despliegue.
- Diagrama de composición estructural (UML 2.0).
- Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML.
- La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes c++, c# y java.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- Patrones GoF (Gang of Four) , EJB (Enterprise JavaBeans) y personalizados.
- Plantillas de proyectos.
- Posibilidad de crear plugins para el programa.

2.8 Mapa de compatibilidad de las actividades del proceso con CMMI

Metas (CMMI)	Prácticas de CMMI	Prácticas del Proceso: Desarrollo de Requisitos.	Cobertura	Comentarios
SG 1	Desarrollar los requisitos del cliente.			
SP 1.1	Obtener las necesidades.	Obtener las necesidades, expectativas y restricciones del cliente.	Totalmente	
SP 1.2	Desarrollar los requisitos de cliente.	Descripción del negocio. Elaborar y refinar los requisitos.	Totalmente	
SG 2	Desarrollar los requisitos de producto.			
SP 2.1	Establecer los requisitos de producto y de componentes de producto.	Elaborar y refinar los requisitos. Establecer los conceptos operativos y escenarios.	Totalmente.	
SP 2.2	Asignar los requisitos de componentes del	Elaborar y refinar los requisitos.	Totalmente.	

	producto.			
SP 2.3	Identificar los requisitos de la interfaz.	Identificar los requisitos de la interfaz.	Totalmente.	
SG 3	Analizar y validar los requisitos.			
SP 3.1	Establecer los conceptos operativos y los escenarios.	Establecer los conceptos operativos y los escenarios.	Totalmente.	
SP 3.2	Establecer una definición de la funcionalidad requerida.	Distribución de los requisitos según la funcionalidad requerida.	Totalmente.	
SP 3.3	Analizar los requisitos.	Analizar los requisitos.	Totalmente	
SP 3.4	Analizar los requisitos para alcanzar el equilibrio.	Analizar los conceptos operativos y escenarios.	Totalmente	
SP 3.5	Validar los requisitos.	Validar los requisitos.	Totalmente	

Tabla 2.4 Mapa de compatibilidad de las actividades del proceso con CMMI.

2.9 Mapa de compatibilidad de las actividades del proceso con ISO 90 003.

(ISO 90 003)	Prácticas de: ISO 90003	Prácticas del Proceso: Desarrollo de Requisitos.	Cobertura	Comentarios
--------------	-------------------------	--	-----------	-------------

1	Determinación de los requisitos relacionados con el producto.	Obtener las necesidades, expectativas y restricciones del cliente. Descripción del negocio. Elaborar y refinar los requisitos.	Totalmente.	
2	Revisión de los requisitos relacionados con el producto.	Analizar los requisitos. Analizar los conceptos operativos y escenarios.	Totalmente.	.
3	Comunicación con el cliente.	Obtener las necesidades, expectativas y restricciones del cliente. Validar los requisitos.	Totalmente.	

Tabla 2.5 Mapa de compatibilidad de las actividades del proceso con ISO 90 003.

2.10 Mapa de compatibilidad de las actividades del proceso con ISO 12207.

(ISO 12207)	Prácticas de: ISO 12207	Prácticas del Proceso: Desarrollo de Requisitos.	Cobertura	Comentarios
1	Análisis de los requisitos del sistema.	Obtener las necesidades, expectativas y restricciones del cliente. Descripción del negocio. Elaborar y refinar los requisitos.	Totalmente	
2	Análisis de los requisitos del software.	Obtener las necesidades, expectativas y restricciones del cliente. Descripción del negocio.	Totalmente	.

		Elaborar y refinar los requisitos. Identificar los requisitos de la interfaz.		
--	--	--	--	--

Tabla 2.6 Mapa de compatibilidad de las actividades del proceso con ISO 12 207.

