CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA PUBLICADO
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Ingeniería de requisitos desde las trincheras.

Requirements engineering from the trenches

MSc. Ing. Yadiel Ramos Rodríguez
Máster en Informática Aplicada
Profesor Asistente
yramosr@uci.cu
Universidad de las Ciencias Informáticas (UCI). Cuba

Ing. Dayana Daniel Hernández
Profesor Instructor
ddaniel@uci.cu
Universidad de las Ciencias Informáticas (UCI). Cuba

Resumen
En el presente trabajo se tratan aspectos prácticos de la ingeniería de requerimientos como los roles que deben participar en la misma así como su función, la utilización de técnicas de recopilación de información, los puntos de partida en la obtención de casos de uso y requerimientos, además de la forma de tratar estos últimos durante el proceso de desarrollo para obtener los mejores resultados. Se abordan las experiencias acumuladas del paso por diferentes proyectos productivos, en los que se aplicaron varias técnicas de captura de requerimientos, hasta llegar a tener un proceso particular que ha dado significativos beneficios.
Palabras clave: Casos de Uso, Proceso de Desarrollo, Requisito, TRI

Abstract
In the present work are discussed practical aspects of engineering requirements and the roles that should attend it as well as its function, using techniques of information gathering, the starting points for the collection of use cases and requirements, and how to treat them during the development process to achieve best results. Also try to review the process from an industrial approach, instead of academic.
Keywords: Development Process, Requisites, TRI, Use Cases

CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Introducción

El siguiente escenario es típico: un analista trabaja con los usuarios para describir los procesos de negocio que serán soportados por el software. El equipo de desarrollo recibe la descripción del analista pero no está familiarizado con los términos de negocio y considera la descripción demasiado informal. Los desarrolladores escriben su propia descripción desde un punto de vista técnico. El usuario no entiende esta descripción pero la acepta para que el proyecto avance. El resultado puede ser un sistema que desde el punto de vista del usuario es difícil de usar y que no cumple con sus expectativas.
Parte de este problema es metodológico, y en parte es intrínseco a las caracterí-sticas de los usuarios. Algunas de las problemáticas que se presentan son las siguientes:

1. Los usuarios no saben qué es lo que quieren.
2. Los usuarios no aceptan como un compromiso los requerimientos escritos.
3. Los usuarios insistirán en nuevos requerimientos después de fijar costos y agendas.
4. Los usuarios no están disponibles y la comunicación con ellos es lenta.
5. Los usuarios no participan en revisiones de avance.
6. Los usuarios no entienden el proceso de desarrollo y no les interesa.

El desarrollo de cualquier proyecto involucra tener en cuenta cómo transformar los deseos, expectativas y forma de trabajo del cliente en algo más técnico; donde dicho cliente lo siga comprendiendo, pero que además sea comprendido por el equipo de desarrollo. Para dar cumplimiento a ello es necesario tener en los equipos de desarrollo el rol de analista. Los analistas poseen un amplio rango de habilidades. La primera y principal es que el analista soluciona problemas, le gusta el reto de analizar y encontrar una respuesta funcional. Los analistas de sistemas requieren habilidades de comunicación que les permitan relacionarse en forma significativa con muchos tipos de personas diariamente, así como habilidades de computación. Para su éxito es necesario que se involucre el usuario final. También deben tener habilidades interpersonales excelentes, buena comunicación verbal y habilidades técnicas y no técnicas de la escritura.

Los analistas proceden sistemáticamente. El marco de referencia para su enfoque sistemático es proporcionado por lo que es llamado el ciclo de vida del desarrollo de sistemas (SDLC). Este puede ser dividido en siete fases secuenciales, aunque

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

en realidad las fases están interrelacionadas y frecuentemente se llevan a cabo simultáneamente. (Kendall, 2009) Las siete fases son:

1. Identificación de problemas.
2. Oportunidades y objetivos.
3. Determinación de los requerimientos de información.
4. Análisis de las necesidades de sistemas.
5. Diseño del sistema recomendado.
6. Desarrollo y documentación del software.
7. Implementación, prueba y mantenimiento del sistema.

De las mismas se puede notar que el trabajo del analista debe estar concentrado en las 4 primeras fases. No obstante la experiencia en desarrollo de software indica que han dado mayores resultados los proyectos en los cuales se ha involucrado un rol intermedio e inicial, que puede ser el mismo analista pero que se suele llamar ingeniero de procesos, y es la persona que se encarga de mostrarle al equipo de desarrollo un modelo mayormente automatizado mediante diagramas SIPOC (Proveedor ==> Ingreso ==> Proceso ==> Salida ==> Cliente), u otras herramientas como la Definición Integrada (IDEF), que le muestren incluso al cliente donde existen problemas en sus procesos, y le de posibles formas de mejorarlo. Así, cuando dichos procesos sean semi-automatizados o automatizados en su totalidad, los mismos tendrán mayores probabilidades de transitar una vida útil y efectiva como el cliente lo quiera.

Además, si inicialmente el analista es capaz de estudiarse dichos diagramas y la información que debe haber recolectado anteriormente el ingeniero de procesos, será mucho más fácil el poder dirigir cualquier técnica de recopilación de información (TRI) deseada. Realmente en ocasiones dicho rol ha tenido que ser ocupado por el propio analista, pero ha sido mejor separarlos debido a que siempre se busca personal con dominio de gestión de procesos, para no cargar ni tener que capacitar a los analistas en esta tarea.

Una vez terminado el trabajo del ingeniero de sistemas, el analista debe encaminar su trabajo mediante las TRI que estime necesarias, hacia lo que se conoce como el levantamiento de requisitos. Las TRI que más se realizan son entrevistas, cuestionario, inspección de registros y observación. Cada uno tiene ventajas y desventajas. Generalmente, se emplean dos o tres para complementar el trabajo de cada una y ayudar a asegurar una investigación completa. A continuación se detalla a grandes rasgos dicho proceso (Perdomo, Zamora Pérez, & Ramírez, 2009):

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Entrevistas: Se utilizan para recabar información en forma verbal, a través de preguntas que propone el analista. Quienes responden pueden ser gerentes, empleados o ambos, los cuales son usuarios actuales del sistema, usuarios potenciales del sistema propuesto o aquellos que proporcionarán datos y serán afectados por la aplicación. El analista puede entrevistar al personal en forma individual o en grupos. En las investigaciones de sistemas, las formas cualitativas y cuantitativas de la información son importantes. La información cualitativa está relacionada con opiniones, políticas y las descripciones cuantitativas tratan con números, frecuencia o cantidades. A menudo las entrevistas dan la mejor fuente de información cualitativa; los otros métodos tienden a ser más útiles en la recopilación de datos cuantitativos.

Cuestionarios: Proporcionan una alternativa muy útil para las entrevistas; sin embargo, existen ciertas características que pueden ser apropiadas en algunas situaciones e inapropiadas en otras. Pueden ser la única forma posible de relacionarse con un gran número de personas para conocer varios aspectos del sistema. Existen dos formas de cuestionarios para conseguir datos; los cuestionarios abiertos y cerrados, y se aplican dependiendo de si los analistas conocen de antemano todas las posibles respuestas de las preguntas o no. Con frecuencia se utilizan ambas formas en los estudios de sistemas.
Observación: El observar las operaciones le proporciona al analista hechos que no podría obtener de otra forma. Leer en relación con una actividad del negocio le brinda una dimensión de las actividades del sistema. Si bien es cierto que entrevistar al personal, ya sea directamente o a través de cuestionarios, también le ayuda y le dice algo más, ninguno de los dos métodos da una información completa. La observación por su parte proporciona información de primera mano en relación con la forma en que se llevan a cabo las actividades. Las preguntas sobre el uso de documentos, la manera en la que se realizan las tareas y si son realizados los pasos específicos como se pre-establecieron, pueden contestarse rápidamente si se observan las operaciones.

Inspección de Registros**:** El término “registro” se refiere a los manuales escritos sobre políticas, regulaciones y procedimientos de operaciones estándar que la mayoría de las empresas mantienen como guía para gerentes y empleados. Los manuales que documentan o describen las operaciones para los procesos de datos existentes, o sistemas de información que entran dentro del área de investigación, también proporcionan una visión sobre la forma en la que el negocio debería conducirse. Normalmente muestran los requerimientos y restricciones del sistema (como cantidad de transacciones o capacidad de almacenamiento de datos) y características de diseño (controles y verificación del procesamiento). Los registros permiten que los analistas se familiaricen con algunas operaciones,

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

oficinas de la compañía y relaciones formales a las que debe darse apoyo. No obstante, no muestran cómo producen las actividades, donde se ubica el poder verdadero para las decisiones, o cómo se realizan las tareas en la actualidad. Los otros métodos con objeto de encontrar datos estudiados en esta sección son más eficaces para proporcionar al analista este tipo de información, no obstante a veces esta técnica ha sido útil para ver los basamentos legales que rigen el futuro sistema, y tener presente bajo qué condiciones se deben trazar l metas.

Una vez empleadas las TRI antes descritas, deben quedar identificadas todos los procesos que lleva a cabo la organización y que se desean automatizar, así como los deseos y necesidades de los clientes.

En este trabajo se muestra una panorámica de cómo se deben enfrentar las etapas iniciales del desarrollo de un software una vez obtenidos los procesos a automatizar. Consiste en realizar una priorización de los requerimientos derivados de los mismos en cuanto a dos de sus características básicas; su complejidad y su importancia para el negocio, así como una propuesta para su realización.

Materiales y métodos

La construcción de un software difícil de usar para el cliente y que no cumple con sus expectativas es un resultado común en el desarrollo de software debido, en parte, a características intrínsecas a los usuarios que no están familiarizados con los procesos de desarrollo y al propio proceso en sí. El problema está dado por la necesidad de diseñar un proceso flexible que tenga como objetivo mitigar los problemas planteados en la introducción sin dejar de cumplir con las exigencias propias del desarrollo.

La propuesta está basada en la experiencia adquirida en el paso por varios proyectos informáticos y la recopilación de datos históricos sobre procesos de desarrollo de software. Para ello se evaluaron los resultados de la utilización de procesos tradicionales y ágiles midiendo los tiempos de implementación de las iteraciones así como la tardanza de la implantación del sistema identificando sus causas. Asimismo se aplicaron encuestas a desarrolladores y especialistas en recopilación de información para detectar los problemas a los que se han enfrentado a lo largo del proceso y las consecuencias que han traído en el mismo en términos de tiempo y desvío de las necesidades del cliente. También fueron evaluados los niveles de aceptación de los distintos productos en cuanto a cumplimiento de las expectativas de los clientes así como de los términos establecidos de tiempo con el fin de obtener resultados desde los puntos de vista de todos los involucrados.

CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

Resultados y discusión

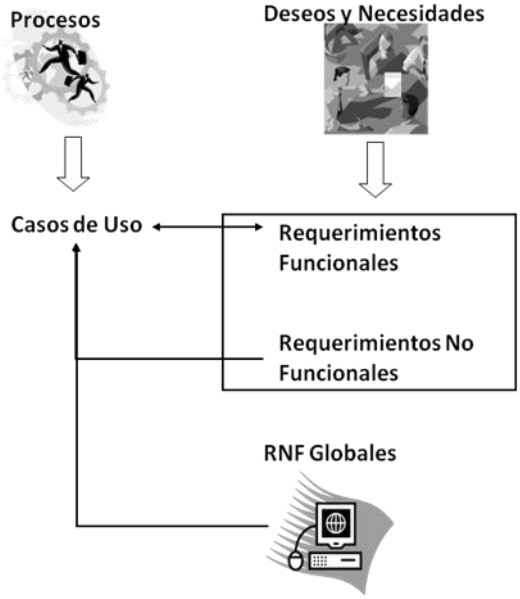

Figura 1: Proceso Requerimientos-Casos de Uso.

Como se muestra en la figura 1, a partir de los procesos identificados surgen los casos de uso; a partir de los deseos y necesidades de los clientes surgen los requerimientos funcionales y no funcionales y se tienen además requerimientos no funcionales globales que responden a las características generales de la aplicación a construir.

En dicha figura se muestra la relación bidireccional entre los casos de uso y los requerimientos, pues se pueden derivar unos de otros en dependencia del que se esté tomando como punto de partida. Por ejemplo, a partir de los casos de uso surgidos de la identificación de los procesos, se van a derivar los distintos requerimientos funcionales que tendrá la aplicación; y de los deseos y necesidades de los clientes, se generan nuevos casos de uso o se insertan nuevas funcionalidades en los ya identificados.

En cuanto a los requerimientos no funcionales tanto específicos como globales, se insertan en los casos de uso teniendo en cuenta cuáles son los que responden a dichos requerimientos.

Lo que se explica es que existe un proceso de retroalimentación a partir de la información obtenida después de la aplicación de las TRI, así como de la identificación tanto de casos de uso como de requerimientos del software, por lo cual no debe existir una restricción en cuanto a qué artefacto genera..

Lo que se propone es realizar una priorización los requerimientos obtenidos a partir de la aplicación de las TRI en cuanto a dos de sus características básicas; su complejidad y su importancia para el negocio. Una vez realizada dicha labor se identifican las iteraciones a realizar, las cuales preferiblemente deben ser lo más cortas posibles dependiendo del tamaño del proyecto, para que una vez obtenida una parte funcional del software el cliente pueda validar el cumplimiento del mismo con sus necesidades reales. Al comienzo de cada iteración se especificará más detalladamente cada uno de los requerimientos incluidos en la misma, lo que posibilitará una interacción bastante frecuente con los clientes, y por lo tanto una

CIGET - MATANZAS
IDICT
INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

descripción lo suficientemente actualizada como para cumplir en el mayor porciento posible las necesidades de los mismos.

Está demostrado que este enfoque funciona por varias razones:

1. Aún puede satisfacer las necesidades de planificación: Al identificar los requisitos de alto nivel en la etapa inicial se tiene suficiente información para producir una primera estimación de costos y el calendario.
2. Se minimiza el desperdicio: Permite concentrarse sólo en los aspectos del sistema que se va a construir.
3. Los clientes darán mejores respuestas: Similarmente, los clientes tendrán un mejor entendimiento del sistema que se está construyendo ya que ellos también se irán retroalimentando al recibir entregables continuamente.

Inicialmente en los proyectos no se empleaba este enfoque ágil, ya que simplemente se guiaban por uno tradicional, y por lo general ocurría que no había prácticamente ninguna retroalimentación con el cliente hasta pasado mucho tiempo, por lo cual si este no se percataba de algún requerimiento (como era generalmente el caso), no habían muchas opciones para resolverlo. En la Figura 2 se muestran las diferencias reales en cuanto a 2 de los proyectos que se han desarrollado, en el primer caso utilizando el enfoque tradicional, y en el segundo uno más ágil.

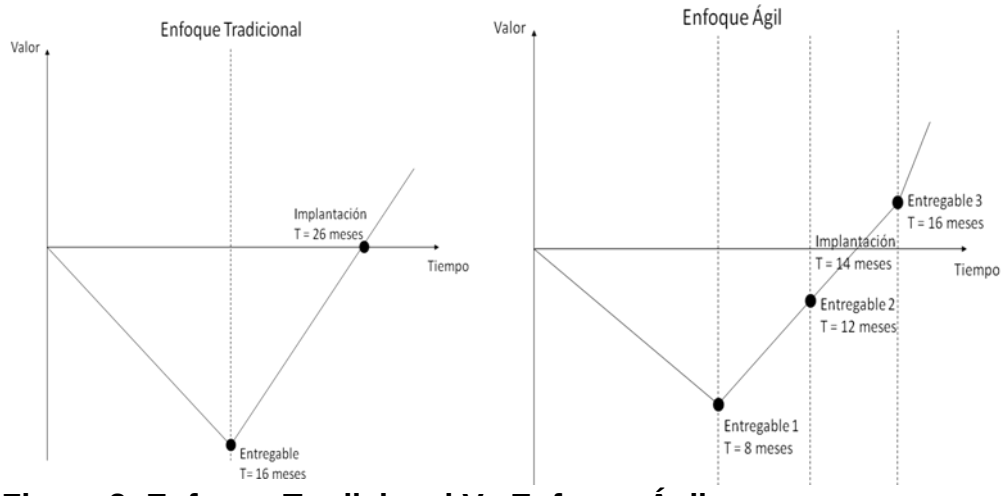

Figura 2: Enfoque Tradicional Vs Enfoque Ágil.

Como se observa, con el enfoque tradicional el sistema quedó implantado en su totalidad después de 26 meses de desarrollo, mientras que utilizando el ágil lo hizo en sólo 14 meses. Con dicho enfoque ágil, no solo se logró gastar menos dinero al comienzo, sino que además se tuvo una retroalimentación y por ende el beneficio

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

de tener un software que funciona ya siendo probado por el cliente. Además, como con un desarrollo ágil se van priorizando los requerimientos de acuerdo a la importancia para el usuario, se mantiene al cliente interesado en lo que se está haciendo y va constantemente expresando qué tan cerca está lo que se está produciendo de sus necesidades reales.

Un ejemplo de cómo distribuir los requerimientos en las distintas iteraciones se muestra en la tabla 1.

		Iteración		
Importancia para el negocio	Alta	2	1	1
	Media	3	3	2
	Baja	4	4	3
		Alta	Media	Baja
		Complejidad		

Tabla 1: Priorización de requerimientos por iteraciones.

En la distribución anterior se muestra un proceso de desarrollo de 4 iteraciones, pero realmente en lo que se debe hacer énfasis es en la forma de priorizar los requerimientos. Es importante valorar la importancia para el negocio por encima de la complejidad, siempre teniendo en cuenta implementar aquellos requerimientos de baja complejidad primero para garantizar el rápido cumplimiento de las necesidades del cliente.

Se debe tener total dominio del alcance de los casos de uso que se vayan a implementar en cada iteración tratando de no mantener casos de uso muy extensos, pues en lugar de ayudar, por lo general lo que hacen es salirse del alcance del mismo, imposibilitando su entendimiento, ejemplo de esto es sólo emplear el patrón CRUD en casos donde no existan demasiadas transacciones, o las mismas sean muy complejas, ya que con casos de uso como estos se pierde la forma de medir el avance del cronograma, pues debido a su tamaño y envergadura, suele tardarse su implementación completa.

A la par de todo el proceso descrito anteriormente, el analista debe ir identificando los requerimientos de calidad, los cuales servirán como punto de partida para la confección de los casos de prueba. Con dichos requerimientos se podrá velar de

CIGET - MATANZAS IDICT INFORMACION Y GESTION TECNOLOGICA
CITMA
ISSN 1029-3450
REVISTA AVANZADA CIENTÍFICA
RNPS 1843

una forma más rigurosa y precisa por el cumplimiento de la calidad del software y no solo concentrarse en revisar contra listas de chequeo.

Conclusiones

La ingeniería de requerimientos es un proceso sumamente delicado, teniendo en cuenta que sienta las bases para la construcción del software. Una mala captura de requisitos provoca resultados desastrosos a una escala mayor que un error en cualquier otra disciplina de desarrollo del software. La guía de pasos propuesta ayuda a enfocar el trabajo de esta etapa hacia las necesidades esenciales para cada momento del desarrollo. Dichos pasos han sido ya puestos en práctica y los resultados obtenidos por diferentes proyectos demuestran su validez.

No es una guía rígida, sino una forma de realizar una ingeniería de requerimientos que minimice los riesgos inherentes al desarrollo del software, así como un enfoque ágil en dicho proceso.

Referencias bibliográficas

Kendall. (2009). Análisis y Diseño de Sistemas. Pearson Education.
Perdomo, E., Zamora C. L., & Ramírez, J. E. (2009). La Recolección de Información. Neiva: Regional Huila.

Bibliografía consultada

Pressman, Roger S. (2002). Ingeniería de Software. Un enfoque práctico. Quinta edición. McGraw-Hill. Madrid. 2002.
Henrik K. (2007). Scrum and Xp from the trenches. Free Online Edition.
Jacobson, I., Rumbaugh, J., Booch, G., El Proceso Unificado de Desarrollo. Addison Wesley.
Gallagher, Brian P. (1999). Software Acquisition Risk Management Key Process Area (KPA) — A Guidebook, Version 1.02. Carnegie Mellon University. Handbook Cmu/Sei-99-Hb-001.
Roppponen, J., Lyytinen, K. (2000). Components of Software Development Risk: Hot to address Them? IEEE transactions on software engineering, 26(2).

Fecha de recepción: 16/04/2011 Fecha de aprobación: 6/06/2012