

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Proceso Base de Ingeniería de Requisitos para las
pequeñas y medianas empresas de desarrollo de software

Tesis en opción al título de
Máster en Calidad de Software

Autora:

Ing. Kariné Ramos Blanco

Tutora:

Msc. Karina Pérez Teruel

La Habana, Cuba

Julio 2013

Agradecimientos

A mis padres, que son mi guía y faro eterno.

A mi hermano, que siempre será mi niño adorado.

A mi gran e inseparable amor, mi jb.

A mi tía Ondina que es mi otra madre.

Al resto de mi familia que me ha apoyado de un modo u otro.

A mi tutora y amiga que ha sido autora de este trabajo a la par.

A mis amigos, que me ayudaron tanto en el desarrollo de este trabajo: a Micha por sus sesiones críticas tan necesarias, a Abel Alba que me dedicó su tiempo aún teniendo muy poco, a Ramses que me ayudó en la revisión y validación, a Denis, Irina, Ani, Dagmay. A Maikel por su ayuda con los excels, jaja. A Dari por su apoyo en todos los tiempos, a Debo por las charlas que ayudaron a tomar decisiones (bidireccionales por supuesto), a Rebe y Yunaldis por su constancia y dedicación en el trabajo con los pilotos.

A los profes del PEFCI que estuvieron ahí ante las constantes consultas.

A mis tesisas de pregrado por su contribución.

A todos, los que aunque no menciono aquí siempre estuvieron al tanto de mis resultados en los cursos, tesinas, pre defensa y me dieron ánimos para culminar.

Declaración de Autoría

Declaro que soy la única autora de este trabajo y autorizo al Centro Calisoft y a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Ing. Kariné Ramos Blanco

Msc. Karina Pérez Teruel

Resumen:

Los requisitos son la voz del cliente y desde el surgimiento del proyecto hay que prestarles especial atención pues pueden definir el camino al éxito. En el presente trabajo se propone un Proceso Base de Ingeniería de Requisitos ligero, que combina las actividades para el desarrollo y administración de requisitos, enfocado en aquellas empresas con restricciones financieras y de recursos como las pequeñas y medianas empresas (PYME). Este proceso contiene los componentes establecidos por el Modelo Cubano de Desarrollo de Aplicaciones Informáticas (MCDAI), modelo en desarrollo que no tiene un proceso de ingeniería de requisitos y que incluirá el propuesto en esta investigación. En la solución planteada se incluyen los roles involucrados, los productos de trabajo que se obtienen, una propuesta de aquellos que formarán parte de la base de conocimientos de la organización y una lista de sugerencias de técnicas y herramientas que apoyen el manejo de los requisitos. Se define además una descripción detallada de un sistema de indicadores que permite medir elementos técnicos y de proceso relacionados con los requisitos, posibilitando a partir de datos históricos mejorar los resultados y predecir comportamientos futuros. Este proceso fue sometido a criterio de expertos y aplicado en dos proyectos en los que se midió además su nivel de satisfacción.

Palabras claves: Ingeniería de Requisitos, Administración de Requisitos, Desarrollo de Requisitos, Indicadores, Procesos, Modelo de Calidad.

Abstract:

The requirements are the client voice and since the beginning of the project should be given a special attention because they may define the success. In this paper is proposed a light Process Base of Requirements Engineering that combines the management and developments requirements activities, focused on those enterprises that have financial and resources constraints such as the small and medium enterprises (SMEs). This process contains the components established by the Cuban Model for Application Development (MCDAI) model not finished yet, that does not have a process of requirements engineering and will include the proposed in this research. This solution includes the roles involved, work products that are obtained and those who are going to be part of the knowledge base of the organization and a list of suggestions of techniques and tools, to support requirements work. Is also defined a detail description of the indicators system that allow to measure technical and process elements related to the requirements, so is possible using historical data to improve results and predict

future behaviors. This process was subjected to expert criteria and applied in two projects where were also measured their level of satisfaction.

Keywords: Requirements Engineering, Requirements Management, Requirements Development, Indicators, Process, Quality Models.

ÍNDICE

Introducción	8
Capítulo 1. Introducción	16
1. Introducción.....	16
1.1 Requisitos	16
1.2 Ingeniería de Requisitos.....	18
1.3 Base de Conocimiento	19
1.4 Características de las Pequeñas y Medianas empresas (PYME)	19
1.5 Productividad	21
1.6 Modelos y estándares de calidad empleados en la investigación.....	22
Conclusiones parciales del capítulo.....	29
Capítulo 2: Propuesta del Proceso Base de Ingeniería de Requisitos	30
2. Introducción.....	30
2.1 Proceso Base de Ingeniería de Requisitos (PB de IR).....	30
2.2 Roles del PB de IR	37
2.3 Productos de trabajo	38
2.4 Productos de trabajo a incorporar en la Base de Conocimientos.....	38
2.5 Mapa de compatibilidad.....	39
Conclusiones parciales del capítulo.....	39
Capítulo 3. Propuesta de técnicas, indicadores y herramientas para el PB de IR	41
3. Introducción.....	41
3.1 Técnicas para administrar y desarrollar los requisitos.....	41
3.2 Sistema de Indicadores	49
3.3 Herramientas	52
Conclusiones parciales del capítulo.....	55
Capítulo 4: Validación de la solución.....	56
4. Introducción.....	56
4.1 Evaluación del Proceso Base de Ingeniería de Requisitos y el sistema de indicadores mediante valoración de expertos.....	56
4.2 Cuasi experimento para validar la solución.....	60
4.3 Evaluación de la satisfacción de los usuarios del Proceso y Sistema de Indicadores	66
4.4 Triangulación de resultados.....	69
Conclusiones parciales.....	70

Conclusiones y Recomendaciones	71
Conclusiones.....	71
Recomendaciones.....	71
Bibliografía.....	72
Referencias Bibliográficas.....	73

Introducción

La industria de software (IS) ha tenido un constante de éxitos y fracasos. La compañía Standish Group ha sido protagonista en la publicación de estadísticas a este respecto desde 1994. Sus primeros estudios incluyeron una muestra de grandes, medianas y pequeñas empresas de los principales segmentos de la industria (banca, seguridad, servicios al por menor y al mayor, manufactura, salud, seguros y organizaciones locales, estatales y federales). El tamaño total fue de 365 encuestados y representó 8.380 aplicaciones. (STANDISH GROUP, 1995)

El Standish Group ha llevado a cabo numerosos grupos focales, encuestas a profundidad y entrevistas ejecutivas sobre unos 50 mil proyectos de tecnología e información (TI) terminados, de numerosos países. Esta compañía tiene entre sus principales objetivos documentar el alcance de los fallos de los proyectos de desarrollo de software, los principales factores de fracaso y las maneras de reducirlo. (HARTMANN, 2006)

Para esta compañía un proyecto es exitoso si se completa en tiempo y dentro del presupuesto y contiene todos los requisitos especificados por el cliente. Se ubica en la categoría de “cancelado” cuando se cancela antes de su terminación o no se implementa. Y se clasifica como “incompleto” cuando el proyecto finaliza pero sobrepasando el tiempo y presupuesto estimado o el producto final no cuenta con todos los requisitos especificados inicialmente.

A través de sus estadísticas publicadas puede constatar una clara tendencia de la IS a organizarse, logrando a través de los años aumentar el número de proyectos exitosos y disminuir los cancelados, ver figura 1.

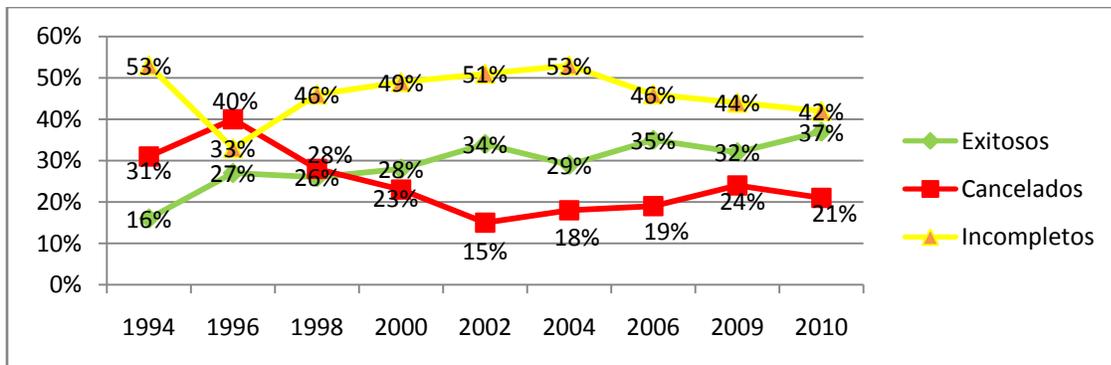


Figura 1. Gráfico elaborado por la autora a partir de datos publicados originalmente por el Standish Group y citados en (DOMINGUEZ, 2009; WALL et al., 2012)

La IS en el mundo está madurando, logrando en mayor medida, que una vez comenzado un proyecto su tendencia sea a culminarse. La industria cubana de software (ICS) tiene que unirse a esta tendencia para insertarse y sobrevivir en el mercado.

Es importante remarcar que aunque la IS se esté moviendo hacia la organización aún existe un porcentaje importante de proyectos que se cancelan. Sobre las **causas** de las cancelaciones, en 1995, el reporte del Standish Group (CHAOS'94, 1995) reveló que tres de las diez principales estaban relacionadas directamente con los requisitos: **requisitos incompletos, la falta de participación de los usuarios y requisitos inestables.**

El hecho de que los usuarios no participen en el proceso lo suficiente, ocasiona malos entendidos que conllevan a la implementación de un producto erróneo, que no contiene lo que el usuario desea (requisitos incompletos), a partir de esto se exige la implementación de nuevos elementos y modificaciones a aquellos acuerdos que fueron establecidos sin la profundidad de análisis requerida (requisitos inestables). Provocando un esfuerzo adicional, teniendo el proyecto que gastar en re trabajo, afectando la calidad y productividad.

En el 2001 la participación del usuario continuaba siendo una preocupación fundamental así como la claridad de la visión y objetivos (CHAOS'01, 2002). Todavía en el 2010 la primera causa del fallo de los proyectos era la falta de participación de los usuarios (STANDISH GROUP, 2010) y en el último reporte en 2012 estos elementos relacionados con los requisitos: participación de los usuarios y objetivos de negocio claros, siguieron mostrándose entre los más importantes. (WALL *et al.*, 2012).

Entre 12 elementos que menciona Charette en (CHARETTE, 2005) relacionados con el fallo de los proyectos se manifiestan: la pobre definición de requisitos del sistema y poca comunicación entre clientes, desarrolladores y usuarios.

Constatándose la atención que deben prestar los proyectos al manejo y desarrollo de los requisitos.

De aquellos proyectos que se culminan, la mayoría presenta problemas con el desarrollo de la totalidad de los requisitos. Acorde a (STANDISH GROUP, 2009) en 1998 el 66% de los requisitos especificados estaban incluidos en el producto final, en el 2000 el 70%, en 2002 el 67%, en 2004 el 64%, en 2006 el 68% y en el 2009 un 67%. Aunque sobrepasan el 50%, no son buenos resultados, el cliente necesita el 100% de los requisitos especificados en el producto que recibirá, quedando insatisfecho sino es de esta forma.

Exceder el tiempo y presupuesto estimado afecta la productividad de los proyectos debido a que se alargan los plazos de entrega y resulta mucho más costosa la producción. También es cierto que cumplir las fechas pactadas y el presupuesto estimado no es suficiente para ser productivos si no se incluye en el producto final todos los requisitos especificados, es decir si se entrega un producto incompleto al cliente.

Esto ocurre debido a que la productividad se entiende como la relación entre la producción obtenida por un sistema de fabricación de bienes o servicios y los recursos utilizados para obtenerla según (GONZALO, 2013). Este autor menciona además que la productividad indica el mejor o peor uso que se hace de los factores de producción de una economía concreta, lo que teóricamente refleja su capacidad de competir con eficacia en el mercado.

Visconti plantea que los caminos para mejorar la productividad están asociados a la existencia de mejores herramientas, mejores procesos de desarrollo y mantención, y mejor planificación.(VISCONTI, 2003)

Acorde a Bastarrica (BASTARRICA, 2011): “han tenido más impacto en la productividad del desarrollo de software en las últimas décadas los aspectos metodológicos que los tecnológicos. Si bien hoy existen en el mercado generadores de código o herramientas de apoyo a los procesos que permiten mejorar la productividad, estos abordan dificultades accidentales del desarrollo de software, en términos de Brooks, y por lo tanto las ganancias que proporcionan son marginales. En cambio, estandarizar los procesos de desarrollo o definir buenas prácticas y seguirlas sistemáticamente parece haber sido mucho más efectivo”.

Las empresas de desarrollo de software han intentado estandarizar e institucionalizar buenas prácticas, a través, principalmente de la implementación de modelos o normas de calidad. Tal es el caso de CMMI (del inglés Capability Maturity Model Integration) y las normas ISO (del inglés International Organization for Standardization), nombrados entre los más populares y en el mercado latinoamericano últimamente han tomado auge algunas iniciativas locales como son Moprosoft y MPS.br.

Respecto a CMMI y las ISO mencionadas, Oktaba opina (OKTABA y PIATTINI, 2008): “(...) estas normas y modelos están concebidos para las organizaciones grandes como el Departamento de Defensa de los Estados Unidos, la NASA, fábricas multinacionales de software, y así sucesivamente. De hecho, hay una tendencia generalizada a enfatizar que el éxito de los programas de mejora de procesos (SPI) sólo es posible para las grandes empresas que cuentan con recursos suficientes para hacer frente a este tipo de prácticas. Esta percepción se basa en el hecho de que los SPI no son viables para pequeñas y medianas empresas (PYME), debido a su estructura organizativa y los altos costos que ello implica”.

Para ubicar una empresa en la categoría de pequeña y mediana empresa (PYME), acorde a la Comisión Europea (EUROPEAN COMMISSION, 2005) deben tenerse en cuenta los límites establecidos en cuanto a: personal, volumen de negocios anuales y balance anual. Estos sitúan a una empresa en la categoría de PYME cuando posee entre 51 y 250 trabajadores, el volumen

de ventas anuales se encuentra entre los 11 y 50 millones de euros y el balance anual entre los 11 y 43 millones de euros. Ver Anexo 1.

Las PYME con frecuencia tienen dificultades para obtener capital o créditos, en particular en la temprana fase de puesta en marcha. Sus recursos limitados pueden también reducir el acceso a las nuevas tecnologías e innovación.(EUROPEAN COMMISSION, 2005).

“Casi todos los expertos coinciden en que las características específicas de las PYME influyen en que los programas de mejora de procesos deben ser aplicados de una manera particular para ellas y visiblemente diferente de como se hace en las grandes organizaciones. Esto no es tan simple como crear una versión minimizada de las que se aplican en las grandes empresas. De hecho, las evaluaciones a las normas internacionales son costosas y requieren mucho tiempo, difíciles de realizar en las pequeñas empresas, su estructura de modelo de proceso es demasiado complejo, y el retorno de la inversión realizada tiene que ser visto desde una perspectiva a largo plazo”.(OKTABA y PIATTINI, 2008)

En el 2003 acorde a (FEBLES, 2003) la Industria Cubana del Software (ICS) estaba formada por alrededor de 250 entidades que bien exportaban o producían software o brindaban algún servicio informático. De estas entidades, solo 46 tenían carácter de empresa de software respecto a las cuales Febles concluyó que: “La industria de software cubana está compuesta fundamentalmente por pequeñas y medianas empresas.” Ver Anexo 2.

Del 18 al 22 de marzo se realizó en La Habana la XV Convención y Feria Internacional de Informática 2013 con una gran representación de las empresas que desarrollan software en Cuba participando un total de 21 (Ver Anexo 3). A través de encuestas aplicadas por la autora a las empresas participantes se obtuvo que el 47% estaba compuesta por una cantidad entre 51 y 250 trabajadores y teniendo en cuenta que el volumen de sus ventas anuales no sobrepasa los 11 millones (euros) se confirma que la ICS está formada en su mayoría por PYME. Ver Anexo 4.

En Cuba, 3 de los centros de desarrollo de la Universidad de las Ciencias Informáticas (UCI) ostentan la evaluación del nivel 2 de CMMI.(CMMI INSTITUTE, 2011)

Obtenida a través de la implementación de un SPI que comenzó en octubre de 2008 y realizó la evaluación en julio de 2011 (2 años y 9 meses después). Durante su implantación se consumió, para la definición de los procesos, 17356 horas, lo que equivale a 723 días de trabajo, con un esfuerzo de 810 horas/hombre. De estas 810 horas/hombres, 117 fueron dedicadas al proceso de Administración de Requisitos (REQM), aproximadamente el 15%. Agregando a estas cifras un total de 400 horas dedicadas a la capacitación de los proyectos sobre los procesos.

(RAMOS *et al.*, 2011). De las cuales 64 fueron dedicadas al proceso de REQM (un 16%). Este proceso involucra 9 del total de 30 roles definidos en el SPI por lo que en esas 64 horas de capacitación participaron el 30% de los miembros de los proyectos.

Dentro de estos datos no se considera el esfuerzo dedicado a capacitar al personal que definió los procesos, el que dedicaron los directivos en la aprobación de los elementos del SPI, el trabajo realizado inicialmente en los proyectos pilotos y posteriormente en el resto de los proyectos de cada centro, el dedicado a las revisiones de los proyectos, la preparación del SCAMPI (del inglés Standard CMMI Appraisal Method for Process Improvement) interno y externo. A estos elementos debe sumarse el costo monetario de una consultoría especializada y el pago de la evaluación final que superan los 70 000 usd.

Puede constatarse el esfuerzo y costo en que debe incurrirse al enfrentar un SPI bajo un modelo internacional como CMMI y el peso que tiene la ingeniería de requisitos dentro de estos. Aquí cabe destacar que no se incluyen prácticas de desarrollo de requisitos (lo que aumentaría las actividades y productos de trabajo a realizar) debido a que solo se implementó hasta el nivel 2 del modelo (nivel administrado donde no se encuentran elementos referentes al desarrollo).

Estos atributos conllevan a que también a nivel organizacional las empresas tengan que dedicar gran parte de sus recursos al SPI, lo que disminuye el destinado a la producción. Estas acciones deben valorarse profundamente por las PYME cubanas de desarrollo de software que enfrentan los mismos problemas que el resto de la industria. Necesitan culminar sus proyectos dentro del tiempo y presupuesto pactado y entregar un producto que incluya todos los requisitos además de organizar su proceso de desarrollo siguiendo buenas prácticas y prestando especial atención a las causas por las que fracasan los proyectos.

Es importante tener en cuenta que existe un problema cultural importante cuando se quiere importar y adoptar, sin más, modelos definidos en otros países, como señala Zahran (1998) “si el proceso no casa con la cultura de la organización será rechazado por el cuerpo organizacional como sucede en los trasplantes de órganos”.(TARDÍO *et al.*, 2011)

El Modelo Cubano para Desarrollo de Aplicaciones Informáticas (MCDAI), en construcción actualmente, tiene entre sus propósitos enfocarse a las características de las PYME de desarrollo de software y en cómo conducirlos hacia el éxito. Está dividido en 4 categorías: Gestión de Procesos de la Organización, Ingeniería, Soporte y Gestión de Proyecto. En la categoría de Ingeniería aún no cuenta con un proceso que encauce las buenas prácticas de ingeniería de requisitos y contribuya al éxito esperado.

A partir del análisis anterior puede resumirse que:

- Las causas de la cancelación de los proyectos están relacionadas en su mayoría con problemas con los requisitos.
- Un alto por ciento de los proyectos que culminan, lo hacen sobrepasando el tiempo, presupuesto o con requisitos incompletos trayendo consigo insatisfacciones.
- La productividad se ve afectada por el aumento del esfuerzo debido al re trabajo y al tiempo y presupuestos excedidos o a la no implementación de la totalidad de los requisitos.
- Las PYME de desarrollo de software aún siguiendo buenas prácticas para el manejo de los requisitos, contenidos en modelos y normas de calidad internacionales, usualmente ven afectada su productividad.
- Las PYME tienen restricciones financieras y de recursos que se contraponen a los altos costos necesarios para implementar las prácticas de los modelos y normas de calidad internacionales.

La situación planteada lleva a definir el siguiente **problema**:

¿Cómo adecuar buenas prácticas de ingeniería de requisitos a las características de las PYME de desarrollo de software para lograr un aumento de la productividad tanto a nivel de organización como de proyecto?

Se define como **objetivo general**: Desarrollar un proceso de ingeniería de requisitos que incluyendo las prácticas normadas internacionalmente y adecuado a las PYME logre el aumento de la productividad.

Objeto de estudio: La Ingeniería de Requisitos.

Campo de acción: Los procesos de ingeniería de requisitos.