2.11 Mapa de compatibilidad de las actividades del proceso con ISO 15504.

ISO (150504)	Prácticas de ISO 15 504	Prácticas del Proceso: Desarrollo de Requisitos.	Cobertura	Comentarios
1	OBTENER LOS REQUISITOS			
1.1	Obtener los requisitos y pedidos del cliente.	Obtener las necesidades, expectativas y restricciones del cliente.	Totalmente	
1.2	Comprender las expectativas del cliente.	Descripción del negocio.	Totalmente	
1.3	Estar de acuerdo con los requisitos.	Obtener las necesidades, expectativas y restricciones del cliente. Elaborar y refinar los requisitos.	Totalmente	
1.4	Establecer una línea base de los requisitos del cliente.	Elaborar y refinar los requisitos.	Parcialmente	El modelo presenta como primera base el documento: Especificación de requisitos del software.
1.5	Establecer mecanismos para consultas con el cliente.	Obtener las necesidades, expectativas y restricciones del cliente.	Totalmente	

2	REQUISITOS DEL SISTEMA.				
2.1	Establecer los requisitos del sistema.	los del	Elaborar y refinar los requisitos.	Totalmente	
2.2	Optimizar solución proyecto.	la del	No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
2.3	Analizar requisitos del sistema.	los del	Analizar los requisitos.	Totalmente	
2.4	Evaluar y actualizar los requisitos del sistema.	los del	Analizar los conceptos operativos y escenarios.	Totalmente	
2.5	Asegurar consistencia.		No se cubre		Esta práctica se cumple en el proceso de administración de requisitos.
2.6	Comunicar requisitos del sistema.	los del	Validar los requisitos.	Totalmente	
3	DISEÑO DE LA ARQUITECTURA DEL SISTEMA.				
3.1	Describir arquitectura del sistema.	la del	Identificar los requisitos de la interfaz.	Totalmente.	

3.2	Definir interfaces.		Elaborar y refinar los requisitos. Identificar los requisitos de la interfaz.		
3.3	Verificar la arquitectura del sistema.		No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
3.4	Evaluar alternativas para la arquitectura del sistema.		No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
3.5	Asegurar la consistencia.		No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
3.6	Comunicar el diseño de la arquitectura del sistema.		No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
4	ANÁLISIS DE LOS REQUISITOS DEL SOFTWARE.				
4.1	Especificar los requisitos del software.		Elaborar y refinar los requisitos. Analizar los requisitos. Analizar los conceptos operativos y escenarios.	Totalmente.	
4.2	Determinar el impacto ambiente operacional.		Analizar los conceptos operativos y escenarios.	Totalmente.	

4.3	Desarrollar un criterio para probar el software.	No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
4.4	Asegurar consistencia.	No se cubre		Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
4.5	Evaluar y actualizar los requisitos.	Elaborar y refinar los requisitos. Analizar los requisitos Analizar los conceptos operativos y escenarios.	Totalmente.	Esta actividad se le da cumplimiento en otro proceso del modelo de calidad.
4.6	Comunicar los requisitos del software.	Validar los requisitos.	Totalmente.	

Tabla 2.7 Mapa de compatibilidad de las actividades del proceso con ISO 15 504.

CONCLUSIONES DEL CAPÍTULO

En el presente capítulo se definió una propuesta del proceso de Desarrollo de Requisitos para la mejora de la calidad del software. Mostrando una descripción gráfica de todas las prácticas a realizar durante este proceso con sus roles definidos y los artefactos que se generan, además de las técnicas y herramientas que sirven de apoyo en la realización de las prácticas.

Como resultado, se concluye que el proceso propuesto es el indicado para suplir las necesidades de las pequeñas y medianas empresas, y transforma las debilidades de otros modelos en fortalezas. Además se garantiza una compatibilidad con las prácticas relacionadas con el desarrollo de Requisitos de los modelos CMMI y las normas ISO de manera que las empresas que adopten este proceso puedan obtener una certificación ISO o una evaluación en CMMI con muy poco esfuerzo.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Introducción

Con anterioridad, en el Capítulo 1, se abordan los principales temas relacionados con el desarrollo de requisitos. En el Capítulo 2 se realiza una propuesta del proceso de Desarrollo de Requisitos mediante una descripción gráfica y textual, se proponen técnicas y herramientas que sirven de apoyo en la realización de las actividades del proceso además de una base de conocimientos.

