

Herramientas CASE para ingeniería de Requisitos

Por: **ALARCÓN, Andrea¹**
SANDOVAL, Erika²

Resumen

En este artículo se presentan apartes del resultado de la investigación sobre algunas herramientas CASE que apoyan las tareas de la Ingeniería de Requisitos (IR). Dicha fase esta contemplada en el proceso de Ingeniería de Software para el desarrollo de proyectos informáticos. Además se pretende analizar, sistematizar e integrar los resultados del estudio con el fin de determinar las principales características, ventajas y desventajas de una herramienta CASE para la IR.

Palabras clave: Requisito, Ingeniería de Requisitos, calidad, Modelo de evaluación, Ingeniería de software asistida por computador (CASE)

Abstract

This article shows a review of the information from some of the CASE (Computer Assisted Software Engineering) tools that support the work of Requirements Engineering (RE) as a phase included in the process of Software Engineering for computer projects development. The objective is to identify the main characteristics, advantages and disadvantages of the studied tools and have a guide to select, acquire and implement a CASE tool for RE.

Keywords: Requirement, Requirements Engineering, Quality, Evaluation Model, Computer Assisted Software Engineering (CASE)

¹M.Sc. en Software libre, Universidad Oberta de Cataluña, España. Esp. en ingeniería de Software. Ing. de Sistemas y Computación. Docente JDC e-mail: acalarcon@gmail.com

²M.Sc. en Software libre, Universidad Oberta de Cataluña, España. Ing. de Sistemas y Computación. Esp. en Gerencia de Sistemas Informáticos. Docente JDC e-mail: emsandoval@gmail.com

INTRODUCCIÓN

El desarrollo de software ha ocupado un lugar importante en la Ingeniería, pero al igual que otras disciplinas, aún presenta fallas. Debido a esto se han planteado técnicas y métodos para minimizar los problemas identificados en la *crisis del software*. Es así como surge la Ingeniería de Software, presentando distintos modelos de procesos que se ajustan a las necesidades y proyectos requeridos. La mayoría de ellos involucran en sus fases iniciales tareas como planeación, levantamiento de información, determinación de las características que debe cumplir el software, agrupadas en lo que hoy se conoce como Ingeniería de Requisitos (IR). Esta fase ocupa un lugar importante en el proceso de desarrollo de software ya que si el personal comprometido no conoce con claridad los requisitos, corre el riesgo de que los resultados obtenidos no sean los esperados, presentando así los mismos problemas de hace cincuenta años: altos costos, baja calidad de software, clientes inconformes e incumplimiento de plazos, entre otros.

Con el ánimo de facilitar las tareas del desarrollo de software, surgen herramientas informáticas que agilizan la labor en la IR. Dichas herramientas son denominadas CASE (Ingeniería de software asistida por computador), y sirven de apoyo para los desarrolladores, desde el principio hasta el final del proceso. Para el caso particular de esta investigación, son de especial interés aquellos instrumentos que se encargan de actividades como: extraer, analizar, documentar, revisar, negociar y validar los requisitos del sistema objeto de estudio.

IMPACTO DE LA INGENIERÍA DE REQUISITOS EN EL DESARROLLO DE PROYECTOS INFORMÁTICOS

En el ambiente informático es crucial la época en la cual el hardware era de mayor tamaño, más costoso y más importante que el software; aunque con el transcurrir del tiempo, éste último ocupa una mejor posición, dando cabida a la comercialización de los primeros ordenadores y al aumento en la demanda de un software un poco más complejo. Se creería que estas son buenas noticias, pero en realidad, tal avance trae consigo la crisis del software; expresión que se utilizó por primera vez en la conferencia organizada por la Comisión de Ciencia de la OTAN en Garmisch, Alemania, en octubre de 1968, y tiene como objeto agrupar la gran cantidad de problemas que elevan el índice de fracasos en los proyectos de desarrollo (Figura 1).



Figura 1. ¿Por qué Fracasan los proyectos? Tomado de *Compendio de Ingeniería del Software*. Junio 2006. Juan Palacio Bañeres

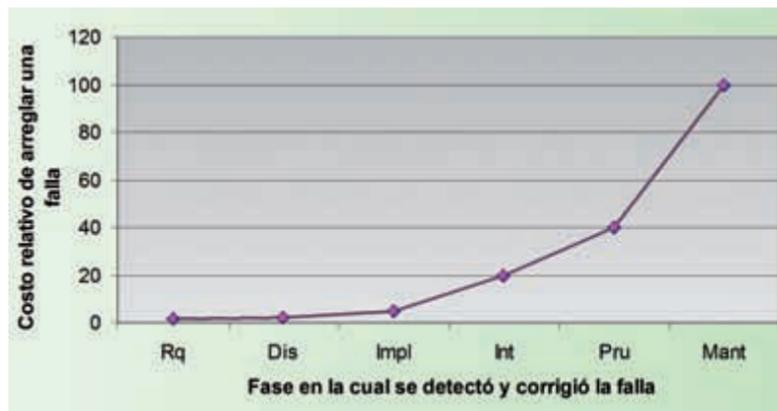
La primera opción para solucionar esta larga lista de problemas, era sencilla, aplicar los métodos de la ingeniería tradicional, a la manufactura de software. Pero bien es sabido que el software se desarrolla o construye, más no se manufactura como sucede con el hardware, un puente o una casa. Entonces, la solución era aplicar al software métodos, técnicas y herramientas de una Ingeniería de desarrollo.