Basada en el problema se formula la siguiente **hipótesis**:

Un proceso de ingeniería de requisitos adecuado a las PYME que incluya normativas internacionales y que implemente buenas prácticas de administración y desarrollo de requisitos contribuirá al aumento de la productividad en los proyectos y a nivel organizacional.

Como soporte al cumplimiento del objetivo general se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Evaluar la experiencia nacional e internacional de la aplicación de modelos y estándares de calidad.
- Diseñar un proceso para la ingeniería de requisitos.

- Realizar propuesta de artefactos para la base de conocimiento de la organización.
- Realizar propuesta de un sistema de indicadores que permita evaluar elementos técnicos y la ejecución del proceso diseñado.
- Validar el proceso y el sistema de indicadores mediante su aplicación en proyectos y mediante técnicas de validación teóricas.

Durante la realización de esta tesis se utilizaron diversos **métodos científicos**: teóricos y empíricos.

Métodos teóricos

Los métodos teóricos que se emplean en esta investigación fueron: el histórico-lógico, el hipotético-deductivo, al analítico-sintético y el sistémico.

El **histórico-lógico** para la revisión de la bibliografía existente sobre los modelos de calidad de software, las mejores prácticas para desarrollar y administrar los requisitos y las técnicas y herramientas recomendadas para apoyar estas prácticas. El **analítico-sintético** en la descomposición del problema de la investigación en elementos separados para la profundización en su estudio y luego sintetizarlos en la solución de la propuesta. El **hipotético-deductivo**, donde partiendo de la suposición del desarrollo de un proceso de ingeniería de requisitos se puede aumentar la productividad en los proyectos de desarrollo de software. Se aplicó también en la investigación el método **teórico sistémico** para visualizar la investigación como todo un sistema y trabajarla acorde a esa visión.

Métodos empíricos

Se utilizó el método **coloquial** para la presentación y discusión de resultados en sesiones científicas; el método de la **entrevista** para obtener informaciones en pos de argumentar la situación problemática y la validación de los resultados. Se aplicó el método de **expertos** para apoyar el valor científico de la investigación y el método **cuasi experimental** para la aplicación del proceso y los indicadores. Se realizaron **encuestas** para conocer elementos esenciales de las empresas cubanas, para la ejecución de la técnica de ladov y para obtener información de los expertos de la investigación.

Técnicas cualitativas:

Técnica de ladov para la validación de la satisfacción del usuario respecto al uso del proceso e indicadores.

Aporte práctico:

- Un proceso base de ingeniería de requisitos que unifique las prácticas de desarrollo y administración de los requisitos con elementos normados internacionalmente y adecuado a las PYME.

- Conjunto de componentes del proceso que viabilizan su interpretación e implantación: descripción gráfica y textual, roles y responsabilidades, productos de trabajo, propuesta de técnicas y herramientas.
- Un sistema de indicadores para evaluar el proceso base de ingeniería de requisitos así como elementos técnicos de los requisitos.

La tesis queda estructurada en 4 capítulos presentados brevemente a continuación:

- Capítulo 1: En este capítulo se define el marco teórico de la investigación. Se realiza un estudio para constatar los niveles de madurez en el desarrollo de la industria de software, centrándose en las actividades relacionadas con la administración y desarrollo de requisitos. Se realiza un análisis del estado del arte de los modelos de calidad existentes, sus ventajas y desventajas.
- Capítulo 2: Se presenta la propuesta del proceso base de ingeniería de requisitos, los roles, productos de trabajo y cuáles de estos formarán parte de la base de conocimiento.
- Capítulo 3: Se presenta una propuesta técnicas y herramientas para apoyar el proceso base de ingeniería de requisitos y se describe el sistema de indicadores elaborado.
- Capítulo 4: Se presenta la validación de los resultados de la investigación utilizando los métodos cuasi experimental, escalamiento de Likert fundamentado en la opinión de expertos y la técnica de ladov para medir la satisfacción de los usuarios del proceso e indicadores. Se utiliza el método de triangulación para verificar la convergencia de los resultados.

Capítulo 1. Introducción

1. Introducción

En este capítulo se describen los principales modelos y estándares de calidad haciendo énfasis en los elementos referentes a los requisitos de cada uno de ellos. Se presenta el estado del arte de la ingeniería de requisitos. Se realiza un estudio sobre la necesidad de la aplicación de un proceso de ingeniería de requisitos. También se especifican las principales terminologías utilizadas en el documento.

1.1 Requisitos

El término requisitos de software puede conceptualizarse acorde a lo planteado en la traducción certificada de la ISO 9000: 2000: se afirma que los clientes necesitan productos con características que satisfagan sus necesidades y sus expectativas y se expresan en la especificación del producto y son generalmente denominadas como requisitos del cliente. Los requisitos son la necesidad o expectativa establecida, generalmente implícita u obligatoria. (ISO 9000:2000, 2000)

Sommerville por su parte plantea que los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información.(SOMMERVILLE, 2005)

Para Pressman, los requisitos del software son la base de las medidas de la calidad y la falta de concordancia con los requisitos es una falta de calidad. (PRESSMAN, ROGER S., 2002)

La IEEE (del inglés Institute of Electrical and Electronics Engineers) por su parte, publicó que un requisito es: "Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal". ("IEEE Standar Glossary Software Engineering Terminology", 2006).

En el SWEBOK (del inglés Software Engineering Body of Knowledge) un requisito es descrito como una propiedad que el software debe tener para resolver algún problema en el mundo real.(IEEE, 2004)

El SEI (del inglés Software Engineering Institute) plantea como parte del glosario del modelo CMMI que un requisitos es: (1) Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo. (2) Una condición o capacidad que debe cumplir o poseer un producto o componente de producto para satisfacer un contrato, un estándar, una

especificación u otros documentos impuestos formalmente. (3) Una representación documentada de una condición o capacidad como en (1) o en (2).

En la presente investigación se toma la definición de requisitos definida por el SEI debido a que incluye elementos esenciales planteados por los estándares y modelos revisados y amplía el concepto, por lo que se considera la más completa y se encuentra acorde a los objetivos que persigue la autora para la realización de su trabajo.

Clasificación de requisitos

Los requisitos han sido definidos, entre otros casos, como funcionales y no funcionales, del cliente y del producto. A continuación se presentan definiciones realizadas sobre estas clasificaciones por distintos autores.

Un requisito funcional describe qué debe hacer el sistema.(SOMMERVILLE, 2004; YOUNG, 2004). Describe las funciones que el software ejecuta, por ejemplo formatear un texto o modular una señal. (IEEE, 2004).

Los requisitos no funcionales son restricciones en el servicio o función ofrecida por el sistema.(IEEE, 2004; SOMMERVILLE, 2004) Incluyen restricciones de tiempo, del proceso de desarrollo y estándares.(SOMMERVILLE, 2004) Especifican propiedades del sistema, como confiabilidad y seguridad.(YOUNG, 2004) Pueden ser requisitos no funcionales: el tiempo de respuesta máximo para determinadas consultas, que el sistema debe estar disponible para el cliente veinticuatro horas al día.

Los requisitos del cliente pueden ser definidos como el resultado de obtener, consolidar y resolver los conflictos entre las necesidades, las expectativas, las limitaciones y las interfaces de las partes interesadas relevantes del producto de una forma que sea aceptable para el cliente. (SOFTWARE ENGINEERING INSTITUTE, 2010a)

Sommerville ubica el concepto de requisito del producto como una clasificación de los requisitos no funcionales planteando: los requisitos no funcionales pueden venir desde características requeridas del software (requisitos del producto). Estos requisitos especifican el comportamiento del producto. Por ejemplo requisitos de rendimiento sobre cuán rápido debe ejecutarse el sistema y cuánta memoria necesita. (SOMMERVILLE, 2004)

Por su parte en (SWEBOK, 2004; YOUNG, 2004) los definen como aquellos que son producidos por un sistema.

Para el SEI, los requisitos del producto son definidos como un refinamiento de los requisitos del cliente en el lenguaje de los desarrolladores, transformando los requisitos implícitos en

requisitos derivados explícitos. El desarrollador usa los requisitos del producto para guiar el diseño y la construcción del producto. (SOFTWARE ENGINEERING INSTITUTE, 2010a)

Los requisitos técnicos, el SEI los define como: propiedades (es decir, atributos) de productos o servicios a adquirir o desarrollar.(SOFTWARE ENGINEERING INSTITUTE, 2010a)

En el sitio de INCOSE en (WILLIAMS, 2006) se define: los requisitos técnicos son características que el sistema debe poseer absolutamente.

Invariablemente algunos autores utilizan los términos requisitos del producto y requisitos técnicos para un mismo concepto. En la presente investigación se utilizará el término requisitos técnicos para referirse a aquellos requisitos que detallan técnicamente las necesidades del cliente, sirven para traducir esas necesidades al lenguaje de los desarrolladores y como entrada para el desarrollo posterior del diseño, construcción y pruebas del sistema. Se toma el concepto de requisito del cliente definido por el SEI debido a que sintetiza la idea que se quiere transmitir en este aspecto de una forma sencilla e integradora.

1.2 Ingeniería de Requisitos

Los requisitos tienen un rol protagónico en el éxito de un proyecto de software, todo lo relacionado con su obtención, análisis, desarrollo y administración es crítico para el proyecto y de la correcta realización de estas actividades depende el posterior cumplimiento de los objetivos establecidos.

Las actividades relacionadas con los requisitos se encuentran englobadas dentro de la Ingeniería de Requisitos (IR), que acorde a Pressman(PRESSMAN, ROGER S., 2002) facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. El proceso de ingeniería de requisitos puede ser descrito en cinco pasos: Identificación de Requisitos, Análisis de Requisitos y Negociación, Especificación de Requisitos, Modelado del Sistema, Validación de Requisitos y Gestión de Requisitos.

Podemos describirla también como el proceso de descubrir, analizar, documentar y verificar los requisitos para un sistema. (SOMMERVILLE, 2005)

Para Karl Wieggers está dividida en dos dominios: la administración de los requisitos y el desarrollo de los requisitos. A su vez el desarrollo de requisitos lo divide en: obtención, análisis, especificación y validación. (WIEGERS, KARL E., 2003)

Herrera plantea que (HERRERA, 2003), dentro de la IR existen cuatro actividades básicas que se tienen que cumplir para completar el proceso. Las mismas ayudan a reconocer la importancia que tiene para el desarrollo de un proyecto de software realizar una especificación y administración adecuada de los requisitos de los clientes o usuarios. Estas actividades son: Extracción, Análisis, Especificación y Validación.

Existe una coincidencia en las actividades descritas por diferentes autores situadas dentro de la Ingeniería de Requisitos: los requisitos deben ser obtenidos, analizados, especificados, validados y administrados.

El peso de las actividades de la IR se realiza en la fase o etapa de Requisitos. Aquí es donde se capturan, analizan, describen y validan y por tanto se elabora la mayoría de los productos de trabajo, elemento consumidor de esfuerzo, además de las actividades mencionadas. Las actividades de administración de requisitos continúan ejecutándose a lo largo del ciclo de vida del proyecto pero tienen un menor peso en cuanto al esfuerzo debido a que se realizan periódicamente (casi siempre al finalizar las iteraciones o fases) o dependen de eventualidades como es el caso del surgimiento de solicitudes de cambios.

1.3 Base de Conocimiento

Los requisitos son claves en el tema de la reutilización y la gestión del conocimiento pues pueden agregar información valiosa para futuros proyectos. De ahí la importancia de salvaguardar los datos importantes relacionados con estos en una base de conocimiento de la organización, desde donde se pueda obtener posteriormente elementos que apoyen la toma de decisiones.

Una base de conocimiento es un repositorio de productos de trabajo, donde se almacenan, por ejemplo, productos de software, planes, reportes, registros y otros documentos como contratos, propuestas, etc.

Las bases de conocimiento son la evolución lógica de los sistemas de bases de datos tradicionales, en un intento de plasmar no ya cantidades ingentes de datos, sino elementos de conocimiento (normalmente en forma de hechos y reglas) así como la manera en que estos elementos han de ser utilizados. (HERNÁNDEZ, CHANTAL PÉREZ, 2002).

1.4 Características de las Pequeñas y Medianas empresas (PYME)

La RAE define PYME como: empresa mercantil, industrial, compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación.

Las PYME se han convertido en el mayor sector empresarial tanto de los países desarrollados como los países en desarrollo. Puede ejemplificarse esta afirmación a través de (MÉNDEZ y

CARMEN, 2007) que plantean: en México las PYME representan más del 97% de las empresas, a diferencia de Estados Unidos, donde representan el 95% y en España el 99%. Estas proporciones por encima del 90% demuestran la importancia que tienen las PYME dentro de una economía.

Las PYME, tienen particular importancia para las economías nacionales, no solo por sus aportaciones a la producción y distribución de bienes y servicios, sino también por la flexibilidad de adaptarse a los cambios tecnológicos y gran potencial de generación de empleos. Representan un excelente medio para impulsar el desarrollo económico y una mejor distribución de la riqueza. Hoy día, los gobiernos de países en desarrollo reconocen la importancia de las PYME por su contribución al crecimiento económico, a la generación de empleo, así como al desarrollo regional y local.(COEPES, 2008)

A continuación se resumen las principales características de las PYME en las que coinciden algunos autores e instituciones autorizadas como (EUROPEAN COMMISSION, 2005; GALINDO, 2007; LEMES y MACHADO, 2007):

- Bajo número de empleados que va desde 51 a 250 personas.
- Su tamaño es pequeño o mediano en relación con las otras empresas que operan en el ramo.
- Bajo volumen de ventas anuales, en las pequeñas se encuentra entre 10 y 43 millones de euros y en las medianas entre 10 y 50 millones.
- Debido al pequeño volumen de beneficios que presentan estas empresas no pueden dedicar grandes fondos a la investigación.
- Tienen dificultades para encontrar financiación a un costo y plazo adecuados debido a sus riesgos inherentes.
- Capacidad de adaptabilidad gracias a su estructura pequeña, tienen más rapidez de movimientos.
- Posibilidad de especializarse en cada nicho de mercado ofreciendo atención directa.
- Está en proceso de crecimiento: la pequeña empresa tiende a ser mediana y esta aspira a ser grande.
- Obtienen algunas ventajas fiscales por parte del Estado, que algunas veces las considera causantes menores dependiendo de sus ventas y utilidades.

Las PYME debido a algunas de sus características, enfrentan un conjunto de problemas que frenan su desarrollo, su capacidad competitiva, productividad y por tanto acceso a los mercados tanto regionales como internacionales. En estos problemas influyen también

aspectos globales como los mencionados por (MEDINA, 2010) : medios financieros limitados, equipos de producción y maquinarias sencillos, personal reducido, sistemas de contabilidad y control sencillos, entre otros.

1.5 Productividad

Sea una PYME o una gran empresa y cualquiera su actividad fundamental tiene entre sus objetivos el aumento de la productividad. La productividad a través de numerosos estudios e investigaciones ha contado con varias definiciones y fórmulas para su cálculo. Una definición tradicional de productividad de software corresponde, por ejemplo, al número de líneas de código fuente (LOC) producidas por persona-mes de esfuerzo, aunque existen muchos problemas con esta definición, partiendo de una definición imprecisa de lo que es una línea de código.(...) Otra explicación se encuentra relacionada con que la codificación en lenguajes de alto nivel requiere un número menor de líneas de código que la codificación de bajo nivel.(VISCANTI, 2003)

Según la definición del economista Klein, la productividad consiste en “la adecuada relación que existe entre la meta lograda y los recursos gastados con ese fin”.

Generalmente se mide la productividad del trabajo, es decir, la producción anual de cada trabajador, cuantificando así qué cantidad de bienes o servicios es capaz de fabricar cada persona con empleo en un periodo determinado. (...) Por tanto, evalúa la cantidad de bienes que produce una empresa según el número de personas que trabajan en ella y la cantidad de tiempo, materiales y recursos necesarios para producir esos bienes.(GONZALO, 2013)

La Real Academia de la Lengua Española (RAE) la define como la relación entre lo producido y los medios empleados, tales como mano de obra, materiales, energía, etc. (REAL ACADEMIA ESPAÑOLA, 2010)

Sommerville menciona al respecto (SOMMERVILLE, 2004) que se puede medir la productividad en un sistema de manufactura contando el número de unidades producidas dividido entre el número de horas-personas requeridas para producirlas. Sin embargo en el software hay varias soluciones. (...) El estimado de la productividad se basa a menudo en medir atributos del software y dividirlos entre el esfuerzo requerido para el desarrollo. Respecto a esta, hay 2 tipos de métricas que han sido utilizadas:

- 1- Métricas de tamaño: relacionadas con alguna salida de una actividad. La métrica más utilizada relacionada con el tamaño es las líneas de código (LOC). Otras métricas son el número de instrucción de código objeto o el número de páginas en el caso de la documentación.

- 2- Métricas de funcionalidad: relacionadas con el total de funcionalidades del software entregado. La productividad es expresada en términos de cantidad de funcionalidades útiles producidas en un tiempo dado. Puntos de función y puntos de objeto son las métricas de este tipo más conocidas.

La productividad también suele estar asociada a la eficiencia y al tiempo: cuanto menos tiempo se invierta en lograr el resultado anhelado, mayor será el carácter productivo del sistema. En una empresa, la productividad es fundamental para crecer o aumentar la rentabilidad.(DEFINICIÓN, 2013)

En resumen, la productividad es igual a la relación existente entre el tamaño de lo que se produce (medido en LOC, puntos de función, puntos de objetos, cantidad de funcionalidades o cualquier otra medida de tamaño) y el tiempo que invierten las personas relacionadas para producido (esfuerzo).

1.6 Modelos y estándares de calidad empleados en la investigación

La productividad tiene un papel muy importante en aras de lograr ventaja competitiva y a la vez está muy relacionada con la calidad de lo que se produce. La utilización de normas y estándares de calidad, ha sido una de las soluciones que han encontrado varias empresas para asegurar la calidad de sus procesos y productos.

Pressman define calidad como: “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”.(PRESSMAN, ROGER, 2006)

A continuación se presentan las principales características de algunos estándares y modelos de calidad.

1.6.1 Normas ISO

La Organización Internacional para la Estandarización (ISO) promueve la estandarización internacional, de tal manera que se facilite el intercambio de bienes y servicios así como el desarrollo científico y tecnológico. En relación al software, existe la guía o reglas generales ISO 9000-3. Esta guía tiene un gran impacto en la industria del software. ISO 9000-3 es una guía y no una norma, que describe los elementos de un sistema de calidad orientado al software. Incluye algunos temas que no se encuentran en las normas ISO 9000 genéricas, tales como Administración de la Configuración o Planeación de Proyectos.(FEBLES, 2003)

En la **ISO 90003:2006** se encuentra dentro de la sección 7, Realización del producto, actividades referentes a los requisitos de software.

Dentro de esta sección se especifican elementos primordiales como los relacionados con los procesos del cliente, la comunicación con él y su representante. Se enfatiza en los requisitos adicionales determinados por la organización, los requisitos relacionados con el producto y el cliente. Incluye también elementos de gestión de cambios remarcando la importancia del establecimiento de un método para la trazabilidad desde los requisitos hasta el producto final. Así como la necesidad de que en los requisitos se tenga en cuenta el ambiente de operación. Establece el registro de las revisiones realizadas a los requisitos y las acciones que se deriven de estas. Propone una serie de riesgos a tener en cuenta en el proyecto y su periódica gestión y actualización entre otras generalidades que resumen buenas prácticas en la administración y desarrollo de los requisitos.

La norma **ISO 12207:2005** establece una arquitectura de alto nivel del ciclo de vida del software. El ciclo de vida comienza con una idea o una necesidad que puede ser satisfecha completa o parcialmente por el software y termina con el retiro de este. La arquitectura es construida con un conjunto de procesos y relaciones entre estos procesos. (ISO 12207, 2005)

Los procesos son agrupados en 3 clases: primarios, de soporte y organizacional. Dentro de los procesos primarios se encuentra el de Desarrollo, donde se plantean elementos relacionados con los requisitos como son las actividades de análisis de requisitos del sistema y análisis de requisitos de software. Ver Anexo 5.

Las normas ISO en su generalidad son estándares difíciles de entender debido en su mayor parte a que se encuentran desagregadas en varios documentos que tienden a confundir a los interesados en su implementación. Estas normas no definen elementos fundamentales como una propuesta de proceso, qué roles deben ejecutar las actividades, cuáles técnicas o herramientas pueden utilizarse para apoyar el proceso. Aunque algunas ISO hacen referencia a las mediciones no realizan una propuesta detallada de indicadores concretos a medir y no se hace alusión a la gestión explícita del conocimiento.

Presentan además un alto costo para la organización en temas de consultoría especializada, esfuerzo necesario para la implementación de las actividades de mejora y la certificación, además del pago por la obtención de la norma que no se distribuye gratuitamente, costos que pueden asumir las grandes empresas pero representan una barrera para las PYME.

1.6.2 Modelo de Madurez y Capacidades Integrado (CMMI)

CMMI es un modelo de madurez de mejora de procesos para el desarrollo de productos y servicios. Incluye las mejores prácticas que tratan las actividades de desarrollo y

mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento.(CMMI, 2006)

Fue creado por el SEI y es el resultado de la integración de varios modelos. CMMI cuenta con 22 áreas de procesos. Un área de proceso es un conjunto de prácticas relacionadas que cuando son implementadas colectivamente, satisfacen un conjunto de objetivos considerados importantes para mejorar esa área de proceso. (CMMI, 2006)

Dos de las áreas del modelo trabajan el tema de los requisitos del software: Administración de Requisitos y Desarrollo de Requisitos.

El propósito de la Administración de Requisitos (REQM) es gestionar los requisitos de los productos y de los componentes del producto del proyecto, e identificar inconsistencias entre esos requisitos y los planes y productos de trabajo del proyecto. (SOFTWARE ENGINEERING INSTITUTE, 2010b)

Esta área de procesos está compuesta por un solo objetivo específico: administrar los requisitos; que contiene a su vez 5 prácticas específicas (SP) que orientan de una forma más detallada cómo cumplir con el objetivo.

Objetivo específico 1. Administrar los requisitos

SP 1.1 Comprender los requisitos.

SP 1.2 Obtener compromiso sobre los requisitos.

SP1.3 Gestionar los cambios a los requisitos.

SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.

SP 1.5 Asegurar alineación entre el trabajo del proyecto y los requisitos.

Estas prácticas señalan que los requisitos no solo estén claramente identificados, sino también que todos los involucrados en el proyecto (cliente - equipo de desarrollo) estén de acuerdo en su significado. Otras de las prácticas en esta área de proceso es la de controlar y documentar de manera adecuada los cambios en los requisitos para de esta forma no perder el control del plan del proyecto.

Un aspecto también importante es el de trazabilidad bidireccional. El equipo de desarrollo debe estar en condiciones de relacionar cuál ha sido el origen de los mismos, cuál es la relación entre los requisitos de bajo nivel y los de alto nivel (por ejemplo, cuáles son derivados y de cuál requisito), cuáles son los artefactos relacionados con los requisitos (por ejemplo, especificaciones, documentos de diseño o planes), y cuáles componentes del producto implementan cada requisito. Esta práctica es sumamente importante para poder realizar un

buen análisis de impacto ante posibles cambios, y fundamental para poder determinar si el alcance del proyecto ha sido cubierto o no.

Respecto al área de Desarrollo de Requisitos (RD), su propósito es obtener, analizar, y establecer requisitos del cliente, del producto y de los componentes de producto.(SOFTWARE ENGINEERING INSTITUTE, 2010b) Esta área contempla 3 objetivos específicos: desarrollar requisitos del cliente, desarrollar los requisitos del producto y analizar y validar requisitos. Dentro de estos objetivos específicos se encuentran las prácticas específicas que responden a cada uno de ellos.

Objetivo específico 1. Desarrollar requisitos del cliente.

SP1.1 Obtener necesidades.

SP1.2 Transformar las necesidades del cliente en requisitos del cliente.

Objetivo específico 2.Desarrollar los requisitos del producto.

SP2.1 Establecer requisitos del producto y sus componentes.

SP2.2 Asignar requisitos a los componentes del producto.

SP2.3 Identificar los requisitos de interfaz.

Objetivo específico 3. Analizar y validar requisitos.

SP3.1 Desarrollar concepto de operación y escenarios.

SP3.2 Desarrollar una definición de requisitos de funcionalidad y atributos de calidad.

SP3.3 Analizar requisitos.

SP3.4 Analizar requisitos para lograr un balance.

SP3.5 Validar requisitos.

Como parte del desarrollo de los requisitos es importante definir correctamente qué necesita el cliente e ir desarrollando técnicamente esas necesidades. También analizar los requisitos, teniendo en cuenta el costo, criticidad y riesgos asociados, de forma tal que permita establecer un balance y priorización para su realización. Otra de las prácticas asociadas es la de ubicar los requisitos en el sistema identificando elementos iniciales de la arquitectura como las interfaces de comunicación y los componentes de producto. La validación de que esos son realmente los requisitos que pidió el cliente es uno de los pasos más importantes en el desarrollo. Este paso le asegura al proyecto que se encuentra en el camino correcto hacia el logro de sus objetivos.

Luego del análisis realizado podemos concluir que CMMI contiene muchas buenas prácticas relacionadas con los requisitos pero no realiza una propuesta del proceso, de qué roles del

proyecto pudieran interactuar, así como tampoco propone posibles herramientas que permitan automatizar algunas actividades del proceso.

En estudios realizados por el SEI se ha comprobado que el tiempo de implementación que las organizaciones necesitan es considerable: para pasar del nivel 1 al 2 se necesitan como promedio 23 meses, del nivel 2 al 3, 20 meses, del 3 al 4, 25 meses y del 4 al 5, 13 meses.(SOFTWARE ENGINEERING INSTITUTE, 2005)

En total, una empresa necesitaría unos 81 meses (6.8 años) para alcanzar el nivel 5 de CMMI.

Además del tiempo de implementación, aunque el modelo asegura no realizar cambios drásticos sino trabajar sobre lo que ya se realiza en la organización, se necesitan muchos recursos tanto humanos como materiales para enfrentar un SPI basado en CMMI. El costo también es un parámetro importante en esta ecuación, aún cuando el acceso al modelo es gratuito, es necesario el apoyo de una entidad consultora para la interpretación del modelo debido a que no es fácil de entender.

Incluso cuando una organización ya ha asumido el costo de una consultoría y del esfuerzo necesario para obtener un nivel de madurez o capacidad teniendo en cuenta los recursos que conlleva, debe enfrentarse al final, al costo de la evaluación del modelo, evaluación que debe actualizarse cada 3 años. Estos elementos hacen que CMMI no sea un modelo adecuado para las PYME.

1.6.3 Modelo Brasileño MPS.br

El MPS.BR es un programa movilizador, de largo plazo, creado en diciembre de 2003, coordinado por la Asociación para Promoción de la Excelencia del Software Brasileño (SOFTEX). (SOFTEX, 2009)

MPS posee tres componentes: Modelo de Referencia (MR-MPS), Método de Evaluación (MA-MPS) y Modelo de Negocio (MN-MPS).La relación entre los componentes del modelo se muestra en el Anexo 6.

En el Modelo de Referencia (MR-MPS), MPS posee 2 áreas de procesos para definir las actividades de administración y desarrollo de requisitos que se encuentran en el nivel G y D.

El proceso Administración de Requisitos se encuentra en el nivel G Parcialmente Gestionado y su propósito es gestionar los requisitos del producto y componentes del producto del proyecto e identificar inconsistencias entre los requisitos y los planes del proyecto y los productos de trabajo del proyecto.(SOFTEX, 2009)

Los resultados esperados de este proceso son (SOFTEX, 2009):

GRE 1. Los requisitos son entendidos, evaluados y aceptados en conjunto con los proveedores de requisitos, utilizando criterios objetivos.

GRE 2. El compromiso del equipo técnico con los requisitos aprobados es obtenido.

GRE 3. La rastreabilidad bidireccional entre los requisitos y los productos de trabajo es establecida y mantenida.

GRE 4. Revisiones en planes y productos de trabajo del proyecto son realizadas con el objetivo de identificar y corregir inconsistencias relacionadas a los requisitos.

GRE 5. Cambios en los requisitos son gestionados durante el proyecto.

El proceso de Desarrollo de Requisitos se encuentra en el nivel D y 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.

Este modelo se enfoca en la búsqueda (SOFTEX, 2009) de que sea adecuado al perfil de empresas con diferentes tamaños y características, públicas y privadas, aunque con especial atención a las micro, pequeñas y medianas empresas. Posee 19 procesos, lo que representan un gran cúmulo de actividades a realizar por el proyecto, que en determinado momento puede llegar a sentirse abrumado por la cantidad de procesos y desistir de la iniciativa de la mejora.

Dentro del modelo se brindan descripciones detalladas de las actividades y algunas técnicas a utilizar, sin embargo, no hace una propuesta de roles ni herramientas, tampoco define indicadores como parte del proceso. Aunque en la gestión de los recursos humanos propone que las empresas establezcan una estrategia para la gestión del conocimiento, no propone desde los procesos aquellos elementos que sirven de entrada para esa estrategia.

Este modelo aclara que (SOFTEX, 2009): “Las actividades y tareas necesarias para lograr el propósito y los resultados esperados no son definidas en esta guía (MPS.BR – Guía General:2009), debiendo quedar a cargo de los usuarios del modelo.”

Para emprender el largo camino de la mejora de procesos mediante la implementación de este modelo las empresas deben dedicar esfuerzo para la definición de las actividades de cada proceso desde cero, deben elaborar los productos de trabajo sin tener una guía que aminore este trabajo. Las PYME necesitan un modelo ligero, que les provea la mayor cantidad de elementos posibles en aras de ahorrar esfuerzos.

1.6.4 Modelo mexicano MOPROSOFT

En el 2002, la Secretaría de Economía, inició el Programa para el Desarrollo de la Industria de Software en México. Así surge el proyecto que da vida al modelo y actualmente aplicable como norma mexicana, MOPROSOFT.

Los procesos de MOPROSOFT abarcan las responsabilidades asociadas a la estructura de una organización que son: la Alta Dirección, Gestión y Operación.(OKTABÁ *et al.*, 2005) Ver Anexo 7

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.

Aunque este modelo está enfocado en las PYME, 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 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 (aunque no de forma detallada), identifican los productos de trabajo resultado de las actividades y cuáles de estos deben formar parte de la base de

conocimiento de la organización. Pero pasan por alto un elemento importante: una propuesta de técnicas y herramientas que le permitan al proyecto apoyar la realización de las actividades que las necesitan.

En el Anexo 8 se muestra un cuadro resumen comparativo de las normas y modelos revisados con los elementos identificados en los acápites anteriores.

Conclusiones parciales del capítulo

- La Ingeniería de Requisitos juega un papel fundamental en los proyectos de desarrollo de software pues es donde se llevan a cabo las actividades de comunicación con el cliente para obtener los requisitos de lo que se construirá. Es donde estos se gestionan, analizan, especifican y validan. En estas actividades es donde la mayoría de los proyectos fallan y se encuentran las principales causas de su cancelación.
- Las PYME enfrentan el problema de aumentar su productividad para competir en el mercado contando con limitantes relacionadas principalmente con la cantidad de personal, variable que les impide sacrificar demasiado esfuerzo, dedicado principalmente a la producción, para redirigirlo en aras de iniciativas de mejora.
- Los modelos y estándares de calidad internacionales se han convertido en la vía principal para asegurar un buen desempeño, identificando pasos claves que no se deben ignorar a la hora de realizar un proyecto. La mayoría de estos contienen procesos enfocados en las grandes empresas y requieren un alto costo en conceptos de tiempo, personal y dinero, haciéndolos inadecuados para las PYME.
- A partir de esta problemática, es necesario definir un proceso de ingeniería de requisitos que permita enfocarse en los mayores problemas (los requisitos) a través de la realización de buenas prácticas normadas internacionalmente, con un bajo costo, de forma que contribuya a aumentar la productividad.

Capítulo 2: Propuesta del Proceso Base de Ingeniería de Requisitos

2. Introducción

En este capítulo se presenta un proceso sencillo para desarrollar y administrar los requisitos de software. Se realiza una propuesta de roles, productos de trabajo que se esperan como resultado de la ejecución de las actividades y plantillas para su realización además de una propuesta de cuáles deben formar parte de la base de conocimiento.

2.1 Proceso Base de Ingeniería de Requisitos (PB de IR)

A continuación se muestra el propósito del proceso definido por la autora, sus directrices, la descripción gráfica y textual y los productos de trabajo, especificando aquellos que formarán parte de la base de conocimiento de la organización.

2.1.1 Propósito

El propósito del proceso es definir y desarrollar los requisitos del cliente y los requisitos técnicos, controlar sus cambios e identificar inconsistencias entre estos y sus productos de trabajo derivados.

2.1.2 Directrices del Proceso Base de Ingeniería de Requisitos

- Definir los requisitos.
- Obtener el compromiso con los requisitos.
- Analizar y desarrollar los requisitos.
- Gestionar los cambios a los requisitos.
- Identificar las inconsistencias entre el trabajo del proyecto y los requisitos.
- Validar los requisitos.
- Gestionar indicadores y lecciones aprendidas del proceso.

2.1.3 Descripción gráfica del Proceso Base de Ingeniería de Requisitos

A continuación se presenta el PB de IR a través de una descripción gráfica, ver figuras 2 y 3. En estas figuras se muestra el nombre del proceso, los criterios de entrada y salida, las columnas donde se describen los roles, las entradas necesarias para ejecutar la actividad, las guías o apoyos traducidos en elementos que sirven como herramientas para la creación de las salidas, luego se muestra el nombre de la actividad en un recuadro con el flujo de las actividades verticalmente y en la última columna los productos de salida.

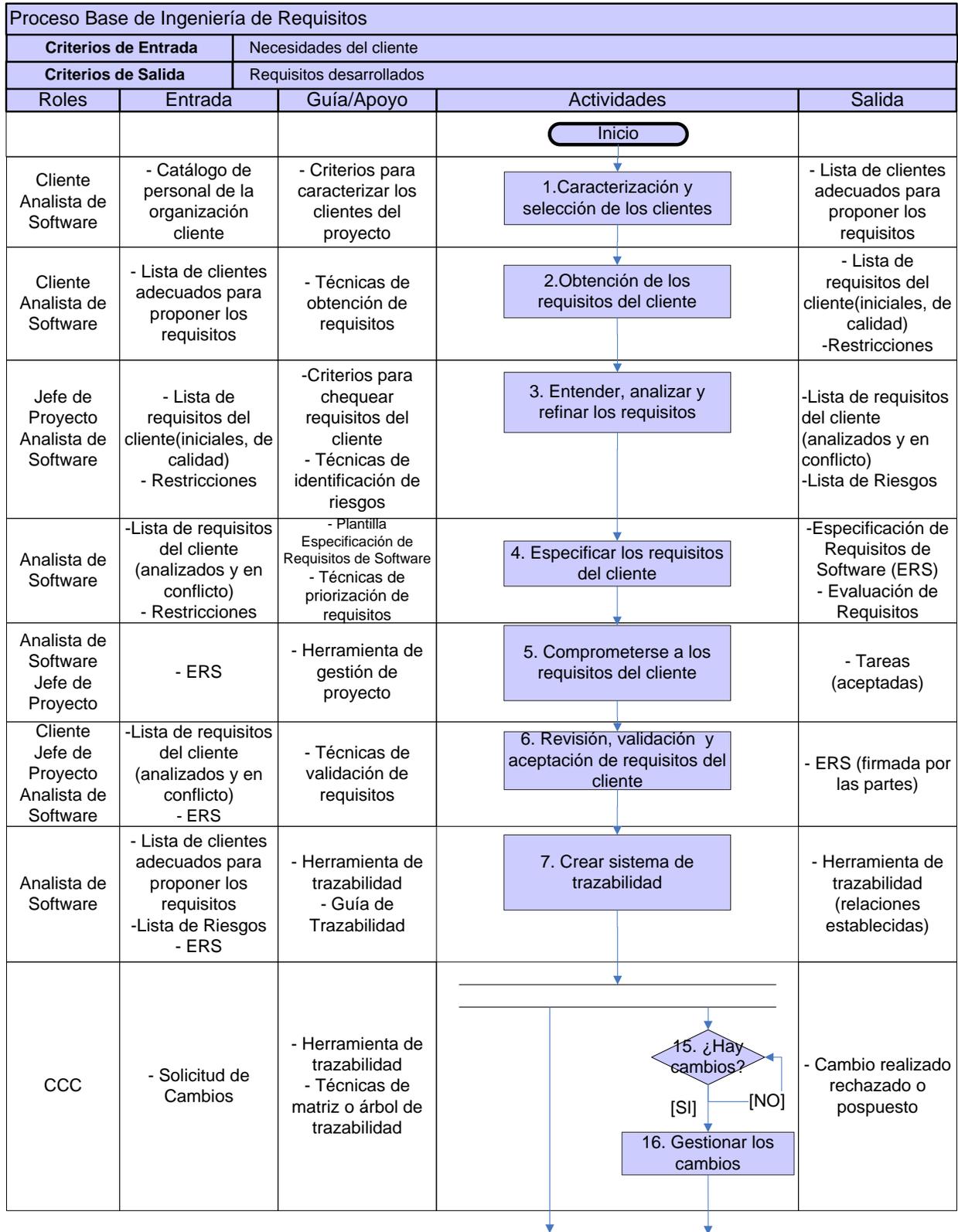
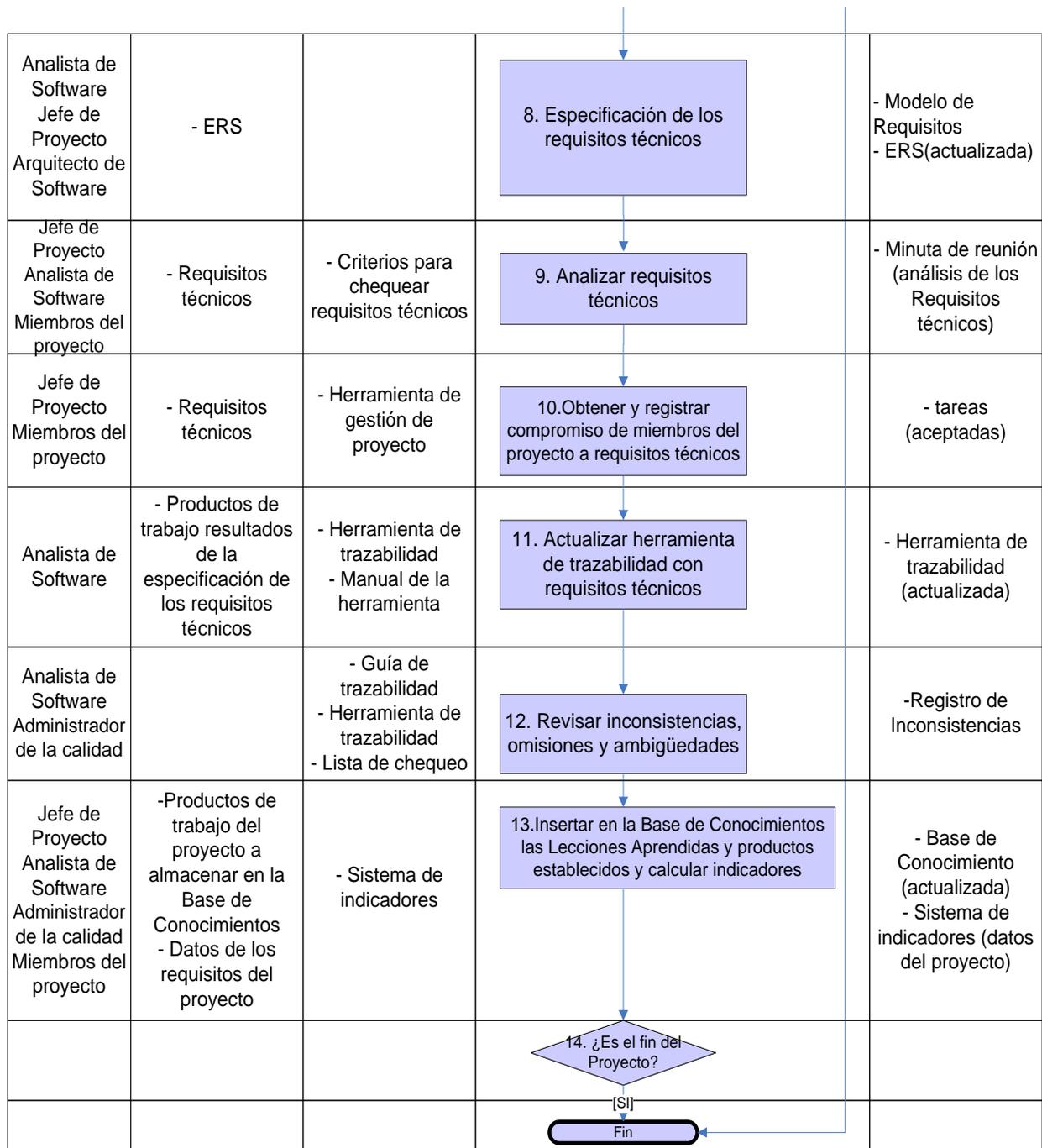


Figura 2. Descripción gráfica del proceso parte 1



En caso de no ser el fin del proyecto, significa que el proyecto va a desarrollar otra iteración, entonces regresa a la actividad a partir de la cual comenzará la próxima iteración

Figura 3. Descripción gráfica del proceso parte 2

2.1.4 Descripción Textual del Proceso Base de Ingeniería Requisitos

Una idea puede comunicarse más fácilmente a través de una imagen que de un texto, pero en ocasiones es necesario transmitir detalles que mediante una imagen solamente, no es posible. En ese caso se encuentran los procesos, donde además de una descripción gráfica, enriquece y da una mayor claridad combinarla con una descripción textual. A continuación se presenta la descripción textual del PB de IR.

Actividad 1.

1.1 El cliente entrega un Catálogo de Personal al Analista de Software que sirve como entrada para caracterizar a cada uno de los clientes del proyecto acorde a criterios previamente establecidos.

1.2 El Analista de Software en conjunto con el cliente caracterizan a cada uno de ellos y realizan una selección inicial que es presentada luego al responsable de la organización cliente. Entre este responsable y el proyecto acuerdan quienes serán los clientes seleccionados para proveerle los requisitos a los Analistas de Software.

Salidas: Lista de clientes adecuados para proponer los requisitos.

Actividad 2.

2.1 Los Analistas de Software se reúnen y seleccionan las técnicas adecuadas para obtener los requisitos.

2.2 Aplican esas técnicas y obtienen unos requisitos iniciales como resultado. (Lista de requisitos del cliente)

2.3 Obtienen además los requisitos de calidad (requisitos no funcionales) y una primera versión de las posibles restricciones del proyecto que plasman en el Plan del Proyecto.

Salidas: Lista de requisitos del cliente (requisitos del cliente iniciales, requisitos de calidad), Restricciones.

Actividad 3.

3.1 Los requisitos del cliente se chequean contra criterios establecidos.

3.2 Se analiza si existen requisitos de proyectos anteriores que puedan reutilizarse.

3.3 Se define la Lista de requisitos del cliente y se identifican requisitos en conflicto y se realizan propuestas de cómo eliminar esos conflictos. Esas propuestas son presentadas posteriormente al cliente.

3.4 El Jefe de Proyecto identifica y evalúa los riesgos relacionados con los requisitos (en cuanto a costo, tiempo, rendimiento) utilizando técnicas de identificación de riesgos.

Salidas: Lista de requisitos del cliente (con los requisitos analizados y los requisitos en conflicto), Lista de Riesgos (riesgos relacionados con los requisitos, insertados en el Plan de riesgos del proyecto).

Actividad 4.

4.1 El Analista de Software prioriza los requisitos utilizando técnicas de priorización y evalúa su complejidad (documento Evaluación de Requisitos).

4.2 El Analista de Software refina los requisitos (en cuanto a descripción y detalle de la funcionalidad), realiza una búsqueda para encontrar nuevos requisitos que pueden estar implícitos (principalmente los referentes a los atributos de calidad), define el modelo conceptual, estableciendo las relaciones de las entidades del subsistema y sus atributos fundamentales, futuras clases persistentes y candidatas al modelo de datos.

4.3 Los Analistas de Software especifican los requisitos del cliente utilizando la Plantilla Especificación de Requisitos de Software.

Salidas: Especificación de Requisitos de Software (ERS) (Priorización de los requisitos, su análisis y detalle, modelo conceptual), Evaluación de Requisitos.

Actividad 5.

5.1 El Jefe de Proyecto les asigna la realización de los requisitos del cliente a los miembros del proyecto responsables de estos, (en su mayoría Analistas de Software) mediante la asignación de tareas en una herramienta de gestión de proyecto.

5.2 Estas personas aceptan la tarea y con esto los requisitos que fueron previamente entendidos, analizados y especificados.

Salidas: tareas (aceptadas).

Actividad 6.

6.1 El Jefe de Proyecto y Analista de Software se reúnen con el cliente para validar la Especificación de Requisitos. En este encuentro se revisan los requisitos en conflicto y se llegan a acuerdos bilaterales. Se utilizan técnicas de validación de requisitos para apoyar la negociación.

6.2 Se firma la ERS por ambas partes.

Salidas: ERS (firmada por las partes).

Actividad 7.

7.1 El Analista de Software orientándose por la Guía de trazabilidad, inserta en la herramienta seleccionada para llevar la trazabilidad, los requisitos pactados, los clientes que los reportaron y los riesgos identificados (relacionados con los requisitos). Se establecen relaciones bidireccionales entre ellos.

Salidas: requisitos, clientes y riesgos insertados en la herramienta de trazabilidad (relaciones establecidas).

Actividad 8.

8.1 El Analista de Software modela los requisitos (descripción más técnica: Modelo de casos de uso, Modelo del dominio, Escenarios de operación, etc.)

8.2 El Jefe de Proyecto y el Arquitecto de Software distribuyen los requisitos en los módulos o subsistemas del proyecto (ubicación de los requisitos).

8.3 El Arquitecto de Software identifica los requisitos de interfaz (entre componentes: relación técnica).

8.4 El Jefe de Proyecto planifica la distribución de las tareas para desarrollar los requisitos.

Salidas: Modelo de requisitos, Especificación de Requisitos de Software (actualizada con requisitos técnicos distribuidos en subsistemas o módulos, requisitos de hardware y escenarios de operación: diagrama de despliegue).

Actividad 9.

9.1 El Jefe de Proyecto realiza una reunión con el equipo para explicar la distribución de las tareas para el desarrollo de los requisitos.

9.2 El Jefe de Proyecto y Analistas de Software explican los requisitos a desarrollar y les proveen los elementos necesarios (requisitos técnicos) para que el equipo comience su trabajo, durante la distribución de las tareas relacionadas con los requisitos técnicos, el equipo los chequea contra los criterios establecidos como parte del análisis de estos.

Salidas: Minuta de reunión (Análisis de los requisitos técnicos).

Actividad 10.

10.1 El Jefe de Proyecto les asigna la realización de los requisitos técnicos a los miembros del proyecto responsables de estos, mediante la asignación de tareas en una herramienta de gestión de proyecto.

10.2 Estas personas aceptan las tareas y con esto la realización de los requisitos que fueron previamente analizados.

Salidas: tareas (aceptadas).

Actividad 11.

11.1 El Analista de Software teniendo en cuenta la Guía de trazabilidad, actualiza la herramienta de trazabilidad con los nuevos artefactos creados a partir de los requisitos (Modelo de Requisitos, Documento de Arquitectura, etc.).

Salidas: Requisitos técnicos relacionados con requisitos del cliente (herramienta de trazabilidad actualizada).

Actividad 12.

12.1 El Analista de Software y el Administrador de la Calidad utilizando una lista de chequeo revisan en busca de inconsistencias que pudieran existir entre los requisitos del cliente, los productos creados a partir de estos y los planes del proyecto. Deben asignarse responsables para corregir las inconsistencias encontradas.

12.2 Las inconsistencias deben ser detectadas y monitoreadas hasta su cierre.

Salidas: Registro de Inconsistencias.

Actividad 13.

13.1 El Administrador de la Calidad apoyándose en el Jefe de Proyecto y Analista de Software recolecta la información necesaria y realiza los cálculos de los indicadores.

13.2 El Jefe de Proyecto junto a su equipo definen las lecciones aprendidas durante la ejecución del proceso.

13.3 El Jefe de Proyecto las centraliza en el documento “Lecciones aprendidas” y las inserta en la base de conocimiento junto a los productos de trabajo que se encuentran bajo su responsabilidad (predeterminados a formar parte de la base de conocimiento)

13.4 Los miembros del proyecto insertan en la base de conocimiento los productos de trabajo que se encuentran bajo su responsabilidad.

Salidas: Base de Conocimiento (actualizada con lecciones aprendidas y productos de trabajo), Sistema de indicadores (con los datos del proyecto).

Pregunta 14.

14.1 Si es el fin del proyecto, ir a fin.

14.2 Si no es el fin del proyecto, regresar a la actividad a partir de la cual se decidió comenzar otra iteración.

Pregunta 15.

15.1 En caso que se solicite un cambio a los requisitos gestionar el cambio acorde a lo planteado en la actividad 16.

Actividad 16.

16.1 La solicitud de un cambio a algún requisito debe ser debidamente documentada, especificando el interesado en el cambio, la prioridad del cambio, el impacto, la cantidad de elementos que se afectan (utilizando técnicas como la matriz y árboles de trazabilidad), el costo del cambio, entre otros elementos de interés del proyecto.

16.2 Esta solicitud debe ser revisada por el Comité de Control de Cambios (CCC) del proyecto, integrado por: el Jefe de Proyecto, Arquitecto de software, Analista de Software, Administrador de la Calidad y otros miembros involucrados con el elemento a cambiar.

16.3 El CCC luego de revisar detenidamente la solicitud y evaluado el costo y beneficio del cambio debe tomar una decisión y comunicarla a los interesados. Si la decisión es implementar el cambio, deben ser asignadas la tareas a los responsables y verificar su cumplimiento.

Salidas: Cambio realizado, rechazado o pospuesto.

2.2 Roles del PB de IR

Un proceso necesita recursos humanos y materiales para su ejecución. Es necesario el compromiso de los miembros del proyecto para llevar a cabo todas las actividades que se definan. Pero no sólo el compromiso es importante, también cada miembro del proyecto debe tener bien definido el rol a cumplir y las responsabilidades a asumir. A continuación se definen los roles involucrados en el PB de IR, y sus responsabilidades son resumidas en el Anexo 9:

- Administrador de la calidad
- Analista de software
- Arquitecto de software
- Jefe de proyecto
- Cliente

2.3 Productos de trabajo

Los productos de trabajo son el resultado de la ejecución de una actividad, son artefactos asociados a las salidas de un proceso. Como parte de la definición del PB de IR se encuentran los siguientes:

1. Lista de clientes adecuados para proponer los requisitos.
2. Lista de requisitos del cliente.
3. Lista de Riesgos (relacionados con los requisitos).
4. Especificación de Requisitos de Software (ERS).
5. Evaluación de Requisitos (como complemento de la ERS).
6. Modelo de requisitos.
7. Registro de Inconsistencias.
8. Solicitudes de Cambio.
9. Lecciones Aprendidas e indicadores del proceso

Estos productos de trabajo junto a las guías y descripción del proceso se encuentran en un CD adjunto, listos para su utilización en las empresas.

2.4 Productos de trabajo a incorporar en la Base de Conocimientos

Existen productos de trabajo que por sí solos o en ocasiones combinados con otras informaciones, son capaces de registrar datos históricos de gran utilidad para tomar decisiones posteriores dentro del mismo proyecto o para el desarrollo de nuevas ideas. Es necesario entonces almacenarlos en una base de conocimiento que permita organizar esta información y poder acceder a ella de la forma y en el momento que se necesite.

De los productos de trabajo del PB de IR, forman parte de la base de conocimiento:

- Especificación de Requisitos de Software y Evaluación de Requisitos (complemento de la ERS)

La ERS es uno de los productos de trabajo más significativo relacionado con los requisitos. Recoge tanto los requisitos funcionales como los no funcionales, por lo que es de vital importancia almacenarlo para mantener registros en caso de que se creen proyectos similares. A partir de estos se puede reutilizar la descripción de los requisitos, su complejidad, criticidad, etc.

- Lista de Riesgos

La lista de riesgos es un producto de trabajo que se mantiene vivo a lo largo del proyecto. Esta lista debe incluirse en el Plan de Riesgos del proyecto, documento que debe formar parte de la

base de conocimiento, ahí se recoge además de la lista de riesgos, los planes de mitigación y contingencia, elementos reutilizables que sirven de experiencia a otros proyectos.

- Registro de inconsistencias.

Muchos de los errores o inconsistencias que se producen en un sistema determinado se vuelven a repetir en la realización de otros similares, por lo que este producto de trabajo debe estar incluido en la base de conocimiento con el fin de agilizar la toma de decisiones respecto a esas inconsistencias y de esa forma incidir en la disminución del tiempo de duración del proyecto.

- Solicitudes de cambio

Los cambios son elementos de máxima importancia en un proyecto, deben ser cuidadosamente gestionados. Mantener un registro de las solicitudes de cambio que han ido sucediendo en el proyecto y las decisiones tomadas a partir de estas, es de mucha utilidad para el propio proyecto y para su reutilización.

- Lecciones aprendidas e indicadores del proceso

Las lecciones aprendidas del proyecto ya sean de carácter técnico, de gestión del proyecto, así como de los procesos que se siguen, son de un alto valor para futuros proyectos. Permiten a la organización no solo mejorar su forma de trabajo sino mantener el enfoque a procesos. El sistema de indicadores debe salvaguardarse para su posterior consulta en próximas iteraciones o por proyectos similares. Sirve de base para retroalimentar estimaciones y con datos históricos suficientes, predecir comportamientos futuros.

2.5 Mapa de compatibilidad

El PB de IR propuesto, asegura a las empresas que lo implementen, su compatibilidad con sus procesos homólogos en los modelos más populares de la industria de software, que fueron además analizados en el capítulo 1. En caso que en un momento determinado, alguna empresa desee y cuente con los recursos para realizar una evaluación o certificación de esos procesos homólogos en modelos o normas internacionales, puede hacerlo sin problemas a partir de la implementación de este proceso. En el Anexo 10 se presenta una tabla con la compatibilidad del PB de IR respecto a sus homólogos en algunas normas ISO, el modelo CMMI, Moprosoft y MPS.br.

Conclusiones parciales del capítulo

- El Proceso Base de Ingeniería de Requisitos es un proceso ligero y fácil de entender. Contiene una descripción gráfica y textual que sirve de base para su implantación en las empresas. Propone los roles que debe tener el proyecto y cuáles son sus

responsabilidades, de manera que el proyecto no tiene la necesidad de definirlos. Cuenta también con un conjunto de productos de trabajo determinados y plantillas para elaborarlos.

- Se realiza una propuesta de los productos de trabajo del proceso que formarán parte de la base de conocimiento de la organización.
- El proceso asegura su compatibilidad con sus homólogos en las normas y estándares de calidad revisados, posibilitando la certificación o evaluación por cualquiera de estos en el momento que se desee.

Capítulo 3. Propuesta de técnicas, indicadores y herramientas para el PB de IR

3. Introducción

En este capítulo se presenta una propuesta de técnicas y herramientas para apoyar la implementación del PB de IR. También un sistema de indicadores que constituye un aporte para la información necesaria que asegura la evaluación y mejora continua.

3.1 Técnicas para administrar y desarrollar los requisitos

A continuación se muestran un conjunto de técnicas recomendadas por cada actividad del proceso donde pueden ser aplicadas. No se propone una técnica específica pues no existe la técnica ideal sino la técnica adecuada. Depende de las características de cada proyecto y sus necesidades la selección de la técnica, o la combinación de estas, que suplan sus exigencias.

Actividad 2. Obtención de los requisitos del cliente

La obtención de requisitos se define como el proceso de identificar las necesidades del negocio, solucionando las posibles disparidades entre las personas involucradas en el mismo, con el propósito de definir y destilar los requisitos para cumplir las restricciones impuestas por las distintas partes.(INTECO, 2008)

Es importante tener clara la importancia de esta actividad, a partir de la cual se define qué se va a hacer en el proyecto y por tanto regirá los pasos restantes.

En (INTECO, 2008) se especifica que es más barato y efectivo descubrir y capturar todos los requisitos durante esta primera fase de búsqueda de requisitos. Aquellos requisitos que no sean descubiertos durante esta fase aparecerán cuando el usuario empiece a operar con el producto, y por lo tanto será mucho más costoso hacer los cambios necesarios para acomodar los nuevos requisitos encontrados.

Para asegurar la calidad de esta actividad pueden utilizarse variadas técnicas. Entre las propuestas por algunos autores se encuentran: entrevistas, tormenta de ideas, casos de uso, prototipos, escenarios de operación, storyboards, mapas conceptuales, técnica del aprendiz, JDA (del inglés Joint Application Development), entre otros.(DURAN y BERNÁRDEZ, 2000; ESCALONA y KOCH, 2002; INTECO, 2008; MENDOZA, 2012; PÉREZ, KARINA, 2007; PÉREZ, YOISY, 2009)

- **Entrevista:** la entrevista se define como la conversación de dos o más personas en un lugar determinado para tratar un asunto. Técnicamente es un método de investigación

científica que utiliza la comunicación verbal para recoger informaciones en relación con una determinada finalidad. (LÓPEZ y DESLAURIERS, 2011).

Las entrevistas resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. (DURAN y BERNÁRDEZ, 2000) Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y análisis de los resultados. La entrevista no es una técnica sencilla de aplicar, pues requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados para obtener de ellos toda la información posible en un período de tiempo siempre limitado. (PÉREZ, YOISY, 2009)

- **Encuestas:** es el método de recogida de información cuantitativa que consiste en interrogar a los miembros de una muestra, sobre la base de un cuestionario perfectamente estructurado. Además la encuesta es una técnica de investigación que consiste en una interrogación verbal o escrita que se le realiza a las personas con el fin de obtener determinada información necesaria para una investigación. (ALELÚ *et al.*, 2011) La técnica de encuesta es ampliamente utilizada como procedimiento de investigación, ya que permite obtener y elaborar datos de modo rápido y eficaz.(...) este procedimiento de investigación que posee, entre otras ventajas, la posibilidad de aplicaciones masivas y la obtención de información sobre un amplio abanico de cuestiones a la vez. (CASAS ANGUITA *et al.*, 2003)
- **Tormenta de ideas (brainstorming):** es la técnica más conocida para generar ideas. Fue desarrollada por Alex Osborn (especialista en creatividad y publicidad) en los años 30 y publicada en 1963 en el libro "Applied Imagination". Esta es una técnica de trabajo grupal donde sus objetivos principales son: romper las limitaciones habituales del pensamiento y producir un conjunto de ideas entre las que poder escoger. El Brainstorming es útil para atacar problemas específicos (más que los generalistas) y allí donde hace falta una colección de ideas buenas, nuevas y frescas. (HARRIS, 2003)
- **Mapas Conceptuales:** los mapas conceptuales son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo

de desarrollo hace el esfuerzo de elaborarlos en un lenguaje común. Sin embargo, deben ser usados con cautela porque en algunos casos pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual. (ESCALONA y KOCH, 2002; PÉREZ, YOISY, 2009)

- **Casos de Uso:** los casos de uso proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales centrándose en el valor añadido para el usuario. (JACOBSON *et al.*, 2000) Los casos de uso describen una secuencia de acciones que realiza un sistema con un resultado de valor perceptible para un usuario/actor. (SOMMERVILLE, 2004) Permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente. En comparación con métodos tradicionales de requisitos, son relativamente fáciles de escribir y más fácil de leer, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual detallada o con otra técnica como pueden ser los diagramas de actividades. (PÉREZ, YOISY, 2009)
- **Storyboards:** el propósito de las storyboards es obtener reacciones tempranas de los usuarios. Estas pueden ser pasivas, activas o interactivas. Las storyboards identifican a los involucrados, explican qué pasa y cómo. Tal vez ninguna técnica de obtención de requisitos ha sido objeto de tantas interpretaciones como la técnica de storyboard. Sin embargo, la mayoría de estas interpretaciones coinciden en que el propósito de los storyboard es obtener el criterio de los usuarios sobre las funcionalidades propuestas para la aplicación en etapas tempranas del ciclo de vida, mucho antes de que sean comprometidos a código.(LEFFINGWELL y WIDRIG, 2003)
- **Escenarios de operación:** es posible llevar a cabo de manera informal la obtención de requisitos basada en escenarios cuando los ingenieros de requisitos trabajan con los clientes en la identificación de escenarios y en la captura de detalles de dichos escenarios. Los escenarios se pueden redactar como texto, complementados por diagramas, fotografías de las pantallas, etcétera. De forma alternativa, se puede adoptar un enfoque más estructurado, como los escenarios de eventos o de los casos de uso. (SOMMERVILLE, 2004)
- **Prototipos:** un prototipo es un borrador de un producto potencial o de una parte del mismo. Es una simulación de los requisitos.(INTECO, 2008) La elaboración de prototipos de un Sistema de información (SI) es una técnica valiosa para la recopilación rápida de información específica acerca de los requisitos de información de los usuarios. Los prototipos efectivos deben hacerse tempranamente en el ciclo de vida del desarrollo de

sistemas, durante la fase de determinación de requisitos. Sin embargo, la elaboración de prototipos es una técnica compleja que requiere el conocimiento completo del ciclo de vida del desarrollo de sistemas antes de que pueda ser lograda satisfactoriamente.(...) Algunos autores recomiendan usar la elaboración de prototipos como parte del ciclo de vida de desarrollo de SI. En esta visión, la elaboración de prototipos es considerada como un método especializado adicional para la determinación de los requisitos de información del usuario.(MENDOZA, 2012)

- **Extracción de fuentes como documentos del negocio, estándares o pautas establecidas en la organización cliente:** otro de los métodos tradicionales para determinar los requisitos es el estudio de los documentos del negocio para descubrir los aspectos importantes, las políticas, las reglas y las directrices, así como ejemplos del uso de la información en la organización. (DURAN y BERNÁRDEZ, 2000; MENDOZA, 2012)
- **JAD :** esta técnica fue introducida por IBM a finales de los años '70, se caracteriza por: realizar reuniones donde participan usuarios claves, gerentes y analistas de sistemas involucrados en el análisis del sistema actual (pretende ser un proceso efectivo), permite determinar conflictos existentes y entender cuáles serán difíciles de resolver, consta de varias sesiones (días y/o semanas) e implica costos de recursos humanos involucrados y de los lugares reservados para realizar los JAD fuera de la organización.(MENDOZA, 2012)
En comparación con las entrevistas individuales, presenta las siguientes ventajas: ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado, todo el grupo, incluyendo los clientes y los futuros usuarios, revisan la documentación generada, no sólo los ingenieros de requisitos, e implica más a los clientes y usuarios en el desarrollo. (DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN, 2010)
- **Técnica del Aprendiz:** esta técnica se basa en la idea del maestro y en la observación del trabajo real. El ingeniero de software representa al aprendiz y el usuario/cliente cumple el rol de maestro. El aprendiz se sienta con el maestro a aprender por medio de la observación, haciendo preguntas y también realizando algún trabajo bajo la supervisión del maestro. Generalmente para el cliente/usuario es difícil explicar completamente su trabajo. En ocasiones tiende a omitir elementos que consideran triviales. La técnica del aprendiz es muy útil para mitigar este problema. Es también una técnica apropiada para un proyecto donde el problema no es estructurado, o sea, en casos en los que se necesita obtener conocimiento tácito de los clientes/usuarios. Tiene como limitante que su implementación requiere de mucho tiempo. (AURUM y WOHLIN, 2005; PÉREZ, KARINA, 2007; PÉREZ, YOISY, 2009; ROBERTSON y ROBERTSON, 2006)

Actividad 3. Entender, analizar y refinar los requisitos

Como parte del análisis de los requisitos deben identificarse y evaluarse los riesgos relacionados con estos. Existen varios métodos y herramientas para la identificación de riesgos. La decisión respecto a qué método utilizar se basa en las características particulares del equipo de proyecto. Entre las técnicas más popularmente aceptadas se encuentran: tormenta de ideas, encuestas, entrevistas, grupos de trabajo, conocimiento por experiencia o documentado, taxonomías de riesgos, plantillas de ingeniería y de camino crítico, entre otras. (MANIASI, 2005; VALLADARES, 2010)

Las tres primeras técnicas fueron explicadas anteriormente, el resto se describe a continuación, acorde a (MANIASI, 2005):

- **Grupos de trabajo:** son un mecanismo muy importante para el análisis de un área o tema particular al realizar un proceso de discusión que posibilita el descubrimiento de riesgos que podrían no ser obvios para el grupo encargado específicamente de la identificación. Generalmente, estos grupos de trabajo están compuestos por personas especializadas en alguna determinada temática relacionada con el proyecto.
- **Conocimiento por experiencia o documentado:** el conocimiento por experiencia es una colección de información que una persona ha obtenido mediante su experiencia. El conocimiento documentado es una colección de información que ha sido documentada respecto a un determinado tema (en este caso particular, riesgos en una determinada área de interés). Es importante considerar que debe validarse la relevancia y aplicabilidad de los conocimientos del experto y de la documentación pertinente al hacer uso de estas técnicas.
- **Taxonomías de riesgos:** las taxonomías son listados de riesgos que han sido encontrados en programas, proyectos o situaciones similares a las que se intenta analizar. Es importante considerar que debe validarse la relevancia y aplicabilidad de la información al hacer uso de esta técnica. Las taxonomías se basan en el uso de preguntas que hacen referencia a situaciones o eventos sobre un área particular de un proyecto o programa y que pueden derivar en una serie de riesgos para el mismo; generalmente estas preguntas se encuentran agrupadas por áreas temáticas (por ejemplo rendimientos, costos, calendario).
- **Plantillas de ingeniería:** constituyen un conjunto de diagramas de flujo respecto a varios aspectos del proceso de desarrollo y se usan como una guía contra la cual validar, empleando una visión de arriba hacia abajo, las actividades a realizar durante un proyecto. Debido a que poseen información de entradas, salidas y acciones esperadas para cada

parte del proceso como así también de los factores involucrados en las mismas, suelen ofrecer un importante marco de referencia. Su principal inconveniente lo constituye el hecho de que estas plantillas pueden estar incompletas por lo cual se recomienda hacer una exhaustiva revisión de las mismas en caso de optarse por su utilización.

- **Plantillas de camino crítico:** estas plantillas reciben también el nombre de plantillas Willoughby y se caracterizan por mostrar áreas de riesgos y proveen un mecanismo de reducción de los mismos para cada una de las etapas de un ciclo de vida de desarrollo de software por el hecho de posibilitar fácilmente su identificación. Debido a que se realizan constantemente nuevas publicaciones actualizadas de estas plantillas, es importante considerar el contar con información reciente al emplear esta técnica.

Wieggers recomienda en (WIEGERS, KARL E., 2003) el almacenamiento de los riesgos en forma de tabla en una hoja de cálculo debido a que facilita ordenar la lista de riesgos en lugar de añadirlos al plan del proyecto o a la especificación de requisitos de software. Recomienda tenerlos en un documento independiente para que sea más fácil de actualizar a lo largo de la duración del proyecto. Por estas razones se recomienda incluir la lista de riesgos asociados a los requisitos al Plan de Riesgos del Proyecto.

Actividad 4. Especificar los requisitos del cliente

La especificación de los requisitos se realiza utilizando un lenguaje natural, se recomienda tener en cuenta el documento normativo Prácticas Recomendadas para la Especificación de Requisitos de Software de la IEEE que establece un conjunto de criterios para desarrollar una buena especificación: naturaleza de la ERS, ambiente, características de una buena ERS, preparación en conjunto, evolución, prototipado, entre otros. (IEEE, 2009)

La plantilla Especificación de Requisitos de Software propuesta en el proceso incluye estos criterios.

Como parte de la especificación de los requisitos se deben definir también las prioridades de cada requisito. La priorización resuelve problemas fundamentales como la competencia que existe entre los plazos de entrega y recursos limitados contra la implementación de las funcionalidades más importantes para el cliente en el tiempo en que se necesitan.

Para llevar a cabo la priorización pueden tenerse en cuenta técnicas como: el Proceso de jerarquía analítica (AHP del inglés Analytical Hierarchy Process), el método de los 100 puntos, asignación numérica (agrupamiento) y priorización basada en valor, costo y riesgo. Las técnicas de priorización pueden ser combinadas para la obtención de mejores resultados.

- **Proceso de Jerarquía Analítica (AHP):** es un método (...) que requiere comparar todos los pares de requisitos para determinar cuál de los 2 tiene una mayor prioridad. Por un lado AHP es un método exigente debido al incremento dramático del número de pares de comparaciones cuando la cantidad de requisitos crece. Por otra parte es muy confiable ya que la gran cantidad de redundancia en las comparaciones por pares hace el proceso bastante insensible a errores. Otra ventaja es que las prioridades resultantes son relativas y se basan en una escala de proporción, que permite evaluaciones útiles de los requisitos.(AAQIB *et al.*, 2009; L.SAATY, 1980)
- **Asignación numérica:** esta técnica consiste en clasificar inicialmente los requisitos en obligatorios, deseables y no esenciales, posteriormente cada interesado asigna un calificación entre 1 y 5 y se extrae el promedio, los requisitos con mayor promedio serán los de mayor prioridad para ser atendidos, si hay varios con igual promedio, el cliente decidirá cuál de ellos debe ser evacuado primero. (HADAD *et al.*, 2009)
- **Método de los 100 puntos:** este método se basa en un sistema de votación donde a cada interesado se le dan 100 puntos que pueden asignar entre los requisitos del modo que deseen. Los requisitos que son más importantes para los interesados son lo que reciben más puntos. Finalmente se priorizan acorde a la cantidad total de puntos que les fue asignada.(MULLA y GIRASE, 2012)
- **Priorización basada en valor, costo y riesgo:** es un método definido por Karl Wieggers, posee un enfoque analítico semi-cuantitativo para priorizar los requisitos. Tiene como apoyo un modelo de hoja de cálculo (Microsoft Excel) simple que ayuda a estimar las prioridades relativas para un conjunto de características del producto.

Este método distribuye un conjunto de prioridades estimadas de forma continua, en lugar de agruparlas en tan sólo unos pocos niveles. Este modelo se puede aplicar de la misma forma para establecer prioridades tanto entre casos de uso individuales que requisitos funcionales. Toma prestado el concepto de QFD (Quality Function Deployment) del valor para el cliente dependiendo tanto del beneficio que brinda al cliente el que una característica específica del producto está presente y la penalización a pagar si no lo está. Debe aplicarse este esquema de prioridades sólo a las funcionalidades negociables, aquellas que no están en la categoría de máxima prioridad. Por ejemplo, no deben incluirse en este análisis de prioridades los elementos que implementan funciones empresariales básicas del producto, que se ven como diferenciadores claves del producto, o que son necesarios para el cumplimiento de regulaciones gubernamentales. Esas características van a incluirse de todos modos. Una vez que haya identificado las características que deben ser absolutamente incluidas en el

producto, calcule las prioridades relativas de los elementos restantes usando este modelo.(WIEGERS, KARL E, 1999)

Actividad 6. Revisión, validación y aceptación de requisitos del cliente

Para realizar la validación de los requisitos se pueden utilizar técnicas como: los prototipos (explicada anteriormente), aplicación de listas de chequeo, definición de criterios de aceptación, entre otros.

- **Listas de chequeo:** consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). (ESCALONA y KOCH, 2002)
- **Criterios de aceptación:** Los criterios y normas enumeradas deben convencer a su cliente que los entregables producidos serán lo suficientemente robustos para garantizar que sus requisitos están plenamente satisfechos. Aunque los criterios que figuran en primer lugar en las listas de criterios de aceptación casi siempre se enfocan a la calidad esperada, se deberá detallar adecuadamente otros criterios de aceptación como: (ANÓN, 2009)
 - La calidad de los nuevos procesos emitidos por el Jefe de Proyecto.
 - La puntualidad de los productos y procesos de entrega.
 - La capacidad del proyecto para producir los entregables dentro del presupuesto.

Actividad 7. Crear sistema de trazabilidad con requisitos y clientes

Para llevar la trazabilidad de los requisitos se recomienda la utilización de una herramienta que automatice este proceso y permita visualizar las relaciones ya sea a través de la matriz de trazabilidad o de su representación en un árbol.

- **Matriz de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. (DURAN y BERNÁRDEZ, 2000)

Actividad 8. Especificación de los requisitos técnicos

Para especificar los requisitos técnicos, pueden aplicarse diferentes técnicas: Casos de Uso para modelar elementos del sistema, Diagramas de flujo, Modelo Entidad Relación, entre otros. (PÉREZ, KARINA, 2007; PÉREZ, YOISY, 2009)

- **Diagramas de flujo:** Los Diagramas de Flujo, o su nueva forma de expresión, los diagramas de actividades elaborados utilizando el lenguaje de modelado UML, tienen la ventaja de una familiaridad razonable. Incluso personas sin entrenamiento en informática saben qué es un diagrama de flujo. La notación UML proporciona una representación visual

que resulta más fácil de entender que otras representaciones como el pseudocódigo. (PÉREZ, KARINA, 2007; PÉREZ, YOISY, 2009)

- **Modelo Entidad Relación:** El Diagrama Entidad Relación (DER) fue propuesto originalmente por Peter Chen para el diseño de sistemas de Bases de Datos relacionales y ha sido ampliado por otros. Se identifican un conjunto de componentes primarios para el DER: objetos de datos, atributos, relaciones y varios indicadores tipo. El propósito primario del DER es representar objetos de datos y sus relaciones. (PRESSMAN, ROGER, 2006)
Si un conjunto de requisitos está asociado con la descripción de la estructura y la relación entre datos dentro del sistema, es frecuentemente conveniente representar esta información en un diagrama Entidad-Relación (DER), utilizando la notación establecida para las técnicas de análisis estructurado. Los DER proporcionan una vista de alto nivel de la arquitectura de los datos. Son específicamente útiles en aplicaciones donde los datos y las relaciones que los gobiernan son complejos. A pesar de ser los DER una competente técnica de modelación, tiene la desventaja de ser difíciles de leer y entender por personas sin preparación técnica. (PÉREZ, KARINA, 2007; PÉREZ, YOISY, 2009)

Resulta muy difícil establecer criterios para seleccionar técnicas apropiadas. Entre estos criterios pueden ser considerados la facilidad de aprendizaje y de uso, la escalabilidad, el costo, la calidad y completitud de los resultados y el tiempo requerido para aplicar las técnicas. Así podemos decir que el uso de lenguajes naturales produce resultados más imprecisos que una descripción con casos de uso y ésta a su vez es más imprecisa que requisitos descritos formalmente. Los casos de uso son apropiados tanto para pequeños como grandes sistemas, mientras que el uso de plantillas resulta menos apto para grandes sistemas. Así mismo, técnicas como JAD son más difíciles de usar y consumen mucho más tiempo que entrevistas, permitiendo en cambio obtener resultados de mayor calidad. (ESCALONA y KOCH, 2002)

3.2 Sistema de Indicadores

Establecer un sistema de mediciones es esencial para toda empresa desarrolladora de software, le permite conocer cuán correctos se están llevando a cabo los procesos en ejecución, y a partir de datos históricos mejorar los resultados obtenidos y posteriormente, con la información suficiente, predecir comportamientos futuros.

Como se ha evidenciado, dentro de los elementos principales en el proceso de ingeniería de requisitos se encuentran los relacionados con el control de los cambios, las revisiones en busca de inconsistencias, los riesgos y el esfuerzo necesario para implementarlo.

La medición del esfuerzo dota a la gerencia de una herramienta para conocer el comportamiento de la implementación de los procesos; cómo se están adaptando los proyectos, si el esfuerzo está aumentando o disminuyendo. Siguiendo esta idea, puede compararse cuánto esfuerzo está costando el proceso contra la utilidad que está dando, de manera que puedan atacarse los puntos débiles donde se descompensa esta comparación. Basándose en estos elementos y en el criterio de expertos que han trabajado el tema de los requisitos, la autora definió el sistema de indicadores. Este sistema de indicadores sirve de base para el monitoreo y control de los requisitos del proyecto en los reportes a la gerencia. A continuación se describen brevemente cada uno de los 8 indicadores. Su descripción detallada se muestra en el Anexo 11.

1. Esfuerzo dedicado al proceso base de ingeniería de requisitos.

El objetivo de este indicador es determinar el esfuerzo necesario para ejecutar el PB de IR con el fin de predecir esta variable en otros proyectos.

Formulación matemática:

$$\text{Esfuerzo dedicado al proceso de requisitos} = \frac{\text{Cantidad de horas dedicadas al proceso}}{\text{Cantidad de personas trabajando en el proceso}}$$

2. Utilidad del proceso base de ingeniería de requisitos.

El objetivo de este indicador es determinar la utilidad del PB de IR. Para esto, es necesario como primer paso, la aplicación de una encuesta que tiene como objetivo de determinar las medidas bases.

Formulación matemática:

$$T = P + N$$

Donde:

T: total de respuesta, P: Total de respuestas positivas y N: Total de respuestas negativas.

$$P_p = \frac{P * 100}{T}$$

Donde Pp: Valor en % de las respuestas positivas de cada categoría de la encuesta.

$$P_n = \frac{N * 100}{T}$$

Donde: Pn: Valor en % de las respuestas negativas de cada categoría de la encuesta.

$$P_{tp} = \frac{\sum P * 100}{\sum T}$$

Donde Ptp: Valor en % del total positivo de utilidad del PB de IR.

$$Ptn = \frac{\sum N * 100}{\sum T}$$

Donde Ptn: Valor en % del total negativo de utilidad del PB de IR.

3. Volatilidad de los requisitos.

El objetivo de este indicador es cuantificar los cambios en los requisitos del software durante el desarrollo para determinar su impacto sobre la planificación del proyecto.

Formulación matemática:

$$T = \sum_{\text{requisito } 1}^{\text{requisito } n} \text{Número de cambios sufridos por el requisito}$$

Donde T: Total de cambios encontrados

$$\% = \frac{\text{Cantidad de cambios que sufrió el requisito} * 100}{T}$$

Donde %: representa la cantidad de cambios que sufrió ese requisito del total de cambios encontrados.

4. Cantidad de elementos inconsistentes por revisiones de inconsistencia.

El objetivo de este indicador es determinar el número elementos inconsistentes por cada revisión de inconsistencia.

Formulación matemática: Número de elementos inconsistentes por revisión de inconsistencia.

5. Tiempo de resolución de inconsistencias.

El objetivo de este indicador es determinar el tiempo de resolución de inconsistencias en cada fase del proyecto.

Formulación matemática: Tiempo en horas empleado para resolver inconsistencias en cada una de las fases.

6. Causas de las inconsistencias.

El objetivo de este indicador es analizar las causas de las inconsistencias entre los requisitos, sus productos de trabajo asociados y los planes del proyecto.

Formulación matemática:

Frecuencia Media = Frecuencia Absoluta / Cantidad de Revisiones.

Total de Frecuencia Media = \sum Frecuencia Media.

Frecuencia Relativa = Frecuencia Media / Total de Frecuencia Media.

Frecuencia Acumulada = \sum Frecuencia Relativa.

7. Acciones correctivas tomadas para resolver las inconsistencias (y además si fueron exitosas o no).

El objetivo de este indicador es monitorear las acciones correctivas tomadas para resolver las inconsistencias (si fueron exitosas o no) durante el desarrollo del proyecto para determinar su impacto sobre la planificación del proyecto.

Formulación matemática:

Número de inconsistencias encontradas.

Número de acciones correctivas aplicadas.

8. Cantidad de riesgos convertidos en problemas.

El objetivo de este indicador es analizar la cantidad de riesgos, relacionados con los requisitos, convertidos en problemas con el fin de resolverlos.

Formulación matemática:

Probabilidad del riesgo

Cuando la probabilidad es 1, el riesgo se materializa, entonces se convierte en problema.

3.3 Herramientas

3.3.1 Herramienta para el cálculo de los indicadores

Recolectar, calcular y analizar indicadores puede convertirse en una tarea extremadamente engorrosa y alta consumidora de esfuerzo.

En aras de disminuir este esfuerzo y lograr cierta automatización, la autora elaboró una herramienta utilizando una hoja de cálculo (Microsoft Excel) que permite no solo calcular automáticamente cada indicador sino también representarlos gráficamente, lo que facilita su interpretación. Ver Anexo 12. Esta herramienta se incluye también en el CD adjunto.

3.3.2 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) proporcionan al ingeniero la posibilidad de automatizar actividades manuales y mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y de diseño asistidos por computadora que utilizan los ingenieros de otras disciplinas, las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto. (PRESSMAN, ROGER, 2006)

De manera general son utilizadas para ayudar al equipo de trabajo a llevar a cabo las actividades del proceso de desarrollo de software, dentro de estas, las de requisitos. Varios autores han realizado propuestas de herramientas para apoyar las actividades de administración y desarrollo de requisitos. Tal es el caso de (ALARCON y SANDOVAL, 2008) que realizan una investigación de las herramientas CASE que apoyan las tareas de la ingeniería de requisitos, entre las analizadas se encuentran: IRqA, RETO, CONTROLA,

OSRMT, RAMBUTAN, JEREMIA, entre otras que referencian de INCOSE (International Council on Systems Engineering).

En (CALLEJAS et al., 2010) se propone una herramienta propia: Heler, y se realiza un estudio comparativo entre cuatro herramientas comerciales: IRqA, RequisitePro, DOORS, CaliberRM y tres herramientas académicas: REM, RETO y Controla, que cumplen con la mayoría de las actividades de la ingeniería de requisitos.

En otras investigaciones como en (PÉREZ, KARINA, 2007) se proponen como herramientas para la especificación de requisitos: ArgoUML, Visual Paradigm para UML y Dia; y para la administración de requisitos propone un conjunto que puede ser utilizado en GNU/Linux: CARE 3.2, Cradle 5.2, Focal Point, IRqA 3.3, PACE, SoftREQ, UGS Teamcenter 2005, Banyan2.2, Contour y OSRMTv1_3.

En el caso de (PÉREZ, YOISY, 2009) divide su propuesta en herramientas para el desarrollo de los requisitos, entre las que menciona: IRqA, RequisitePro, DOORS, CaliberRM y para la administración de requisitos: Rational Rose, Enterprise Architect, ArgoUML, Visual Paradigm para UML.

En el artículo Evaluación de herramientas de gestión de requisitos de (ROSIQUE et al., 2010) plantea como las cuatro herramientas más importantes: RequisitePro, IRqA, CaliberRM y DOORS.

A partir de las herramientas revisadas se proponen las siguientes:

- **IRqA:** IRqA es una herramienta de administración y definición de requisitos, diseñada específicamente para soportar visualmente todas las actividades relacionadas con los requisitos. Ofrece flexibilidad para apoyar visualmente múltiples procesos y metodologías en Ingeniería de Software y Sistemas. Provee un menor tiempo de implementación y orientación al usuario. Posee una arquitectura de gran alcance, y utiliza estándares. Asegura la apertura, integración, requisitos de comunicación y soporte a equipos distribuidos. (VISURE, 2010)
- **RequisitePro:** IBM Rational RequisitePro es una herramienta de gestión de requisitos. Ayuda a los equipos de proyectos a gestionar sus necesidades, escribir buenos casos de uso, mejorar la trazabilidad, fortalecer la colaboración, reducir el trabajo redundante del proyecto y aumentar la calidad.(IBM, 2012a)
- **Enterprise Architect:** Enterprise Architect provee opciones de modelado para el ciclo de vida completo: Negocios y los sistemas de TI, Ingeniería de Software y Sistemas, Desarrollo en tiempo real y embebidos. Con capacidades integradas de gestión de requisitos, Enterprise Architect ayuda a trazar especificaciones de alto nivel a los modelos de

análisis, diseño, implementación, prueba y mantenimiento utilizando UML, SysML, BPMN y otros estándares abiertos. Es un sistema multi-usuario, diseñado para ayudar a los equipos a construir sistemas robustos y de fácil mantenimiento. Provee capacidades de reportes y documentación con una alta calidad.(SPARX SYSTEMS, 2012)

- **CaliberRM:** Es una herramienta que permite visualizar, administrar y entregar requisitos a los agentes del proyecto. Simplifica y reutiliza los requisitos de todos los proyectos, genera escenarios de casos de prueba y se integran con el desarrollo de código y herramientas de prueba para la verificación y validación. Realiza un análisis de impacto, comprender los impactos del cambio y la alineación con los objetivos de negocio a través de las relaciones de trazabilidad. Visualiza y reduce los costos, asegurando la entrega de los más altos requisitos de calidad a sus grupos de interés más rápido a través de guiones gráficos, prototipos y simulaciones.(BORLAND, 2012)
- **OSRMT:** Herramienta libre para la gestión de requisitos, cuyas principales características son: trabaja en arquitectura cliente/servidor, desarrollada bajo Java; la versión 1.3 trae un módulo para manejar la trazabilidad y lo introduce para el control de cambios; así mismo, genera la documentación de los requisitos tratados.(ALARCON y SANDOVAL, 2008)
- **Rational Rose:** IBM Rational Rose Enterprise proporciona un lenguaje de modelado común que permite acelerar la creación de software de calidad. Incluye soporte a UML y es uno de los productos más completos de la familia de Rational Rose. Soporta Análisis, ANSI C + +, Patrones de Rose J y Visual C + +, Enterprise JavaBeans 2.0, ingeniería e ingeniería inversa para las construcciones más comunes de Java 1.5. Ofrece capacidades de análisis y generación de código, configurables con capacidades de sincronización de modelo a código, así como cuenta con una gestión más granular y el uso de modelos con los componentes de los modelos controlables por separado.(IBM, 2012b)
- **ArgoUML:** ArgoUML es la principal herramienta de modelado UML de código abierto. Incluye soporte para todos los diagramas del estándar UML 1.4.Funciona en cualquier plataforma Java y está disponible en diez idiomas. Se distribuye bajo la licencia EPL(Eclipse Public License).(TIGRIS.ORG, 2012)
- **Visual Paradigm para UML (VP-UML):** es una herramienta CASE de diseño UML para ayudar al desarrollo de software. VP-UML soporta los principales estándares de la industria tales como UML, SysML, BPMN, XMI, etc. Ofrece un conjunto completo de herramientas para los equipos de desarrollo de software destinadas a la captura de requisitos, planificación de software, planificación de controles, modelado de clases, modelado de datos, y un largo etc.(VISUAL PARADIGM, 2012)

INCOSE en (INCOSE, 2011) su actualización del reporte de las herramientas de administración de requisitos, lista un total de 34 herramientas para la administración de requisitos entre las que se encuentran la mayoría de las propuestas, aunque se agregaron algunas con las características de código abierto que no se incluyen en dicho reporte.

Conclusiones parciales del capítulo

- Se realiza una propuesta de técnicas y herramientas como apoyo al proyecto en el proceso de selección. Se describen un conjunto de estas, explicando sus características principales, de forma que sea más fácil su selección. Estos elementos predeterminados pretenden contribuir a la disminución de esfuerzo en el proyecto, en concepto de su definición y elaboración, permitiéndole aumentar su productividad.
- El sistema de indicadores propuesto tiene entre sus objetivos controlar y monitorear los requisitos en el proyecto. Así como llevar el registro de datos que pueden servir de históricos para futuras tomas de decisiones y predicciones de comportamiento. Los indicadores que miden elementos del proceso sirven de apoyo para su evaluación y mejora continua.

Capítulo 4: Validación de la solución

4. Introducción

En el presente capítulo se describe la validación de los resultados de la investigación y se explican los métodos utilizados. Para validar el PB de IR y el sistema de indicadores se realizó un cuasi experimento y se aplicaron otras técnicas como la escala de Likert fundamentada en la opinión de los expertos y la técnica de ladov.

4.1 Evaluación del Proceso Base de Ingeniería de Requisitos y el sistema de indicadores mediante valoración de expertos.

Una de las vías utilizadas para validar el proceso y sistema de indicadores es a través de las opiniones de los expertos que fueron analizadas posteriormente utilizando la escala de Likert. Con este método se pretende utilizar la información fiable de los expertos en aras de validar la hipótesis planteada y valorar la utilidad del resultado de la investigación. Para esto se utilizaron dos grupos de expertos, uno para validar el PB de IR y otro para el sistema de indicadores.

4.1.1 Selección de los expertos

En (FEBLES, 2003) a través de la utilización del método presentado por Cyret y March en 1965 se selecciona la cantidad de expertos, obteniendo como resultado de esta selección 7 expertos. Varios investigadores que han utilizado este mismo método coinciden en que 7 es la cantidad adecuada (HERNÁNDEZ, YAILÍN, 2011; PÉREZ, YOISY, 2009). Astigarraga menciona (ASTIGARRAGA) que aunque no hay forma de determinar el número óptimo de expertos, estudios señalan que si bien parece necesario un mínimo de siete expertos habida cuenta que el error disminuye notablemente por cada experto añadido hasta llegar a los siete, no es aconsejable recurrir a más de 30 expertos, pues la mejora en la previsión es muy pequeña y normalmente el incremento en coste y trabajo de investigación no compensa la mejora.

La calidad de los expertos cumple un papel importante en este tipo de método. Seleccionar a los expertos idóneos para cada investigación es decisivo. Uno de los procedimientos más usados para objetivar la correcta selección es el método de autoevaluación, que consiste en la evaluación de las competencias que cada experto realiza de sí mismo acorde a un cuestionario.

A partir de estos datos se calcula el coeficiente de competencia (K) para lo que se tiene en cuenta la autovaloración del experto acerca de su competencia (K_c coeficiente de conocimiento) y las fuentes de argumentación (K_a coeficiente de argumentación) mediante la siguiente fórmula (1):

$$K = \frac{Kc+Ka}{2} \quad (1)$$

El Ka se calcula a partir de una escala predeterminada (Ver Anexo 13) que se incorpora a la encuesta aplicada a los expertos. Las encuestas presentadas a los dos grupos se encuentran en el Anexo 14.

A partir de estos datos se especifica el grado de coeficiente de competencia,

Si $0,8 \leq K < 1,0$ coeficiente de competencia Alto.

Si $0,5 \leq K < 0,8$ coeficiente de competencia Medio.

Si $K < 0,5$ coeficiente de competencia Bajo.

En el Anexo 15 se muestra una tabla con un resumen de los grados y coeficientes de competencia, coeficiente de conocimiento y coeficiente de argumentación de cada grupo de expertos. A partir de estos se evidencia que en los dos grupos (para validar el PB de IR y para validar el sistema de indicadores) el 71.4% posee un grado de coeficiente de competencia alto y un 28.6% un grado medio confirmándose que la mayoría de los expertos seleccionados tienen un grado de coeficiente de competencia alto.

Los expertos seleccionados para valorar el PB de IR poseen amplios conocimientos en el área de ingeniería de requisitos, modelos y normas internacionales y en calidad de software. El 100% es Máster y profesor asistente, además han tenido interacción con la parte productiva por lo que tienen el punto de vista del proyecto. Los encuestados sobre el sistema de indicadores, el 57.2 % es Máster y el 42.8% Ingeniero en Ciencias Informáticas que se tomaron como expertos debido a la experiencia de su trabajo como parte del Grupo de la Medición del Centro de Calidad de la UCI (actual Calisoft) que posteriormente desarrolló el proceso de Medición y Análisis, se encontraba a cargo de la definición, retroalimentación y ajuste del método de estimación entre otras tareas relacionadas. La caracterización de cada experto se encuentra en el Anexo 16.

4.1.2 Escalamiento de Likert

Con el fin de ofrecer un resultado que visualice mejor las valoraciones de los expertos que respondieron el cuestionario final, se aplicó la escala de Likert, donde se otorga una puntuación entre 1 y 5 a cada ítem. Ver Anexo 17.

Luego se calculó un índice porcentual (IP) según la fórmula (2), que integra en un solo valor la aceptación del grupo de evaluadores sobre las características del proceso e indicadores.

$$IP = \frac{5 (\% \text{ de } MA) + 4(\% \text{ de } A) + 3(\% \text{ de } MnA) + 2(\% \text{ de } PA) + 1(\% \text{ de } I)}{5} \quad (2)$$

Donde:

- MA: Muy adecuado (o Muy importante)

- A: Adecuado (o Importante)
- MnA: Medianamente adecuado (o Medianamente importante)
- PA: Poco Adecuado (o Poco Importante)
- I: Inadecuado (Sin importancia)

En la figura 4 se observa que la aceptación de los expertos (IP) en la valoración del PB de IR sobrepasa el 80%.

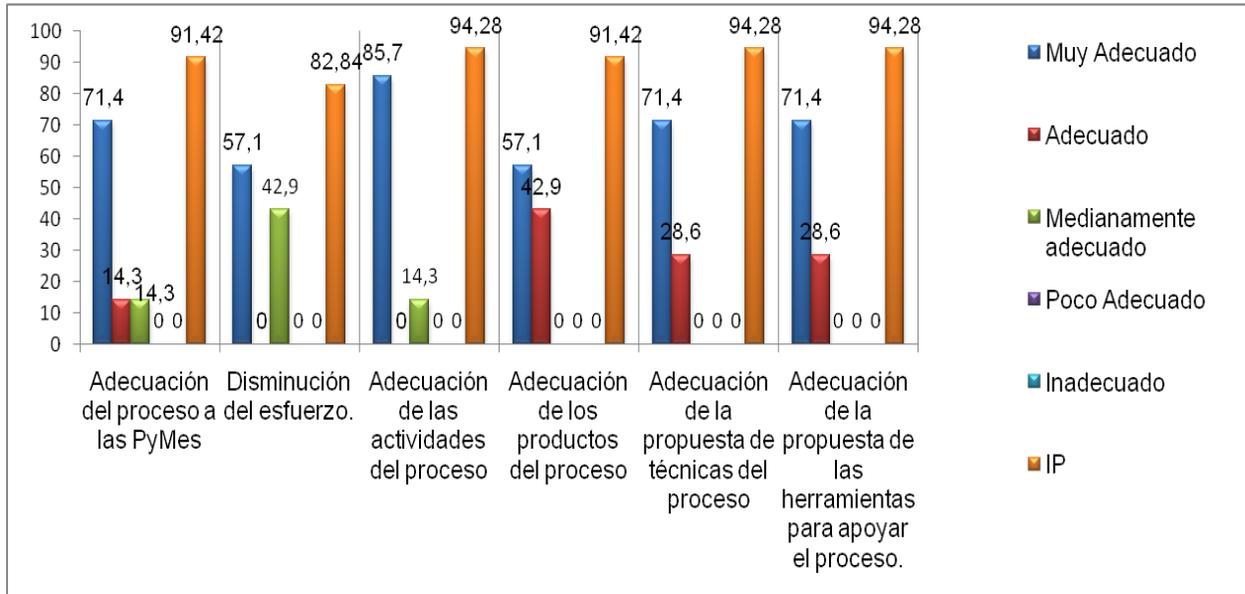


Figura 4. Resultados de la aplicación de la escala de Likert (PB de IR)

El procesamiento realizado a través del escalamiento de Likert evidencia que los componentes del proceso son valorados en un alto por ciento como muy adecuados, a lo que se une la valoración de que su aplicación aminora el esfuerzo en las PYME de desarrollo de software, y al necesitarse menos esfuerzo para las prácticas de IR se aumenta la productividad.

Este mismo análisis se realizó con la valoración de los expertos sobre el sistema de indicadores como puede verse en la figura 5.

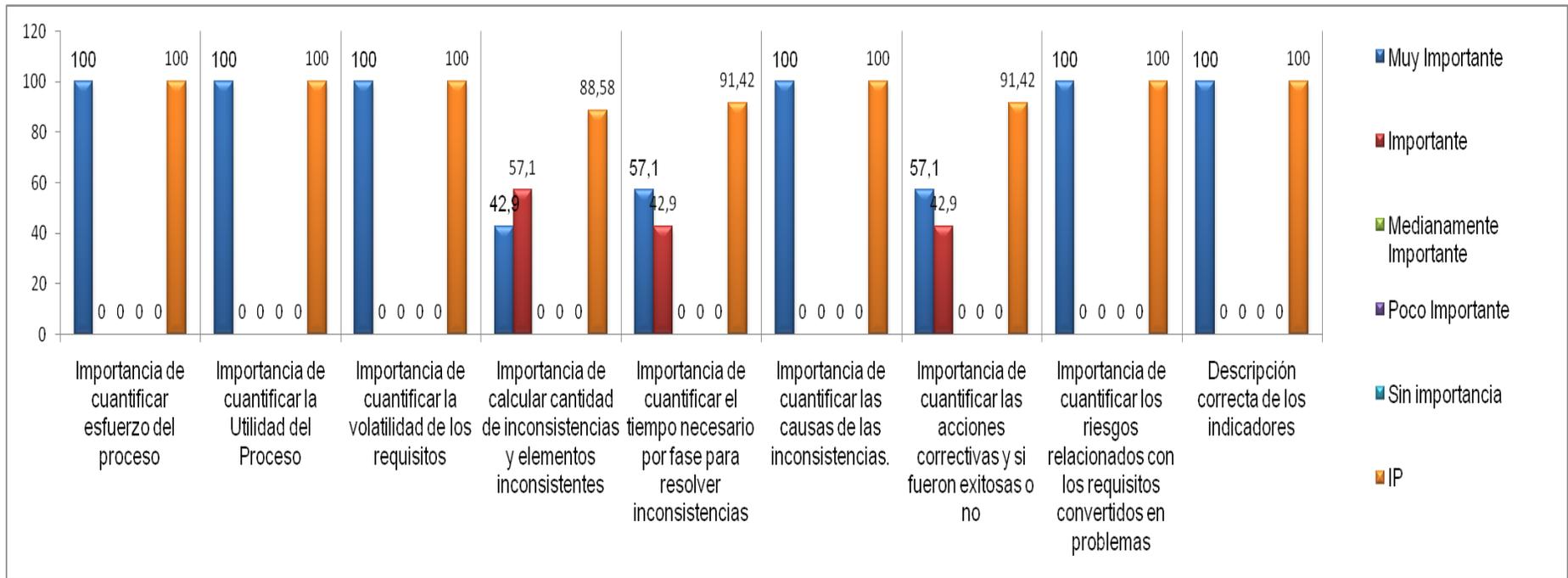


Figura 5. Resultado de la valoración de los expertos sobre el sistema de indicadores

Esta información permite arribar a la conclusión de que los expertos valoraron correctamente el sistema de indicadores propuesto, el 100 % consideró a los indicadores: Esfuerzo dedicado al proceso, Utilidad del PB de IR, Volatilidad de los requisitos, Causas de las inconsistencias y Cantidad de riesgos convertidos en problemas como muy importantes para el proyecto. También coincidieron en que los indicadores han sido descritos correctamente.

Respecto a los indicadores: Cantidad de elementos inconsistentes por revisiones de inconsistencia, Tiempo de resolución de inconsistencia y Acciones correctivas tomadas para resolver las inconsistencias, los expertos opinaron que son muy importantes en un 42.9%, 57.1% y 57.1% respectivamente, el otro porcentaje planteó que eran importantes para el proyecto. En resumen la aceptación de los expertos respecto a los indicadores sobrepasa el 85%.

4.2 Cuasi experimento para validar la solución

4.2.1 Productividad y esfuerzo

Acorde a la definición de la Real Academia de la Lengua Española (RAE), esfuerzo es el empleo de elementos costosos en la consecución de algún fin. (REAL ACADEMIA ESPAÑOLA, 2010) El PMBOK lo define como: la cantidad de unidades laborales necesarias para terminar una actividad del cronograma o un componente de la estructura de desglose del trabajo. Generalmente se expresa como horas, días o semanas de trabajo del personal.(GUÍA DEL PMBOK, 2004) Consecuentemente el esfuerzo se denomina como el empleo de tiempo para realizar el trabajo del proyecto y la unidad de medida es horas/hombres.

Como se explicó en el capítulo 1, la productividad está estrechamente relacionada con el esfuerzo y el tamaño. Aunque numerosas investigaciones han arribado a la conclusión que en esta relación influyen otros factores, si coinciden con el hecho de que producir la misma cantidad con un menor esfuerzo sin afectar la calidad del producto conlleva al aumento de la productividad.

A continuación se presenta el cuasi experimento realizado mediante el pilotaje del proceso e indicadores.

4.2.2 Cuasi experimento para validar el PB de IR

Para adentrarse en la descripción de la implantación del PB de IR en proyectos pilotos, cabe destacar que al esfuerzo real dedicado a la ejecución de las actividades de administración y desarrollo de requisitos se debe agregar el esfuerzo que le tomó a la organización la interpretación e implementación del área de proceso de REQM del modelo CMMI, lo que afecta la productividad a ese nivel (organización).

En el caso de la UCI, el esfuerzo fue de 117h/h (RAMOS *et al.*, 2011) y en el caso de los proyectos pilotos fue de 0h/h debido a que tomaron el PB de IR definido en el presente trabajo, tal y como se propone, pues se ajustó totalmente a sus características y brindaba elementos suficientes (descripción gráfica y textual del proceso, descripción de roles, plantillas y guías para la elaboración de los productos de trabajo) para comenzar su implantación sin necesidad de dedicar esfuerzo a su instanciación.

Como se explicó en el capítulo 1, en la UCI se definió solamente el proceso de REQM por lo que se añadiría el esfuerzo para definir el proceso referente a las actividades de desarrollo de requisitos. Evidenciándose el gasto de esfuerzo a nivel de la organización y la consecuente afectación a la productividad.

El PB de IR se implantó en 2 proyectos pilotos: Sistema para certificaciones (piloto 1) y Sistema para la gestión de consultorías (piloto 2) del Centro Calisoft.

4.2.2.1 Selección de la muestra para comparación de resultados

A partir de los resultados obtenidos durante el cuasi experimento se hizo necesario escoger proyectos que tuvieran procesos definidos y en ejecución, relacionados con los requisitos, además de otros criterios.

Criterios para la selección de la muestra:

1. Proyectos con los procesos de administración y desarrollo de requisitos definidos y en ejecución.
2. Proyectos pequeños en cuanto a cantidad y complejidad de requisitos y cantidad de miembros del proyecto, similares a los utilizados para el pilotaje.
3. Proyectos con un ambiente tecnológico equivalente a los pilotos (software de gestión y utilización de tecnologías web).

A partir de estos criterios son candidatos los proyectos dentro de los Centros de la UCI (tienen el proceso de Administración de Requisitos definido y aplican prácticas de desarrollo de requisitos). Se tuvo en cuenta además que a partir de la utilización del método de estimación post-arquitectura la cantidad y complejidad de requisitos fuese equivalente. De los 14 centros de la UCI se logró entrevistar 12, para un 85.7% del total. De estos 12 centros, 9 no tenían proyectos que cumplieran con las características para la selección (Ver Anexo 18). De los 3 restantes: en el Centro CIDI, 2 proyectos contenían las similitudes necesarias para la comparación y en los centros CESIM, FORTES y DATEC, 1 proyecto en cada uno, en total 5 proyectos. En el Anexo 19 se resumen las similitudes para establecer la comparación.

4.2.2.2 Comparación de Resultados

De los pilotos y los proyectos de la muestra se tomaron la cantidad de requisitos a implementar y el esfuerzo real dedicado a las tareas de IR y se dividieron para obtener la productividad. Ver tabla 12. Se calculó además el por ciento en que fueron más productivos los pilotos (tomando el valor del piloto menos productivo para mantener un enfoque pesimista) respecto a cada proyecto. Concluyéndose que los pilotos que utilizaron el PB de IR fueron más productivos en las actividades de IR que los proyectos de la muestra, que utilizaron el proceso de REQM de la UCI y aplicaron prácticas de desarrollo de requisitos normadas por lineamientos de calidad definidos.

Tabla 1. Análisis y comparación de resultados

	Cantidad de requisitos a implementar	Esfuerzo Real de las tareas de administración y desarrollo de Requisitos hasta la fase de Análisis y Diseño (H/H)	Productividad	% relativo de aumento de productividad
Proyecto CESIM	25	1772	0,01	88,2
Proyecto Datec	29	284	0,10	14,9
Proyecto 1 CIDI	22	550	0,04	66,7
Proyecto 2 CIDI	29	255	0,11	5,2
Proyecto FORTES	34	964	0,04	70,6
Piloto 1	18	150	0,12	
Piloto 2	26	144,2	0,18	
Promedio de aumento de la productividad (%)				49,1

4.2.3 Cuasi experimento para validar el sistema de indicadores

4.2.3.1 Cuasi experimento a través de la aplicación del sistema de indicadores en los proyectos pilotos

El sistema de indicadores se utilizó en los proyectos pilotos. De los 8 indicadores se aplicaron 7, no pudo aplicarse el de volatilidad de los requisitos debido a que no hubo solicitudes de cambios durante el período.

Los resultados obtenidos de la aplicación de los indicadores se muestran a continuación.

Para el Piloto 1:

Esfuerzo

Los resultados de la aplicación del indicador se observan en la figura 6.

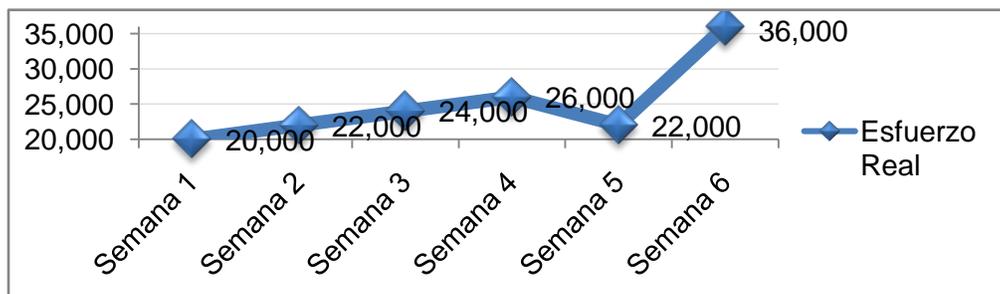


Figura 6. Indicador Esfuerzo dedicado al proceso del piloto 1

El esfuerzo se mantuvo relativamente constante hasta la última semana donde se gastó más del dedicado hasta ese momento debido a que hubo que redefinir algunos requisitos. Estos aún no son números alarmantes para el proyecto, teniendo en cuenta que en estas primeras fases se consume la mayor cantidad de esfuerzo relacionado con los requisitos, es donde se define

lo que se construirá posteriormente. Estos números por sí solos no aportan mucha información, necesitan ser comparados con el esfuerzo estimado hasta la fecha y calcular la desviación respecto a lo planificado.

Para la gerencia sin embargo le da una idea de cuánto le está costando al proyecto seguir los pasos predeterminados por el proceso y comparándolos con otros proyectos y con la estimación general del mismo, realizar una revisión en caso de que esté costando más de lo esperado.

Este indicador puede contrastarse con el de utilidad que muestra a la gerencia cuan útil, acorde al criterio de los proyectos que lo utilizan, está siendo el proceso. En la figura 7 se observa el indicador utilidad.

Utilidad del Proceso Base de Ingeniería de Requisitos.

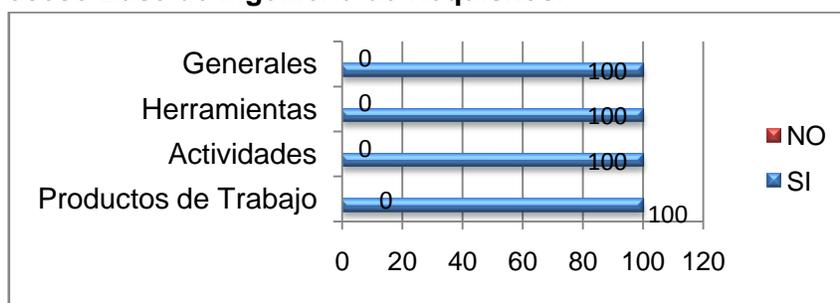


Figura 7. Indicador Utilidad del Piloto 1

El piloto 1 consideró que los productos de trabajos, actividades y herramientas del proceso le son de completa utilidad. Plantearon además como lección aprendida que la priorización de los requisitos les permitió organizar mejor el trabajo y enfocarse en las prioridades del cliente. También opinaron que las revisiones de inconsistencias les permitieron detectar errores a tiempo y aumentar la calidad de los requisitos.

Cantidad de elementos inconsistentes por revisiones de inconsistencia

Este indicador evidenció la cantidad de elementos inconsistentes encontrados en la revisión de inconsistencias que se realizó al culminar la fase de requisitos, ver figura 8.

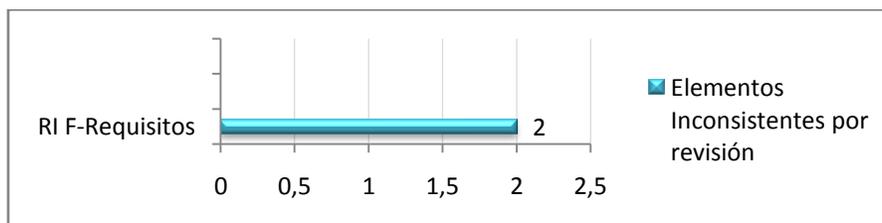


Figura 8. Indicador cantidad de elementos inconsistentes por revisiones de inconsistencia en el Piloto 1

A partir de este gráfico se puede observar que existían 2 elementos inconsistentes al finalizar la fase de Requisitos. El cálculo de este indicador permite valorar al proyecto la calidad con que se está realizando el trabajo y la eficiencia de las revisiones de inconsistencias. En este caso hubo pocos elementos inconsistentes.

Tiempo de resolución de inconsistencias

El indicador Tiempo de resolución de inconsistencias del proyecto puede verse en la figura 9.

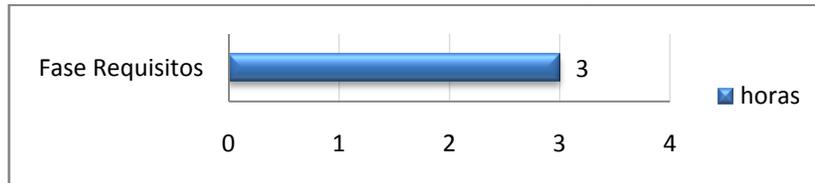


Figura 9. Indicador Tiempo de resolución de inconsistencias del Piloto 1

Acorde al proceso, las inconsistencias deben ser detectadas, asignadas a un responsable para su resolución y monitoreadas hasta su cierre. El tiempo que toma la resolución de las inconsistencias es de importancia para el proyecto debido a que le permite registrar datos que sirvan para futuras estimaciones relacionadas con inconsistencias similares. En la fase de requisitos el proyecto gastó 3 horas en la resolución de la inconsistencia encontrada, un tiempo aceptable en relación a la cantidad y complejidad de las inconsistencias.

Causas de las inconsistencias

Las inconsistencias tienen una fuente que las origina y debe tenerse en cuenta para este indicador. Ver figura 10.

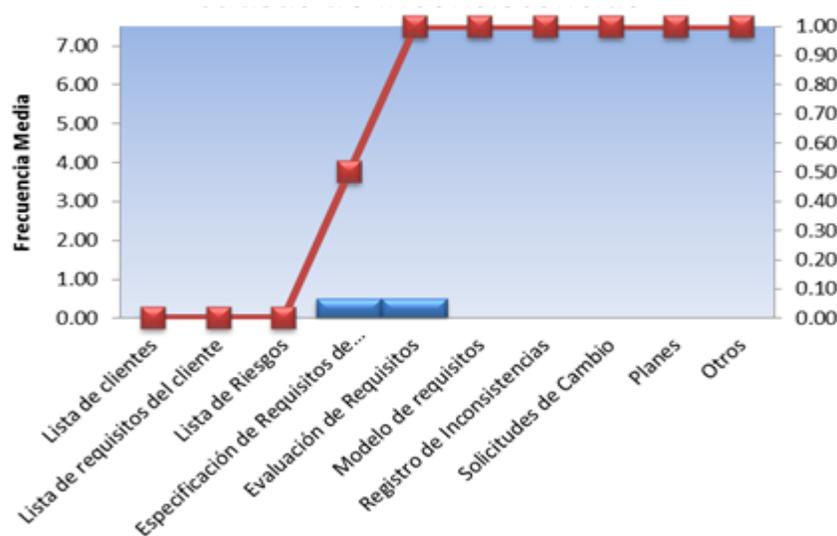


Figura 10. Indicador Causas de las inconsistencias del Piloto 1

Los productos de trabajaron que fueron la causa de las inconsistencias fueron la Especificación de Requisitos de Software y el documento Evaluación de Requisitos (inconsistencia: no se

había calculado la complejidad de algunos requisitos). Esto alerta al proyecto a tener un mayor cuidado en próximas iteraciones o nuevas versiones con estos productos de trabajo.

Acciones correctivas tomadas para resolver las inconsistencias

Se deben tomar acciones correctivas para resolver las inconsistencias. Cuando no son efectivas, se dice que no tuvieron éxito y entonces debe tomarse otra acción correctiva hasta encontrar la adecuada. Es importante conocer cuáles acciones correctivas fueron exitosas y cuáles no. En la tabla 2 se muestra la acción correctiva tomada ante la inconsistencia encontrada y si fue exitosa o no.

Tabla 2. Acciones correctivas tomadas para resolver las inconsistencias

Inconsistencias	Acción correctiva	Exitosa
Inconsistencia 1. En el documento los requisitos del 8 al 24 no tienen calculada la complejidad.	Calcular la complejidad de los requisitos que faltan	Sí

En la figura 11 se muestra la cantidad de acciones correctivas exitosas y fallidas que tuvo el proyecto respecto a la resolución de las inconsistencias. En este caso fue una sola y resolvió la inconsistencia.

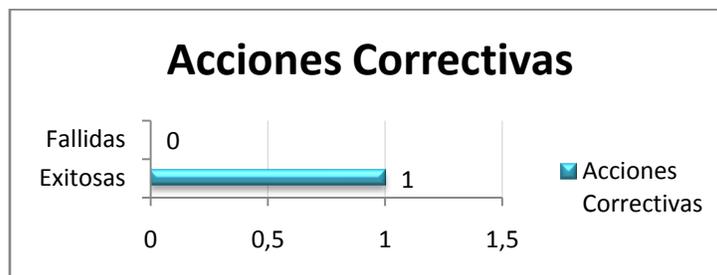


Figura 11. Indicador Acciones correctivas tomadas para resolver las inconsistencias del Piloto 1

Cantidad de riesgos convertidos en problemas

Es vital conocer los posibles problemas que vienen aparejados a la realización de los requisitos. De ahí la importancia de identificar y evaluar los riesgos asociados con estos y controlarlos para que no vayan a convertirse en problemas. En ocasiones esto es inevitable, entonces debe conocerse cuál riesgo se convirtió en un problema y bajo qué condiciones, de manera que esta información sea utilizada tanto por el proyecto para nuevos riesgos como por la gerencia para nuevos proyectos. En la figura 12 se muestran los riesgos del piloto 1 asociados a los requisitos.

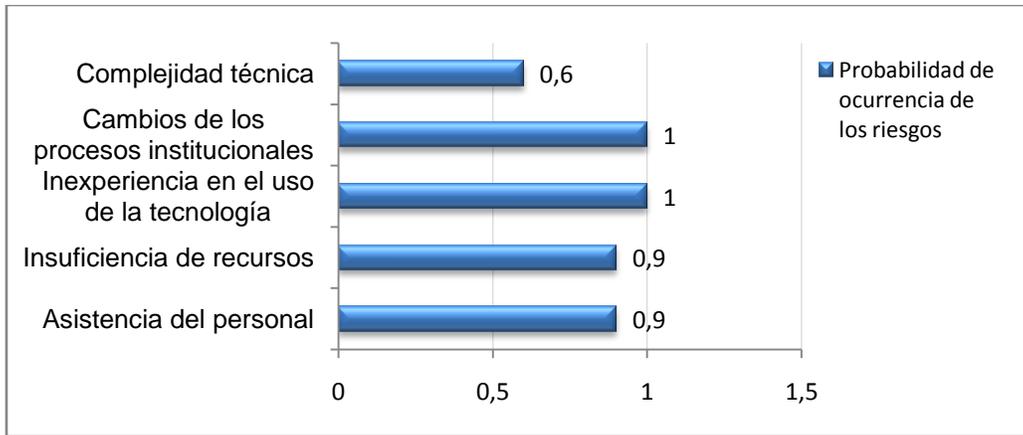


Figura 12. Indicador Cantidad de riesgos convertidos en problemas del piloto 1

Durante la definición de los requisitos se realizaron cambios en los procesos de la organización, lo que ocasionó que se tuvieron que redefinir algunos requisitos y aumentara el esfuerzo del proyecto en re trabajo. Los nuevos miembros asignados al proyecto no poseían la experiencia y conocimientos técnicos necesarios para asumir eficientemente las tareas, hecho que trajo como consecuencia que no se avanzara al ritmo planificado.

4.3 Evaluación de la satisfacción de los usuarios del Proceso y Sistema de Indicadores

La técnica de V.A. ladov en su versión original fue creada por N. V. Kuzmina para el estudio de la satisfacción por la profesión en carreras pedagógicas. Posteriormente ha sido utilizada por diferentes investigadores para evaluar la satisfacción de determinados sujetos respecto a un elemento específico. En la presente investigación se aplica para evaluar la satisfacción de los proyectos pilotos (usuarios) respecto al sistema de indicadores y el PB de IR.

Para el empleo de la técnica se utilizó una encuesta que contenía las preguntas referentes al sistema de indicadores y al proceso. Ver Anexo 20.

La técnica de ladov se fundamenta en las relaciones que se establecen entre tres preguntas cerradas y dos abiertas. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" (ver tablas 3 y 4) e indica la posición de cada sujeto en la escala de satisfacción.

Tabla 3. Cuadro lógico de ladov (modificado por López, 1993) para el PB de IR

	1. ¿Considera el Proceso Base de Ingeniería de Requisitos complejo y difícil de entender?		
	No	No sé	Sí
3. ¿El Proceso de Requisitos utilizado es	2. ¿Si Ud. fuera a realizar otro proyecto utilizaría el Proceso propuesto para llevar a cabo el desarrollo y administración de los		

de su agrado?	requisitos?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Tabla 4. Cuadro lógico de ladov (modificado por López, 1993) para el sistema de indicadores

	1. ¿Considera que los indicadores aplicados carecen de utilidad?								
	No			No sé			Sí		
3. ¿La información brindada a partir de la aplicación de los indicadores es de su agrado?	2. ¿Si Ud. fuera a realizar otro proyecto aplicaría estos mismos indicadores?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada cual en la escala de satisfacción siguiente:

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

Para ambos casos (proceso e indicadores) en los 2 pilotos fue 1, clara satisfacción. Esta se necesita para calcular el Índice de Satisfacción Grupal (ISG) mediante la fórmula (3):

$$ISG = \frac{A(+1) + B(+2) + C(0) + D(-0.5) + E(-1)}{N} \quad (3)$$

Donde:

- A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5
- N representa el número total de sujetos del grupo.

El ISG oscila entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

El ISG obtenido para ambos casos fue 1: $ISG = \frac{2(+1)}{2} = 1$

El ISG 1 significa el máximo de satisfacción respecto al proceso y sistema de indicadores.

Además contempla además dos preguntas complementarias de carácter abierto. Estas preguntas permiten conocer las causas de los niveles de satisfacción obtenidos.

A estas preguntas los pilotos respondieron:

Para el PB de IR:

Pregunta 4

Piloto 1: No, considero que el proceso es muy efectivo y los productos de trabajo cumplen las expectativas de los miembros del proyecto.

Piloto 2: No. Las actividades y productos de trabajo son los adecuados.

Pregunta 5

Piloto 1: Considero que el proceso es efectivo, permite obtener una visión más clara de las funcionalidades que el cliente desea que la aplicación a desarrollar tenga, dejando el margen de error en cuanto al proceso de definición de requisitos reducido.

Piloto 2: Sí, considero de mucha utilidad el proceso definido para la administración y el desarrollo de los requisitos de un proyecto, pues mediante el mismo se pudieron definir, desarrollar y validar los requisitos del cliente, identificando además inconsistencias en el trabajo realizado por el equipo de proyecto y los productos de trabajo asociados a dicho proceso.

Para el sistema de indicadores:

Pregunta 4

Piloto 1: No, los resultados de aplicar los indicadores propuestos reflejan el estado del proceso de requisitos en el proyecto.

Piloto 2: Me gustaría incluir algún indicador que midiese que tan correcta fue la Priorización de los requisitos realizada.

Pregunta 5

Piloto 1: No. Considero son adecuados y suficientes para el proyecto.

Piloto 2: El indicador Volatilidad, me gustaría que brindara la información de que en caso de existir algún cambio en algún requisito, qué tipo de cambio fue el que se efectuó.

El análisis de los resultados de esta técnica permitió confirmar la satisfacción de los usuarios. Se obtuvieron además recomendaciones para enriquecer el sistema de indicadores. De manera

general los pilotos quedaron satisfechos con el proceso y el sistema de indicadores considerándolos adecuados y útiles para sus proyectos.

4.4 Triangulación de resultados

La triangulación es definida por Denzin (1978) como la combinación de metodologías para el estudio del mismo fenómeno. (...)La triangulación es llamada también “convergencia metodológica”, “método múltiple” y “validación convergente”, pero en todas estas nociones subyace el supuesto de que los métodos cualitativos y cuantitativos deben ser considerados no como campos rivales sino complementarios. En todos los diversos diseños de triangulación está implícita la asunción de que su efectividad se basa en la premisa de que las debilidades de cada método individual van a ser compensadas por la fortaleza contra balanceadora del otro. (CORUJO, 2003)

La triangulación no solamente garantiza la validez de un estudio mostrando que sus conclusiones no dependen del modo utilizado para recolectar y analizar los datos, sino también permite enriquecer las conclusiones, otorgar mayor confiabilidad, mayor nivel de precisión y contrastar la consistencia interna del estudio. Puede utilizarse como un enfoque para fundamentar más el conocimiento obtenido con los métodos cualitativos, entendiendo que fundamentar no apunta a la evaluación de los resultados sino a la extensión sistemática de las posibilidades de producción de conocimiento.(STASIEJKO *et al.*, 2009)

4.4.1 Aplicación de la triangulación y resultados alcanzados

A continuación se presenta la estrategia de triangulación de los resultados de la validación obtenidos en la investigación. Ver figura 13.

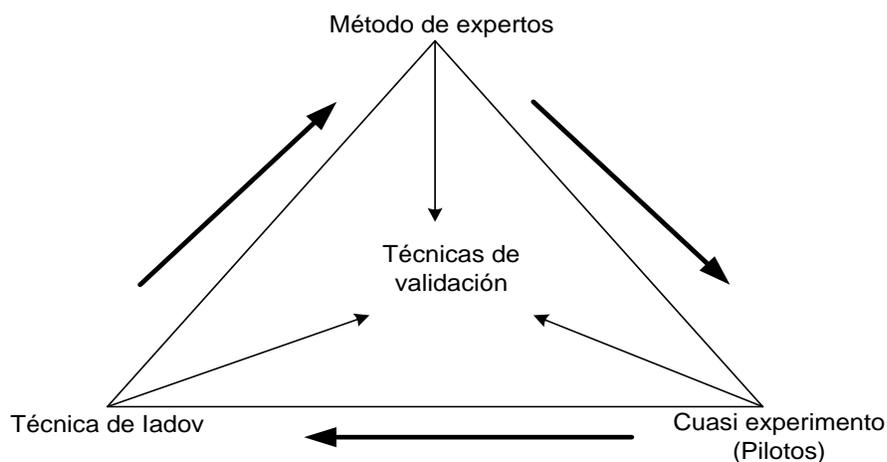


Figura 13. Triangulación de resultados de la validación

El proceso e indicadores se ubicaron bajo la lupa de los expertos (aplicación del método de expertos) que valoraron los indicadores en su mayoría como muy importantes para el proyecto y al proceso como adecuado a los objetivos de la investigación. A partir de esta primera validación que refería a la investigación encontrarse en la senda correcta se realizó un cuasi experimento mediante el pilotaje del PB de IR y los indicadores. Este pilotaje se ejecutó en 2 proyectos, recolectando el resultado de su aplicación. Mediante el método entrevista y observación se obtuvieron datos sobre proyectos similares que permitieron realizar las comparaciones correspondientes permitiendo confirmar la hipótesis planteada. A estos proyectos pilotos se les aplicó la Técnica de ladov con el fin de conocer el nivel de satisfacción desde el punto de vista del usuario respecto a la utilización del proceso e indicadores. Esta técnica arrojó resultados positivos: los clientes alegaron un máximo de satisfacción e hicieron recomendaciones de mejora.

Conclusiones parciales

- Mediante la utilización del método de expertos se validó el PB de IR y el sistema de indicadores propuestos, la mayoría de los expertos coincidieron en que el proceso disminuye el esfuerzo en los proyectos, variable que permite confirmar el aumento de la productividad.
- Con la implantación en proyectos pilotos, se ratificó la opinión de los expertos, los pilotos tuvieron una mayor productividad que otros proyectos similares.
- La técnica de ladov empleada para evaluar la satisfacción de los que usaron el proceso e indicadores corroboró su satisfacción. Estos se consideraron útiles y adecuados.
- Utilizando el método de triangulación metodológica se constató la convergencia de los resultados obtenidos a partir de la aplicación de los métodos y técnicas de validación. Se arribó a la conclusión de que la solución propuesta es adecuada y resuelve el problema de la investigación.

Conclusiones y Recomendaciones

Conclusiones

- Se obtuvo un proceso base para la ingeniería de requisitos compuesto por descripciones gráfica y textual, roles y responsabilidades, productos de trabajos con sus plantillas predeterminadas, propuesta de técnicas y herramientas y un sistema de indicadores. Estos componentes contribuyen a aligerar el peso que constituyen los primeros pasos en un programa de mejora debido a la incertidumbre de elegir una nueva forma de trabajo.
- Se propusieron los productos de trabajo del proceso que formarán parte de la base de conocimientos de la organización propiciando la reutilización, el registro de datos históricos para la toma de decisiones y la mejora continua.
- El PB de IR diseñado reduce el tiempo para las empresas desarrolladoras de software por concepto de definición de proceso. Sus componentes apoyan el aumento de la producción permitiendo a los proyectos centrarse en el desarrollo evitando costos relacionados con su investigación, búsqueda y definición.
- El sistema de indicadores obtenido incluye indicadores que miden elementos técnicos que permiten controlar los requisitos del proyecto y del proceso que favorecen su optimización.
- Los Jefes de Proyectos, Analistas, Arquitectos de Software y Administradores de la calidad obtienen un claro entendimiento de qué deben hacer a través de la descripción de sus responsabilidades y la gráfica del proceso que los va situando en cada actividad.
- A través de la aplicación práctica se demostró que el proceso es útil y de fácil adopción por los proyectos.
- La triangulación de los resultados de la valoración de los expertos, la aplicación práctica y la evaluación de la satisfacción desde el punto de vista de uso, confirma la adecuada solución al problema de la investigación.

Recomendaciones

- Diseñar una estrategia de implantación del PB de IR y el Sistema de Indicadores en los proyectos de desarrollo de software de las PYME del país.
- Optimizar el proceso e indicadores a partir de los datos recolectados durante la aplicación de la estrategia de implantación.
- Acotar la propuesta de técnicas y herramientas a partir de las experiencias de su uso.
- Definir actividades de capacitación para todos los miembros de los proyectos de desarrollo de software y en especial para Jefes de Proyectos y Analistas.

Bibliografía

AURUM, A. y WOHLIN, C. Engineering and Managing Software Requirements. New York: Springer, 2005. ISBN 10 3-540-25043-3.

CASSIDY, A. y GUGGENBERGER, K. A Practical Guide to Information Systems Process Improvement. Boca Raton: CRC Press LLC, 2001. ISBN 1-57444-281-3.

DAVIES, J.; VAN HARMELEN, F., et al. Towards the semantic web: ontology-driven knowledge management. John Wiley & Sons, Inc. New York, NY, USA, 2002.

DYBÅ, T.; DINGSØYR, T., et al. PROCESS IMPROVEMENT IN PRACTICE A Handbook for IT Companies. Editado por: Basili, V. R. Boston: KLUWER ACADEMIC PUBLISHERS, 2004. ISBN 1-4020-7869-2.

ELEJABARRIETA, F. J. y IÑIGUEZ, L. CONSTRUCCION DE ESCALAS DE ACTITUD TIPO THURST Y LIKERT. 1984.

FERNÁNDEZ, I. Page 1. NTP 15: Construcción de una escala de actitudes tipo Likert. 2009, nº Disponible en: http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/001a100/ntp_015.pdf.

GUÍA DEL PMBOK. Guía de los Fundamentos de la Dirección de Proyectos. Editado por: Edición, R. 2004.

JACOBSON, I.; BOOCH, G., et al. El proceso unificado de desarrollo de software. Madrid: 2000. 464 p. ISBN 84-7829-036-2.

LEFFINGWELL, D. y WIDRIG, D. Managing Software Requirements: A Use Case Approach, Second Edition. Boston: Addison Wesley, 2003. ISBN 0-321-12247-X.

L.SAATY, T. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill International Book Co., 1980.

OKTABA, H.; ESQUIVEL, C. A., et al. Modelo de Procesos para la Industria de Software MOPROSOFT. Secretaría de Economía - México, 2005.

OKTABA, H. y PIATTINI, M. Software Process Improvement for Small and Medium Enterprises Techniques and Case Studies. Editado por: Global, I. 2008.

PERSSE, J. R. Process Improvement Essentials. O'Reilly, 2006. 350 p. ISBN 0-596-10217-8.

PRESSMAN, R. Ingeniería de Software un enfoque práctico. 2006.

SOFTWARE ENGINEERING INSTITUTE. CMMI for Development, Version 1.3.2010b. 482 p.

SOFTWARE ENGINEERING INSTITUTE. Process Maturity Profile, Software CMM 2004 Year End Update. 2005

SOMMERVILLE, I. Software Engineering. 8 ed. 2004.

STAAB, S. y STUDER, R. Handbook on ontologies. Springer Verlag, 2004.

STEPANEK, G. Software Projects Secrets: Why Software Project Fail. 2005.

SWEBOK. Guide to the Software Engineering Body of Knowledge. 2004.

UNIVERSIDAD DE SALAMANCA. Construcción de Escalas tipo Likert. Medición y Evaluación Educativa. 2005.

WIEGERS, K. E. Software Requirements, Second Edition. Redmond: Microsoft Press 2003. ISBN 0735618798.

WIEGERS, K. E. More About Software Requirements: Thorny Issues and Practical Advice. Redmond: Microsoft Press 2006. ISBN 0735622671.

YOUNG, R. The Requirements Engineering Handbook 2004.

Referencias Bibliográficas

AAQIB, I.; FARHAN, K. M., *et al.* A CRITICAL ANALYSIS OF TECHNIQUES FOR REQUIREMENT PRIORITIZATION AND OPEN RESEARCH ISSUES. *International Journal of Reviews in Computing*, 2009, vol. 1, nº Disponible en: <http://www.ijric.org/volumes/Vol1/2Vol1.pdf>. ISSN 2076-3336.

ALARCON, A. y SANDOVAL, E. *Herramientas CASE para ingeniería de Requisitos*. 2008, vol. 6, Disponible en: <http://www.revistasjdc.com/main/index.php/ccient/article/view/37>.

ALELÚ, M.; CANTÍN, S., *et al.* *Estudio de Encuestas. Métodos de Investigación*. 2011, Disponible en: http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/ENCUESTA_Trabajo.pdf.

ANÓN. Project Management Professionals in Mexico: Correcta Identificación de los Criterios de Aceptación de Entregables del Proyecto. 2009, nº Disponible en: <http://pmpmexico.blogspot.com/2009/03/definicion-para-asegurarse-de-que.html>.

ASTIGARRAGA, E. *EL MÉTODO DELPHI* Donostia: Universidad de Deusto, [Consultado el: 28 de junio de 2013]. Disponible en: http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan%20Diego/Plnaifi_Cuencas_Pregrado/Sept_29/Metodo_delphi.pdf.

AURUM, A. y WOHLIN, C. *Engineering and Managing Software Requirements*. 2005.

BASTARRICA, C. *Productividad en la Industria TIC* Disponible en: <http://dccuchile.blog.terra.cl/2011/04/27/productividad-en-la-industria-tic/>.

BORLAND. *Define, visualize, manage and deliver stakeholder requirements.*

CALLEJAS, M.; CASTILLO, L. Y., et al. *Heler: Una herramienta para la ingeniería de requisitos automatizada.* 2010, vol. 6,

CASAS ANGUITA, J.; REPULLO LABRADOR, J., et al. *La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I).* 2003, vol. 31, Disponible en: <http://www.elsevier.es/es/revistas/atencion-primaria-27/la-encuesta-como-tecnica-investigacion-elaboracion-cuestionarios-13047738-investigacion-cualitativa-2003>.

CMMI INSTITUTE. *Published Appraisal Results.* 2011

CMMI, P. T. *CMMI® for Development, Version 1.2* Software Engineering Institute. Carnegie Mellon,

COEPES. *IMPORTANCIA DE LAS PYMES EN MÉXICO* Comisión Estatal para la Planeación de la Educación Superior, Disponible en: <http://www.comunicacion.ugto.mx/coepes/the-news/255-pymesroque>.

CORUJO, B. M. *Ciclo de Complementación Curricular. Licenciatura en Kinesiología y Fisiatría. TRIANGULACIÓN EN LA INVESTIGACIÓN CIENTÍFICA.* En 2003.

CHAOS'01. *Extreme chaos. The Standish group report.* [Consultado el: septiembre 2010 Disponible en: http://www.standishgroup.com/sample_research.

CHAOS'94. *THE STANDISH GROUP REPORT.* 1995

CHARETTE, R. N. *Why Software Fails* Disponible en: <http://spectrum.ieee.org/computing/software/why-software-fails>.

DEFINICIÓN. *Definición de* Disponible en: <http://definicion.de/productividad/>.

DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN. *Ingeniería del Software. I.T.I Sistemas. Capítulo 2. Ingeniería de Requisitos.* En 2010.

DURAN, A. y BERNÁRDEZ, B. *Metodología para la Elicitación de Requisitos de Sistemas Software.* Versión 2.3. 2000, nº Disponible en: http://www.infor.uva.es/~mlaguna/is1/materiales/metodologia_elicitacion.pd.

ESCALONA, M. J. y KOCH, N. *Ingeniería de requisitos en aplicaciones para la web. Un estudio comparativo*. Sevilla: Universidad de Sevilla, 2002, Disponible en: http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fwww.sistemas.edu.bo%2Fsalgado%2Fsis3390%2FRequisitos%2FIngenier_a%2520de%2520Requisitos%2520en%2520Aplicaciones%2520Web%2520-%2520Un%2520estudio%2520compatativo.pdf&ei=gHO2UIq5PIeS0QG7yYDgBA&usg=AFQjCNH4PdR_tnjH8Pm3AJiKsKt7uSfWmw&cad=rja.

EUROPEAN COMMISSION. *The New SME definition User guide and model declaration* de 2012]. Disponible en: http://ec.europa.eu/enterprise/policies/sme/files/sme_definition/sme_user_guide_en.pdf.

FEBLES, A. *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software*. Tesis de Doctorado en Ciencias Técnicas, Instituto Superior Politécnico "José Antonio Echeverría". Facultad de Ingeniería Industrial. Centro de Estudios de Ingeniería y Sistemas, 2003.

GALINDO, A. *REPERCUSIONES DE LA DEFINICIÓN DEL TAMAÑO EMPRESARIAL EN LOS RESULTADOS EMPÍRICOS SOBRE EFICIENCIA Y FINANCIACIÓN*. En *Las Medianas, Pequeñas y Micro-Empresas del Siglo XXI*. 2007.

GONZALO. *Definición de productividad* (Economía para Todos). Disponible en: <http://www.territorioempresas.com/2013/01/08/definicion-de-productividad/>.

GUÍA DEL PMBOK. *Guía de los Fundamentos de la Dirección de Proyectos*. Editado por: Edición, R. 2004.

HADAD, G.; DOORN, J., et al. *Facilitando la Asignación de Prioridades a los Requisitos*. En 2009.

HARRIS, R. *Técnicas de creatividad* Disponible en: http://www.innovaforum.com/tecnicabrain_e.htm.

HARTMANN, D. *Interview: Jim Johnson of the Standish Group* Disponible en: <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>.

HERNÁNDEZ, C. P. Explotación de los corpórea textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento. *IRIS*, 2002, vol. 18, nº Disponible en: <http://elies.rediris.es/elies18/522.html>. ISSN 1139-8736.

HERNÁNDEZ, Y. *Guía para la Arquitectura de la Información de contenidos web en dispositivos móviles*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2011.

HERRERA, L. J. "Ingeniería de Requerimientos, Ingeniería de Software". 2003, nº

IBM. *Rational RequisitePro. A requirements management tool* [Consultado el: Noviembre 2012

---. *Rational Rose Enterprise* [Consultado el: Noviembre 2012 Disponible en: <http://www-01.ibm.com/software/awdtools/developer/rose/enterprise/>].

IEEE. *IEEE Recommended Practice for Software Requirements Specifications*. Editado por: Society, S. E. S. C. O. T. I. C. 2009, Disponible en: <http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>.

---. *SEWBOK, Guide to the Software Engineering Body of Knowledge*. 2004.

"IEEE Standar Glossary Software Engineering Terminology". 2006, nº Disponible en: http://standards.ieee.org/reading/ieee/std_public/description/se.

INCOSE. *INCOSE SE Tools Database: Requirements Management Tool Summaries and Vendor Contact Information*. INCOSE. 2011

INTECO. *GUÍA AVANZADA DE GESTIÓN DE REQUISITOS*. 2008, Disponible en: www.inteco.es/file/TnOlvX7kM5rIBliD4wnTxQ

ISO 9000:2000. *ISO 9000:2000 Sistemas de gestión de la calidad — Fundamentos y vocabulario*. 2000.

ISO 12207. *INTERNATIONAL STANDARD ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES*. 2005.

JACOBSON, I.; BOOCH, G., et al. *El proceso unificado de desarrollo de software*. Madrid: 2000. 464 p. ISBN 84-7829-036-2.

L.SAATY, T. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill International Book Co., 1980.

LEFFINGWELL, D. y WIDRIG, D. *Managing Software Requirements: A Use Case Approach, Second Edition*. Boston: Addison Wesley, 2003. ISBN 0-321-12247-X.

LEMES, A. y MACHADO, T. *LAS PYMES Y SU ESPACIO EN LA ECONOMÍA LATINOAMERICANA*. En *SEGUNDO ENCUENTRO INTERNACIONAL SOBRE Las Medianas, Pequeñas y Micro-Empresas del Siglo XXI*. 2007.

- LÓPEZ, R. E. y DESLAURIERS, J.-P. *La entrevista cualitativa como técnica para la investigación en Trabajo Social*. 2011, vol. No 61 junio de 2011, Disponible en: <http://www.margen.org/suscri/margen61/lopez.pdf>.
- MANIASI, S. D. *Identificación de Riesgos de Proyectos de Software en Base a Taxonomías*. Maestría, ITBA. 2005.
- MEDINA, R. *La calidad como estrategia de mejora continua para las PyMES mexicanas*. Licenciatura, Facultad de Contaduría y Administración. Universidad Veracruzana, 2010.
- MÉNDEZ, M. y CARMEN, C. D. *Los Retos de las PYMES para consolidar el desarrollo económico de México*. Universidad de Sonora, 2007.
- MENDOZA, L. E. *Sistema de Información II Teoría. Determinación de requerimientos entrevistas, cuestionarios, observaciones, JAD, prototipos, case, groupware*. En 2012.
- MULLA, N. y GIRASE, S. Comparison of various Elicitation Techniques and Requirement Prioritisation Techniques. *International Journal of Engineering Research & Technology (IJERT)*, 2012, vol. I, nº 3, Disponible en: <http://www.ijert.org/browse/may-2012-edition?download=153%3Acomparison-of-various-elicitation-techniques-and-requirement-prioritisation-techniques&start=120>. ISSN 2278-0181.
- OKTABÁ, H.; ESQUIVEL, C. A., et al. *Modelo de Procesos para la Industria de Software MOPROSOFT*. Secretaría de Economía - México, 2005.
- OKTABÁ, H. y PIATTINI, M. *Software Process Improvement for Small and Medium Enterprises Techniques and Case Studies*. Editado por: Global, I. 2008.
- PÉREZ, K. *Modelo de Referencia para la Ingeniería de Requisitos en Proyectos de Bioinformática*. Maestría, Universidad de las Ciencias Informáticas, 2007.
- PÉREZ, Y. *Técnicas y herramientas de la ingeniería de requisitos adecuadas para simuladores virtuales*. Maestría, Gestión de Proyectos Informáticos. Universidad de las Ciencias Informáticas, 2009.
- PRESSMAN, R. *Ingeniería de Software un enfoque práctico*. 2006.
- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. . 5ta ed. 2002.

RAMOS, K.; SUÁREZ, A., et al. *Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas*. Cuba: 2011, Disponible en: rcci.uci.cu/index.php/rcci/article/download/87/75. ISBN 1994-1536.

REAL ACADEMIA ESPAÑOLA. *Diccionario de la Lengua Española - Vigésima segunda edición* Vigésima segunda edición. ed. Disponible en: http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=calidad

ROBERTSON, S. y ROBERTSON, J. *Mastering the Requirements Process Second Edition*. 2006.

ROSIQUE, F.; JIMÉNEZ, M., et al. Evaluación de herramientas de gestión de requisitos. *III Jornadas de Introducción a la Investigación de la UPCT*, 2010, nº

SOFTEX. *MPS.BR - Mejora de Proceso del Software Brasileño Guía General*. Brasil: 2009. 55 p. ISBN 978-85-99334-15-7.

SOFTWARE ENGINEERING INSTITUTE. *CMMI for Development (CMMI-DEV)* [Consultado el: Octubre 2010 Disponible en: <http://www.sei.cmu.edu/cmmi/tools/dev/index.cfm>.

SOFTWARE ENGINEERING INSTITUTE. *CMMI for Development, Version 1.3*. 2010b. 482 p.

SOFTWARE ENGINEERING INSTITUTE. *Process Maturity Profile, Software CMM 2004 Year End Update*. 2005

SOMMERVILLE, I. Ingeniería del Software. En séptima edición ed. 2005, p. 687.

SOMMERVILLE, I. *Software Engineering*. 8 ed. 2004.

SPARX SYSTEMS. *Visual Modeling Platform* [Consultado el: Noviembre 2012 Disponible en: <http://www.sparxsystems.com/products/ea/index.html>.

STANDISH GROUP. *Chaos Report*. 1995

STANDISH GROUP. *Chaos summary 2009*. 2009

STANDISH GROUP. *CHAOS SUMMARY FOR 2010*. 2010

STASIEJKO, H. A.; TRISTANY, S. R., *et al.* LA TRIANGULACIÓN DE DATOS COMO CRITERIO DE VALIDACIÓN INTERNO EN UNA INVESTIGACIÓN EXPLORATORIA. En 2º Congreso Internacional de Investigación de la Facultad de Psicología de la Universidad Nacional de La Plata. La Plata. Argentina. 2009.

SWEBOK. *Guide to the Software Engineering Body of Knowledge*. 2004.

TARDÍO, M. A.; FEBLES, A., *et al.* *Primeras ideas de un Modelo cubano de referencia para el desarrollo de aplicaciones informáticas*. Cuba: Ediciones Futuro, 2011, ISBN 1994-1536.

TIGRIS.ORG. *ArgoUML* [Consultado el: Noviembre 2012 Disponible en: <http://argouml.tigris.org/>].

VALLADARES, A. S. *Proceso de Gestión de riesgos para proyectos de desarrollo de software de Softel*. Maestría, Gestión de Proyectos Informáticos. Universidad de las Ciencias Informáticas, 2010.

VISCONTI, M. *Marcello Visconti*. En *Ingeniería de Software Avanzada*. 2003.

VISUAL PARADIGM. *Visual Paradigm for UML - UML tool for software application development* [Consultado el: Noviembre 2012 Disponible en: <http://www.visual-paradigm.com/product/vpuml/>].

VISURE. IrqA general information. 2010, nº Disponible en: http://www.visuresolutions.com/c/document_library/get_file?uuid=de2734ea-f173-437b-a216-a4cf6091f1cc&groupId=10826.

WALL, R.; KOZA, E., *et al.* *Executive Sponsorship & Stakeholder Engagement*. Editado por: Group, P.-P. 2012, Disponible en: http://www.pm-partners.com.au/wp-content/uploads/2012/09/Executive-Engagement_OnTarget-Slides_Release-09.2012.pdf.

WIEGERS, K. E. *First Things First: Prioritizing Requirements*. *Software Development*, 1999, nº Disponible en: <http://www.tarrani.net/linda/prioritizing.pdf>.

WIEGERS, K. E. *Software Requirements, Second Edition*. Redmond: Microsoft Press 2003. ISBN 0735618798.

WILLIAMS, L. M. *The Elements Of Technical Requirements Management*. En *For INCOSE Atlanta Chapter*. 2006.

YOUNG, R. *The Requirements Engineering Handbook* 2004.