Se dedica entonces el presente capítulo a la validación de la propuesta utilizando el criterio de un grupo de expertos, realizándose la descripción de cómo fue ejecutado y los resultados obtenidos.

3.1 Validación de la Investigación

Como la propuesta de la presente investigación es la creación del proceso de Desarrollo de Requisitos que debe ser utilizado en el MCCDAI, proceso que no se desarrolla en el momento en que esta se obtiene, se hace imposible realizar la aplicación como método de valoración. Por lo dicho anteriormente, se aplicarán métodos subjetivos para la validación del mismo, pues tampoco es posible hacer uso de otros como los de tipo extrapolativo o causal, que relacionan variables que existieron en el pasado, debido a la carencia de datos de entrada.

3.2 Método Delphi

Para la validación del proceso de Desarrollo de Requisitos, se utiliza como método de criterio de expertos el Delphi. Se tiene en cuenta para su utilización la posibilidad que ofrece de obtener información de forma independiente y de evitar evaluaciones superficiales.

Delphi es un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo. Esto se realiza con la ayuda de cuestionarios sucesivos con la finalidad de poner de manifiesto las convergencias de opiniones y deducir eventuales consensos. Su capacidad de predicción se basa en el empleo sistemático de un juicio intuitivo emitido por un grupo de expertos.

3.3 Nivel de competencia de los especialistas

El experto es elegido por sus conocimientos demostrados sobre el tema consultado. Para lograr la independencia en sus criterios estos son aislados y sus opiniones serán recogidas de forma anónima. Así se obtendrán sus opiniones reales y no una falseada por un proceso de grupo.

Las características de los expertos influyen decisivamente en la confiabilidad de los resultados obtenidos. Por tal razón se seleccionaron un grupo de analistas y de especialistas con reconocida experiencia en el tema de calidad, con prestigio ante el colectivo, capacidad de análisis y pensamiento lógico, creatividad y por supuesto, disposición para participar en la validación. Junto con estas características se tuvieron en cuenta otras más personales como la seriedad ante el trabajo, la honestidad, sinceridad y responsabilidad. Con la elección de los expertos atendiendo a estas características se espera que garantice la calidad del proceso, a pesar de la juventud del colectivo de trabajo de la universidad y sus no tantos años de experiencia.