Dicho proceso reúne un conjunto de etapas o pasos ordenados, con el fin de lograr un objetivo; en este caso, un producto software de calidad. Cuando se habla del proceso de desarrollo de software se enfatiza en las necesidades de los usuarios, traducidas en requisitos de software, y estos a su vez son transformados en diseño directamente convertido en la implementación del código, debidamente probado, documentado y certificado para su uso operativo.

En teoría, la Ingeniería de Software debería funcionar en la obtención de un producto de calidad,

pero después de años de experiencias se siguen presentando los problemas de la Crisis del software; es decir, altos costos, baja calidad, clientes inconformes e incumplimiento de plazos, entre otros. Algunos estudios han demostrado que la mayoría de fallencias tienen su origen en las primeras etapas de la ingeniería de software, fases en las cuales el trabajo está orientado a la obtención de información para entender qué es lo que el cliente desea. Es entonces cuando se decide emplear más tiempo y dedicación a dichas tareas y aplicar un enfoque sistemático y disciplinado cuyo resultado es la aplicación la Ingeniería de Requisitos.

La ingeniería de requisitos ha adquirido una especial importancia desde mediados de los años 70, hasta hoy, cuando se le considera una etapa clave en el desarrollo del software, ya que al tener claros los requisitos del sistema, el desarrollo del mismo se convierte en una tarea organizada que garantiza resultados exitosos. Además, como se observa en la Figura 2, mientras más pronto se encuentren y solucionen errores en el proceso, menores son los costos de los mismos. Tradicionalmente la Ingeniería de Requisitos, se ha entendido como una parte borrosa del ciclo de vida del software, de la que se obtiene una especificación formal de unas ideas informales. La IR es en esencia la aplicación de principios, métodos, técnicas y herramientas en pro del descubrimiento de los requisitos de un producto software; de igual manera, permite el análisis y documentación de los objetivos, funciones y restricciones de dicho sistema de computación, pero presenta una falencia: no hay acuerdo sobre lenguajes, métodos y herramientas, para su ejecución.



Rq: Requisitos y especificaciones
Dis: Diseño
Impl: Implementación
Int: Integración
Pru: Pruebas de aceptación
Mant: Mantenimiento

Figura 2. Costo relativo del arreglo de una falla en cada fase del ciclo de vida del software tradicional. Basado en Schach, S. (2006). Ingeniería de Software clásica y orientada a objetos.

HERRAMIENTAS CASE, HACIA UNA INGENIERÍA DE REQUISITOS COMPUTARIZADA

A medida que pasa el tiempo se logra entender que el empleo del software es una buena opción para agilizar y sistematizar las tareas en el desarrollo de procesos. El desarrollo de software no es la excepción; en este caso dichas herramientas se han denominado CASE (Ingeniería De Software Asistida Por Computador). Estas incluyen un conjunto de programas que facilitan la optimización de un producto ofreciendo apoyo permanente a los analistas, ingenieros de software y desarrolladores.

CASE es la aplicación de métodos y técnicas que dan utilidades a los programas, por medio de otros, procedimientos y su respectiva documentación. En esta investigación se hace referencia a las herramientas que ayudan a la gestión de requisitos; es decir al proceso de identificación, asignación y seguimiento de los mismos, incluyendo interfaz, verificación, modificación y control de cada requisito, durante el ciclo de vida del proyecto. Los cambios/actualizaciones de requisitos deben ser gestionados para asegurar que se mantenga la calidad del producto.

Hasta hace poco tiempo las herramientas para la gestión de requisitos de software se limitaban a editores de texto, los cuales hacían de esta tarea una labor tediosa y confusa. Actualmente, se cuenta con múltiples opciones, como las que se mencionan a continuación:

IRQA 4³

Herramienta CASE de Ingeniería de Requisitos, diseñada para soportar las actividades realizadas en el proceso de especificación de sistemas. Ésta facilita y formaliza la comunicación entre el cliente, el proveedor y los distintos miembros del equipo de desarrollo. Facilita la captura, organización y análisis de las condiciones, así como la especificación de la solución mediante el apoyo metodológico adaptable a cada cliente.

RETO

Esta herramienta propone un modelo de requisitos para capturar los aspectos funcionales del sistema; básicamente, mediante tres técnicas complementarias entre sí: la definición de la Misión del Sistema, la construcción del Árbol de Refinamiento de Funciones y el desarrollo del Modelo de Casos de Uso. Además, se introduce un Proceso de Análisis que permite traducir el Modelo de Requisitos en el Mo-

³Visure Solutions, The IRQA Company. <http://www.visuresolutions.com>



delo Conceptual, manteniendo la trazabilidad entre ambos y propiciando una representación de la información en el segundo prototipo.

CONTROLA

Herramienta de apoyo al proceso de ingeniería de software en pequeñas empresas. