

La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software

Michael Arias Chaves*

Recepción: 16 de junio de 2006

Aprobación: 7 de julio de 2006

Resumen:

Con la realización de este artículo se trata de dar un mejor panorama acerca de los conceptos y características de la Ingeniería de Requerimientos (IR), buscando resaltar su importancia dentro del ciclo de desarrollo de proyectos de desarrollo de software, conocer las diferentes alternativas o técnicas que existen para identificarlos, así como mostrar la importancia que tienen herramientas automatizadas dentro de este proceso de administración de requerimientos.

Palabras claves: Requerimientos, ingeniería de requerimientos, herramientas, técnicas.

Abstract:

With the realization of this article it is to give a better panorama about the concepts and characteristic of the Engineering of Requirements (IR), looking for to stand out their importance inside the cycle of development of projects of software development, to know the different alternative or techniques that exist to identify them, as well as to show the importance that have the automated tools inside this process of administration of requirements.

Key Words: Requirements, Requirements Engineering, Tools, Techniques

1. Introducción

A través de los años se ha podido constatar que los requerimientos o requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que será uno de los principales mecanismos de control con los que se contará durante la etapa de desarrollo. Además la especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema y es contra lo que se va a estar verificando si se están cumpliendo las metas trazadas.

Es muy frecuente escuchar entre los conocedores del desarrollo de software (programas de computadoras), que un gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos. Dentro de esa mala administración se pueden encontrar factores como la falta de participación del usuario, requerimientos incompletos y el mal manejo del cambio a los requerimientos.

* Profesor del Departamento de Ciencias Naturales de la Sede de Occidente de la Universidad de Costa Rica
[mike_arias@hotmail.com
]

La Ingeniería de Requerimientos (IR) cumple un papel primordial en el proceso de producción de software, ya que se enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas.

2. Ingeniería de requerimientos: conceptos y características

Como se menciona anteriormente, la ingeniería de requerimientos sirve como una base sólida en el proceso de desarrollo de software, por lo que antes de pasar a tratar los aspectos referentes a la administración adecuada de los requerimientos, es importante primero definir lo que es un requerimiento y cuales serían las características deseables que deberían de tener.

2.1 ¿Qué son Requerimientos?

Se presenta a continuación la definición existente en el glosario de la IEEE de lo que es un "Requerimiento":

1. "Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo". (Std 610.12-1900, IEEE: 62)
2. "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".
(Std 610.12-1900, IEEE: 62)

También, Ian Sommerville presenta una definición acerca de lo que es un "Requerimiento":

3. "Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste". (Sommerville, 2005: 108)

Analizando las definiciones anteriores, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

2.2 Tipos de Requerimientos

Los requerimientos de software pueden dividirse en 2 categorías: requerimientos funcionales y requerimientos no funcionales.

Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.

Por otra parte los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

2.3 Características de un Requerimiento

Es importante no perder de vista que un requerimiento debe ser:

Especificado por escrito: Como todo contrato o acuerdo entre dos partes.

Posible de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿cómo se sabe si se cumplió con él o no?

Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento.

No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

2.4 Dificultades para definir los requerimientos

Durante la etapa de especificación de requerimientos se pueden presentar muchos inconvenientes los cuales son importantes de identificar y prevenir, a continuación se presenta un listado con los problemas más comunes en este proceso:

Los requerimientos no son obvios y vienen de muchas fuentes.

Son difíciles de expresar en palabras (el lenguaje es ambiguo).

La cantidad de requerimientos en un proyecto puede ser difícil de manejar.

Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.

El usuario no puede explicar lo que hace

Tiende a recordar lo excepcional y olvidar lo rutinario

Hablan de lo que no funciona

Los usuarios tienen distinto vocabulario que los desarrolladores.

Usan el mismo término con distinto significado

2.5 Ingeniería de requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa.

Algunos otros conceptos de ingeniería de requerimientos son:

“Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”. (Pressman, 2006: 155)

“La ingeniería de requerimientos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema”. (Sommerville, 2005: 82)

En síntesis, el proceso de ingeniería de requerimientos se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto de software determinado, donde es muy importante tomar en cuenta que el aporte de la IR vendrá a ayudar a determinar la viabilidad de llevar a cabo el software (si es factible llevarlo a cabo o no), pasando posteriormente por un subproceso de obtención y análisis de requerimientos, su especificación formal, para finalizar con el subproceso de validación donde se verifica que los requerimientos realmente definen el sistema que quiere el cliente.

2.6 Importancia de la ingeniería de requerimientos

Según la autora Lizka Johany Herrera en su documento de la ingeniería de requerimientos, los principales beneficios que se obtienen de la Ingeniería de Requerimientos son (2003: 3):

Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.

Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.

Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la IR, ya que es una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo.

Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.

Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

2.7 Actividades de la ingeniería de requerimientos

Dentro del mismo documento mencionado anteriormente (Herrera, 2003: 6), se dice que dentro de la IR existen cuatro actividades básicas que se tienen que llevar a cabo para completar el proceso. Estas actividades ayudan a reconocer la importancia que tiene para el desarrollo de un proyecto de software realizar una especificación y administración adecuada de los requerimientos de los clientes o usuarios. Las cuatro actividades son: extracción, análisis, especificación y validación, y serán explicadas a continuación cada una de ellas.

2.7.1 Extracción

Esta fase representa el comienzo de cada ciclo. Extracción es el nombre comúnmente dado a las actividades involucradas en el descubrimiento de los requerimientos del sistema. Aquí, los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar, etc. Es importante, que la extracción sea efectiva, ya que la aceptación del sistema dependerá de cuan bien éste satisfaga las necesidades del cliente.

2.7.2 Análisis

Sobre la base de la extracción realizada previamente, comienza esta fase en la cual se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento.

Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requerimientos; en esta etapa se leen los requerimientos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requerimientos.

2.7.3 Especificación

En esta fase se documentan los requerimientos acordados con el cliente, en un nivel apropiado de detalle.

En la práctica, esta etapa se va realizando conjuntamente con el análisis, se puede decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML (Lenguaje de Modelado Unificado), que es un estándar para el modelado orientado a objetos, por lo que los casos de uso y la obtención de requerimientos basada en casos de uso se utiliza cada vez más para la obtención de requerimientos.

2.7.4 Validación

La validación es la etapa final de la IR. Su objetivo es, ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento especificado para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos.

Se puede apreciar que el proceso de ingeniería de requerimientos es un conjunto estructurado de actividades, mediante las cuales se obtiene, se valida y se logra dar un mantenimiento adecuado al documento de especificación de requerimientos, que es el documento final, de carácter formal, que se obtiene de este proceso. Es necesario recalcar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. Hay muchas maneras de organizar el proceso de ingeniería de requerimientos y en otras ocasiones se tiene la oportunidad de recurrir a consultores, ya que ellos tienen una perspectiva más objetiva que las personas involucradas en el proceso.

3. Técnicas y herramientas utilizadas en la ingeniería de requerimientos

3.1 Técnicas utilizadas en las actividades de IR

Existen varias técnicas para la IR propuestas para ingeniería de requerimientos (Herrera, 2003: 12), y de las cuales en este artículo solo se abarcarán cinco de ellas. Es importante resaltar que estas técnicas pueden ser aplicables a las distintas fases del proceso de la IR, haciendo la salvedad de que hay que tomar en cuenta las características propias del proyecto en particular que se este desarrollándose para aprovechar al máximo su utilidad.

3.1.1 Entrevistas y Cuestionarios

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema.

Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. El éxito de esta técnica, depende de la habilidad del entrevistador y de su preparación para la misma.

3.1. 2 Sistemas existentes

Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, podemos analizar las interfases de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas.

3.1.3 Lluvia de ideas (Brainstorm)

Este es un modelo que se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requerimientos para el sistema. No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando en base a distintos criterios como, por ejemplo, "caro", "impracticable", "imposible", etc.

Las reglas básicas a seguir son:

Los participantes deben pertenecer a distintas disciplinas y, preferentemente, deben tener mucha experiencia. Esto trae aparejado la obtención de una cantidad mayor de ideas creativas.

Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, porque sino se crea un ambiente hostil que no alienta la generación de ideas.

Por más locas o salvajes que parezcan algunas ideas, no se las debe descartar, porque luego de maduras probablemente se tornen en un requerimiento sumamente útil.

A veces ocurre que una idea resulta en otra idea, y otras veces podemos relacionar varias ideas para generar una nueva.

Escribir las ideas sin censura.

3.1.4 Prototipos

Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requerimientos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final.

Entonces, para validar los requerimientos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiéndonos conseguir

una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva.

El desarrollo del prototipo comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un “diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo.

3.1.5 Casos de Uso

Los casos de uso son una técnica para especificar el comportamiento de un sistema. El sitio en Internet wikipedia.org, define a un caso de uso como:

“Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas” (http://es.wikipedia.org/wiki/Caso_de_uso).

Los casos de uso permiten entonces describir la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor, es una descripción de un conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema. La mayoría de los requerimientos funcionales, sino todos, se pueden expresar con casos de uso. Según el autor Sommerville, los casos de uso son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente, se han convertido en una característica fundamental de la notación UML (Lenguaje de modelado unificado), que se utiliza para describir modelos de sistemas orientados a objetos.

3.2 Herramientas automatizadas para la Administración de Requerimientos

En el desarrollo de software se cuenta con una ventaja proporcionada por las herramientas CASE. Las herramientas CASE (Ingeniería del Software Asistida por Computadora) se le conoce a todo aquel software que es usado para ayudar a las actividades del proceso de desarrollo del software, en donde se ubica la ingeniería de requerimientos, que se ha venido tratando en este artículo. Estas herramientas se concentran en capturar requerimientos, administrarlos y producir una especificación de requisitos.

Existen muchas y muy variadas herramientas CASE que pueden ser utilizadas por los desarrolladores de software en sus proyectos, y de la forma más conveniente para ellos. Si es importante hacer ver que estas herramientas fungen como un medio facilitador para agilizar y mejorar los procesos involucrados en

todo el ciclo de vida presentado por la IR, y que en conjunto ayudan a la construcción final de un producto de software terminado.

Estas herramientas permiten entre otras cosas tener un mayor control en proyectos complejos, reducir costos y retrasos en los proyectos, ayudan a determinar la complejidad y los esfuerzos necesarios.

En este apartado se presentan características generales de una de las herramientas más utilizadas para este propósito: *RequisitePro*, y recomendada sitio en Internet Rational.com.

3.2.1 RequisitePro

RequisitePro es la herramienta que ofrece Rational Software para tener un mayor control sobre los requerimientos planteados por el usuario y todos aquellos requerimientos técnicos o nuevos requerimientos de usuario que surjan durante el ciclo de vida del proyecto.

En RequisitePro los requerimientos se encuentran documentados bajo un esquema organizado de documentos; estos esquemas cumplen completamente con los estándares requeridos por algunas de las instituciones a nivel mundial más reconocidas en el desarrollo de software, tales como: IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), ISO, CMM (Modelo de Capacidad de Madurez) y por el RUP (Proceso Unificado Racional)

Esta herramienta se integra con aplicaciones para la administración de cambios, herramientas de modelado de sistemas y con herramientas de pruebas. Esta integración asegura que los diseñadores conocen los requerimientos del usuario, del sistema y del software en el momento de su desarrollo.

El desarrollo de software es una tarea de equipo, de tal forma, es crítico que todos los miembros del equipo posean un entendimiento compartido de la visión de sus proyectos, metas, especificaciones y requerimientos; pero, ¿cómo puede conseguirse cuando los equipos se encuentran geográficamente distribuidos y funcionalmente aislados, no pudiendo comunicarse entre si en tiempo y forma? La solución a esta necesidad es IBM Rational RequisitePro. IBM Rational RequisitePro es una solución fácil de usar, es una herramienta de administración de requerimientos que le permite al equipo crear y compartir sus requerimientos utilizando métodos familiares basados en documentos potenciados por la aplicación de las capacidades de una base de datos, tales como la trazabilidad y análisis de impacto. El resultado es una mejor comunicación y administración de requerimientos con una mayor probabilidad de completar los proyectos en tiempo, dentro del presupuesto y superando las expectativas. Los proyectos exitosos comienzan con una buena administración de requerimientos, cuanto más efectiva sea su ejecución, mayor será el resultado en calidad y satisfacción del cliente.

Según la promoción hecha en Internet mediante la página Web para esta herramienta, algunas de sus ventajas son:

- Un producto potente y fácil de utilizar para la gestión de requisitos y casos de uso que propicia una mejor comunicación, mejoras en el trabajo en equipo y reduce el riesgo de los proyectos.

- Combina la interfaz conocida y fácil de utilizar de los documentos de Microsoft Word con potentes funciones de base de datos para conseguir la máxima eficacia en análisis y consulta de requisitos.
- Proporciona a los equipos la posibilidad de comprender el impacto de los cambios.
- Garantiza que todos los componentes del equipo estarán informados de los requisitos más actuales para asegurar la coherencia.
- Proporciona acceso basado en Web para los equipos distribuidos.

La ventaja de utilizar herramientas como la de RequisitePro, es que el desarrollo de software se ve beneficiado de muchas maneras, y en el caso de la ingeniería de requerimientos, le ayuda notablemente, ya que como se ha venido hablando en el desarrollo de este artículo, la IR constituye una de las etapas más importantes a tomar en cuenta en el ciclo de desarrollo de software, ya que en ella se definen los requerimientos con los que debe de contar el software; e incluso, podría llegar a determinar la viabilidad de implementar ese software no es del todo posible, y poder cancelar a tiempo un desarrollo no productivo.

4. Conclusiones

La evolución de los estudios encarados por la Ingeniería de Requerimientos se fue dando paulatinamente. Sin embargo, a partir de los 90, los esfuerzos se concentraron en la búsqueda de técnicas, métodos y herramientas que pudieran ser aplicados durante el proceso de definición de requerimientos para arribar a una etapa de diseño exitosa, dejando de lado la obtención de una metodología capaz de adaptarse a cualquier tipo de sistema y paradigma, brindando un marco de trabajo referencial, independiente del método a aplicar.

Es muy importante mencionar que el poder formular una especificación de requerimientos completa y consistente, es un paso muy importante para evitar cometer errores en la definición de los requerimientos, ya que los mismos pueden resultar muy caros de corregir una vez desarrollado el sistema. De ahí, la vital importancia que tiene la ingeniería de requerimientos en generar una adecuada especificación que contemple claramente y sin ambigüedades los requerimientos del sistema a desarrollar, con el fin primordial de evitar que los proyectos fracasen debido a una mala elaboración de la definición y especificación de requerimientos.

El proceso de la Ingeniería de Requerimientos sirve para recopilar la información necesaria para establecer la funcionalidad que se quiere alcanzar con el sistema. Para ello, se debe de contar con buenos métodos y técnicas para hacerlo, además de una comunicación fluida y constante con el cliente, ya que los requerimientos deben reflejar las necesidades reales que el cliente quiere satisfacer. Las revisiones deben involucrar al cliente y al staff de contratistas para validar los requerimientos del sistema. Como proceso, la administración de requerimientos es fundamental en todo proyecto de desarrollo de software, ya que se debe de contar con una especificación clara y completa desde las fases iniciales para

no tener problemas posteriores que impliquen un retraso en el cronograma, un presupuesto erróneo, o hasta la posible cancelación del proyecto. Es importante que el documento que se obtenga de esta etapa sea un reflejo real del acuerdo de las partes involucradas.

Hay que notar el aporte que ha venido a proporcionar la utilización de técnicas como la especificación, la lluvia de ideas y el desarrollo de prototipos, que ayudan a definir requerimientos de una manera concisa y real.

Además, hoy en día, es importante tomar en cuenta el uso de patrones en el desarrollo de software. Un patrón es un conjunto de información que aporta una buena solución ya probada a un problema que se presenta en un contexto determinado. Para elaborarlo se aíslan sus aspectos esenciales y se añaden cuantos comentarios y ejemplos que sean necesarios. ¿En qué ayuda esto en la IR? Esto ayuda a diseñar correctamente en menos tiempo, ayuda a construir problemas reutilizables y extendibles, y facilita la documentación y la comunicación con otros miembros del equipo de desarrollo, ventajas importantes que se deben de tomar en cuenta por parte los desarrolladores de software, ya que si se logra tener un lenguaje en común dentro del grupo de trabajo, se reduce el impacto de los riesgos al utilizar soluciones ya probadas, y amplía el conocimiento de los participantes del desarrollo, permitiéndoles representar mejores sistemas de una manera más estructurada.

La aparición de herramientas automatizadas para la administración de requerimientos, como lo es "*RequisitePro*", sirven de apoyo a los procesos de Ingeniería de Software, que se concentran en capturar requerimientos, administrarlos y producir una especificación de requisitos.

Ésta, y muchas otras herramientas, permiten tener un mayor control en proyectos complejos, reducir costos y retrasos en los proyectos, ayudan a determinar la complejidad y los esfuerzos necesarios; sin duda alguna, una gran ayuda para establecer ideas claras de lo que realmente se necesita para llevar a cabo una exitosa Ingeniería de Requerimientos, y por ende, un comienzo prometedor cuando se quiere tener éxito con un proyecto de software.

Por último, es necesario dar a conocer que alrededor del mundo existen estándares enfocados en el mejoramiento de los procesos de desarrollo de software, citando entre ellos a los estándares propuestos por la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), el SEI (que propone el modelo de capacidad de madurez, mejor conocido como CMM), el PMI (Project Management Institute, que ofrece certificaciones para el área de administración del proyectos) y las ya conocidas normas ISO, en cuyas normas también se involucran apartados referentes al desarrollo de software.

Para citar un ejemplo, hay empresas nacionales como el caso de Exactus, que ha logrado obtener certificaciones bajo los niveles 2 y 3 del CMM, lo que significa que esta empresa trabaja consistentemente con prácticas administrativas y de ingeniería en sus procesos, convirtiéndose en una de las casi 400 organizaciones de software en el mundo que han sido auditadas exitosamente para optar por esos niveles de certificación, según lo publicado en la página Web de esta organización.

La importancia de la existencia de esta serie de estándares radica en la posibilidad que tienen las empresas nacionales de contar con una serie de documentos que puedan servir como una guía para orientar mejor sus procesos de desarrollo (incluyendo la parte importante que corresponde a la ingeniería

de requerimientos), en vías de seguir siendo un país que es visto con buenos ojos a nivel mundial, aprovechando entre muchas otras cosas, el cuarto lugar que ostenta Costa Rica a nivel mundial con respecto a la tasa de alfabetización adulta (95.8% en población mayor a 15 años), según el reporte de Desarrollo Humano emitido por la ONU (Organización de la Naciones Unidas) en el año 2004 (CINDE, 2004); y abonado también a que la industria del software in Costa Rica ha emergido como una de las industrias más dinámicas en Latinoamérica, experimentando un crecimiento explosivo, donde se encuentran alrededor de 300 compañías relacionadas con el software que van desde pequeñas hasta grandes empresas que se dedican al desarrollo de software tanto para el mercado local como internacional, ofreciendo la más alta calidad en sus productos.

5. Bibliografía

Libros

Pressman, Roger S. 2006, “**Ingeniería del Software: Un enfoque práctico**”, Sexta edición, México DF, Editorial McGraw Hill.

Sommerville Ian, 2005, “**Ingeniería del Software**”, Séptima edición, México DF, Editorial Pearson.

Páginas Web

IEEE Std 610.12-1990, “**IEEE Standard Glossary of Software Engineering Terminology**”, recuperado el 24 de mayo de 2006 en: http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html

Herrera J., Lizka Johany (2003) “**Ingeniería de Requerimientos, Ingeniería de Software**”, Recuperado el 25 de mayo de 2006 en: <http://www.monografias.com/trabajos6/resof/resof.shtml>

Montes Meyhuay Magno, “Ingeniería de Requerimientos”, recuperado el 25 de mayo de 2006 en: www.proamazonia.gob.pe/bpa/ingenieria_requerimientos.htm

Organización Exactus, recuperado el 29 de mayo de 2006 en: www.exactus.com

Racional RequisitePro, recuperado el 30 de mayo de 2006 en: <http://www.rational.com.ar/herramientas/requisitepro.html>

Enciclopedia Wikipedia, recuperado el 19 de julio de 2006 en: http://es.wikipedia.org/wiki/Caso_de_uso

Coalición Costarricense de Iniciativas de Desarrollo (CINDE), recuperado el 19 de julio de 2006 en: <http://www.cinde.org/esp-servicios.shtml#AspectosGenerales>.