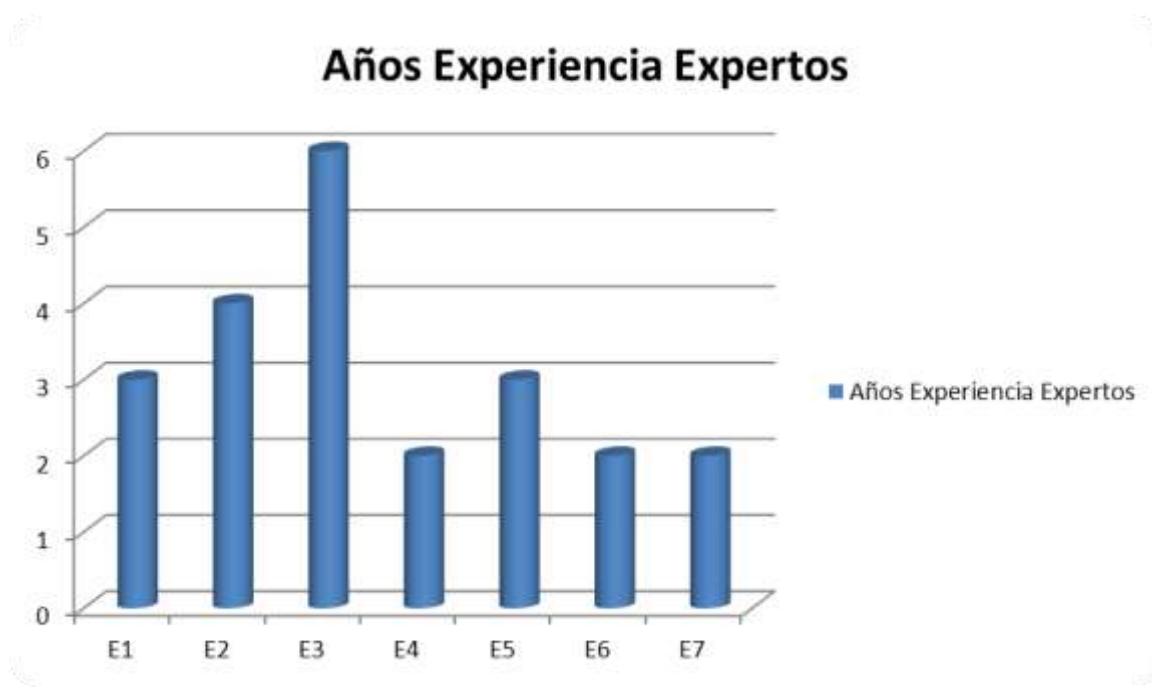


Figura 3.1 Representación de los Años de Experiencia de los Expertos.

Como se puede observar en la **Figura 3.1**, los años de experiencia de trabajo en el tema, presentado en la encuesta, de los expertos encuestados nunca es menor de 2 años

Ocupación Expertos

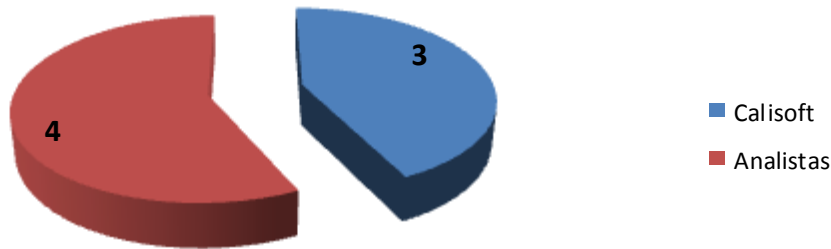


Figura 3.2 Ocupación de los Expertos.

Como se puede observar en la **Figura 3.2**, se trató de llevar un equilibrio entre los encuestados, tratando de que fuesen expertos tanto de calidad como analistas de proyectos productivos del centro CEGEL, ya que el mismo se encuentra en un proceso para alcanzar una evaluación internacional del nivel 2 de CMMI.

3.4 Explotación de resultados.

Para valorar y procesar los resultados de la encuesta se utilizó el software estadístico Statistical Product and Service Solutions (SPSS).

Los resultados de la evaluación de los indicadores por parte de los expertos que se muestran a continuación constituyen la entrada al cálculo del coeficiente W, anteriormente explicado. La concordancia se considera aceptable si $W > 0.5$.

Exp\Cri	C1	C2	C3	C4	C5	C6
E1	0	1	1	1	1	4
E2	1	1	1	1	1	4
E3	1	1	1	1	1	3
E4	0	1	0	1	1	3
E5	1	0	0	1	0	3

E6	1	1	1	1	1	4
E7	1	1	0	1	1	3

Tabla 3.1. Matriz de evaluación de los criterios por experto.

Los resultados de los cálculos realizados en el SPSS son los siguientes:

Variables	Rango Promedio
C1	2.79
C2	3.14
C3	2.43
C4	3.50
C5	3.14
C6	6

Tabla 3.2. Rangos de la evaluación de los indicadores.

N	7
W de Kendall	,737
Chi-Cuadrado	25.806
gl	5
Sig. Asintot	,000

Tabla 3.3. Estadísticos de contraste.

El coeficiente W ofrece el valor que posibilita decidir el nivel de concordancia entre los jueces. El valor W oscila entre 0 y 1. El valor 1 significa una concordancia de juicios total y el valor 0 un desacuerdo total; obviamente la tendencia a 1 es lo deseado, siendo en este caso el valor de W de 0.737, se considera aceptable el nivel de concordancia entre los expertos. Para dar validez a la decisión se acude además al valor de chi-cuadrado. En la Tabla 3 se tiene el número de casos válidos, el valor del estadístico W, su transformación en chi-cuadrado, sus grados de libertad y el nivel crítico (Sig. asintót.). Puesto que el valor del nivel crítico (0,000) es menor que 0,05; se puede rechazar la hipótesis

de concordancia nula, la hipótesis nula no es más que la importancia de las seis variables es la misma, cuestión que no demuestra un buen ordenamiento de las variables, por lo que se puede concluir que entre las puntuaciones de las seis variables estudiadas existe asociación significativa.

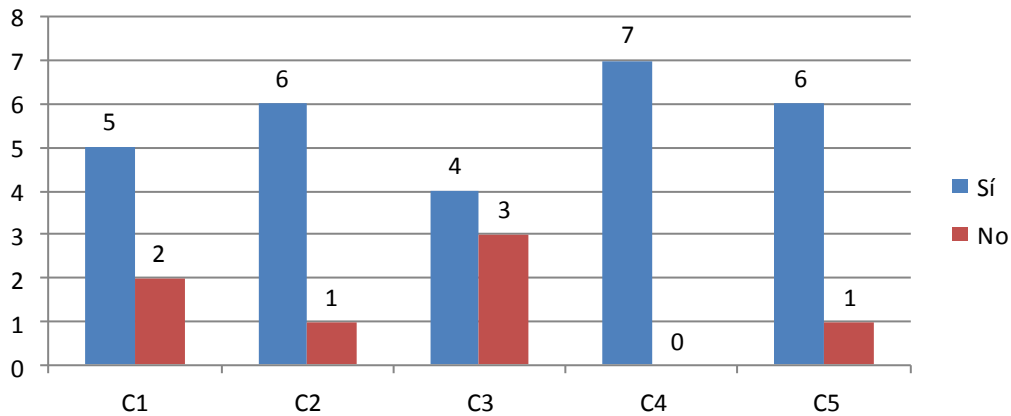


Figura 3.3 Respuesta de los expertos de acuerdo a los criterios establecidos.

En la **Figura 3.3** se muestra la cantidad de expertos que estuvieron de acuerdo y en desacuerdo con cada uno de los criterios, exceptuando el criterio 6 que se muestra en la **Figura 3.4**.

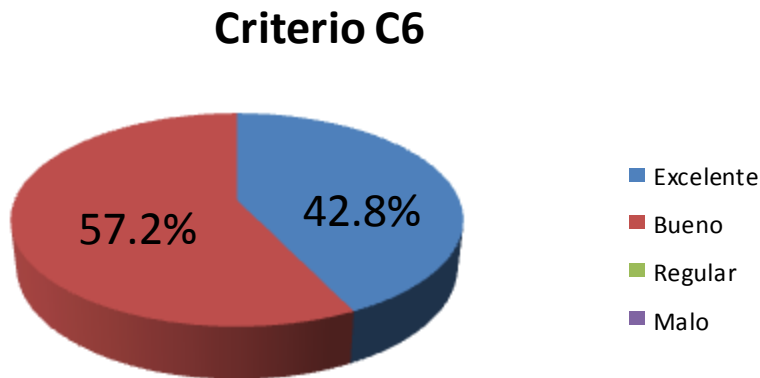


Figura 3.4 Respuesta de los expertos respecto al criterio C6.

Como se había anunciado anteriormente, en esta gráfica se muestran los criterios de los expertos con respecto al criterio C6, concluyendo que ningún experto señaló el proceso de Desarrollo de Requisitos propuesto para el MCCDAI como regular o malo, quedando todos los criterios divididos solo entre Excelente y Bueno.

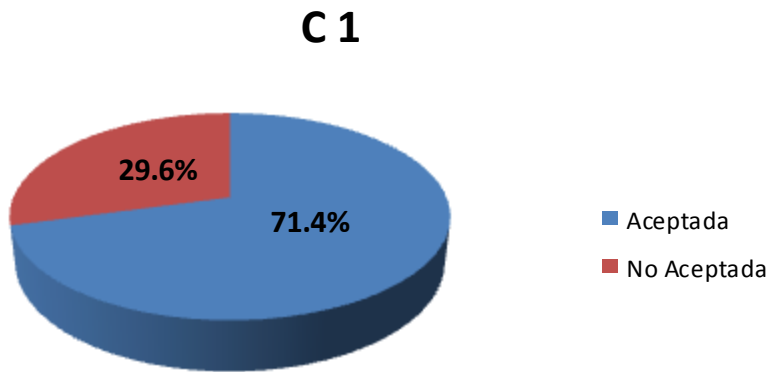


Figura 3.5 Respuesta de los expertos respecto al criterio C1.

El Criterio C1 (Prácticas integradas al proceso) tuvo una aceptación del 71.4% por parte de los expertos encuestados, por lo que representa una mayoría de aceptable para la aprobación del mismo.

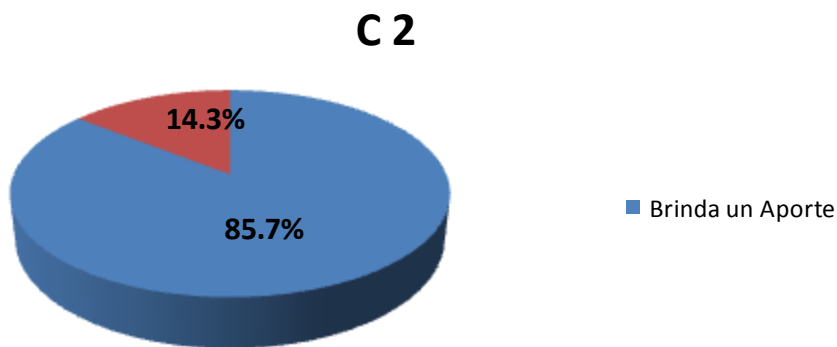


Figura 3.6 Respuesta de los expertos respecto al criterio C2.

En el Criterio C2 (Aporte brindado por el proceso para la creación del MCCDAI) tuvo una aprobación del 85.7% por parte de los expertos encuestados, por lo que también representa una mayoría de aceptable para la validación del mismo.

C 3

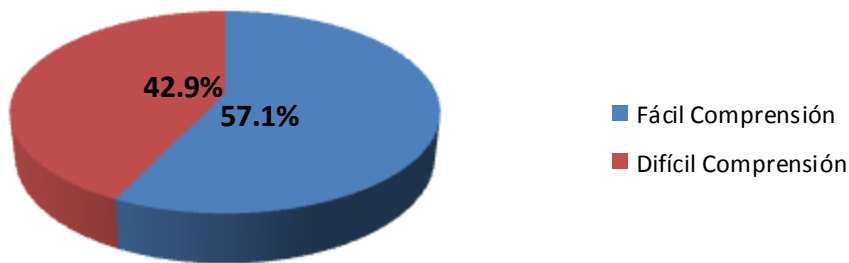


Figura 3.7 Respuesta de los expertos respecto al criterio C3.

En el Criterio C3 (Fácil comprensión del proceso) fueron aceptadas por el 57.1% de los especialistas que participaron en la encuesta, siendo por lo tanto, aceptada por la mayoría del mismo.

En el Criterio 4 se hacía referencia a los roles que realizan las prácticas durante el proceso, con un 100% de concordancia de los expertos, se está 100% seguro que los roles planteados en el Capítulo 2 son los correctos para realizar las prácticas.

C 5

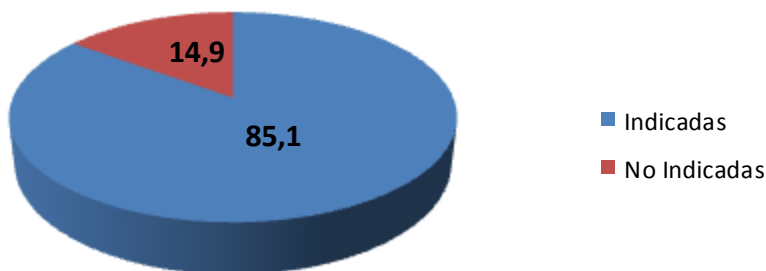


Figura 3.8 Respuesta de los expertos respecto al criterio C5.

En el Criterio 5 (Las técnicas y herramientas sugeridas para facilitar el trabajo durante las actividades) son las indicadas, con un 85.1% de concordancia entre los expertos, cumple con las necesidades para facilitar el desarrollo de las prácticas propuestas en el proceso Desarrollo de Requisitos.

Conclusiones del capítulo

El método Delphi, una técnica que cumple más de medio siglo de vida, sigue teniendo vigencia en los ámbitos académicos y profesionales. El recurso al conocimiento de los expertos sigue estando plenamente justificado y el contexto incierto en el que con cada vez mayor frecuencia deben tomarse importantes decisiones, acrecienta la necesidad de su uso. Con la aplicación de este método se logra recurrir a un grupo de expertos en el tema que avalan de forma positiva el resultado de la presente investigación. En este capítulo se consigue validar el proceso de Desarrollo de Requisitos para el MCCDAI, dando cumplimiento de esta forma a la hipótesis formulada en un principio.

CONCLUSIONES GENERALES.

Con la realización del presente trabajo de diploma se arribó a las siguientes conclusiones:

La definición del Marco teórico de la investigación permitió conocer los principales elementos teóricos vinculados con el desarrollo de los requisitos en los diferentes modelos de calidad del software existentes a nivel internacional.

La definición del proceso de Desarrollo de Requisitos, realizado a partir del estudio de los elementos teóricos estudiados, contribuirá a la creación del MCCDAI ya que asegura el cumplimiento de las buenas prácticas y mejora las deficiencias de los modelos estudiados en el Marco teórico.

El método de evaluación utilizado arrojó resultados positivos, demostrado con las calificaciones obtenidas por cada uno de los expertos en cada criterio. Con la utilización del método Delphi se demostró que el proceso de Desarrollo de Requisitos definido, ayudará a la creación del MCCDAI.

RECOMENDACIONES

Se recomienda:

- Se recomienda profundizar en el estudio del proceso, realizando la validación del mismo con un mayor número de expertos que avalen la propuesta de su realización, así como de sus artefactos, roles y herramientas.
- Capacitar al personal de aseguramiento de la calidad de los diferentes proyectos de la Universidad de las Ciencias Informáticas, en los diferentes aspectos presentes dentro del proceso de Desarrollo de Requisitos del MCCDAI.
- Se aplique el proceso de Desarrollo de Requisitos del MCCDAI, en los distintos proyectos productivos de la Universidad de las Ciencias Informáticas.

Bibliografía

UCI. (2011). *www.uci.cu*. Obtenido de Produccion, Investigacion: <http://www.uci.cu/>

Pressman, Roger S. 2005. *Un Enfoque Práctico*. Mexico : Mcgraw-hill Mexico, 2005.

Hofmann, Hubert F. y Lehner, Franz. 2001. *Requirements Engineering as a Success Factor in Software Projects*. 2001.

Hooks, Ivy F. y Farry, Kristin A. 2001. *Customer Centered Products: Creating Successful Products Through Smart Requirements Management*. 2001.

Vismar Santos Hernández. 2009. Gestipolis. *Estudio sobre la industria del Software a nivel mundial. Caracterización en América Latina y Cuba*. [En línea] 2009. <http://www.gestipolis.com/administracion-estrategia/estudio-sobre-la-industria-del-software-en-america-latina.htm>.

Ministerio de Ciencia, Tecnología y Medio Ambiente. 2011. Oficina Nacional de Normalización. [En línea] 2011. <http://www.nc.cubaindustria.cu/>.

Chrissis, Mary Beth, Konrad, Mike y Shrum, Sandy. 2009. *Guía para la Integración de Procesos y la Mejora de Productos*. 2009.

Caferino, Hernan. 2006. *Una Introducción a CMMI*. s.l. : White Paper, 2006.

Sanches , Fernando . 2007. *www.revista-ays.com. CMMI vs. ITIL: dos enfoque complementarios*. [Online] 2007. www.revista-ays.com/DocsNum15/Tribuna/Fernando.pdf.

Chrissis, Mary Beth . 2010. *"CMMI : Guidelines for Process Integration and Product Improvement de SEI"*. 2010.

SOFTEX, sociedad. 2009. *MPS.BR - Mejora de Proceso del Software Brasileño. Guía de Implementación – Parte 4: Fundamentos para Implementación del Nivel D del MR-MPS*. BRASIL : s.n., 2009.

Oktaba, Hanna. 2005. *Modelo de procesos para la Industria del Software MoProSoft*. 2005.

Zúñiga , Gloria . 2009. *Tablas comparativas del Proceso Unificado y MOPROSOFT*. [En línea] 2009. <http://apascualreyes.blogspot.com/>.

ISO. 2006. *INGENIERIA DE SOFTWARE —DIRECTIVAS PARA LA APLICACIÓN DE LA NC ISO 9001:2001 AL SOFTWARE DE COMPUTACIÓN (ISO/IEC 90003:2004, IDT)*. 2006.

JTC1, "Standardization in the field of information technology systems and equipments." . 1995. *INTERNATIONAL STANDARD ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES*. 1995.

IN3302, ISO/IEC JTC1/SC7. 2005. *FDIS 15504-5 - Information Technology — Process Assessment — Part 5: An exemplar Process Assessment Model*. 2005.

López, Jorge. 2009. Curso de Auditoría y Garantía de Calidad. *www.monografias.com*. [En línea] noviembre de 2009. <http://www.monografias.com/cgi-bin/search.cgi?query=curso+de+auditoria+y+garantia+de+calidad&categoria=Software>.

Jeffrey Baumgartner. 2007. Brainstorming: una guía paso a paso. *www.mujeresdeempresa.com*. [En línea] 12 de Julio de 2007. <http://www.mujeresdeempresa.com/negocios/070701-brainstorming-paso-a-paso.asp>.

Técnicas Recolección de Datos. **Garrido, Eunice Eugel. 2010.**

Stanton , William , Etzel , Michael and Walker , Bruce Mc Graw. 2009. *Fundamentos de Marketing*. 2009.

Wesley, Addison , et al. 1999. *El Lenguaje Unificado de Modelado*. 1999.

OMG (Object Management Group). 2000. Guía UML. [Online] 2000. <http://www.osmosislatina.com/lenguajes/uml/index.htm>.

Booch, Grady, Rumbaugh, Jim and Jacobson, Ivar . 2008. *UML El lenguaje unificado de modelado*. 2008.

Hitschfeld, Nancy. 2010. Universidad de Chile. *Tutorial UML. Modelo de Clases*. [Online] 2010. <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>.

UDIS, Unidad Docente de Ingeniería del Software. 2009. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software. *Validación de los requisitos*. [Online] 2009. http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf.

Sommerville, Ian. 2005. *Ingeniería del Software*. 2005.

Cáceres, Paloma. 2010. *Análisis e Ingeniería de Requisitos. Documentación, Validación y Gestión de Requisitos*. 2010.

Hurtado, Toskano and Bruno, Gerard. 2010. tesis: El proceso de análisis jerarquico (AHP) como herramienta para la toma de decisiones en la selección de proveedores. 2010.

Telecomunicaciones, Oficina de Informática. 2009. Manual de Procedimientos_Gestión de Solicitudes y Requerimientos. 2009.

González, Rubén y Pérez, Sergio. 2010. <http://es.scribd.com>. *Introducción al Rational Rose*. [En línea] 2010. <http://es.scribd.com/doc/55623280/LESE-2-Introduccion-a-Rational-Rose>.

Headquarters, Company. 2011. <http://www.visual-paradigm.com/>. *Visual Paradigm*. [En línea] 2011. <http://www.visual-paradigm.com/>.

Systems, Company Sparx. 2011. Sparx Systems. *Enterprise Architect*. [En línea] 2011. <http://www.sparxsystems.com/products/ea/index.html>.

UML, Star. 2011. Star UML. *Star UML*. [En línea] 2011. <http://staruml.sourceforge.net/en/>.

Corporate and Sales Headquarters, North America. 2011. <http://www.magicdraw.com/>. *Magicdraw*. [En línea] 2011. <http://www.magicdraw.com/>.

Calisoft. 2010. Evaluación final del Programa de Mejora. *Calisoft centro de Calidad*. [En línea] 2010. <http://calisoft.uci.cu/index.php/noticias/6-noticias/94-spm>.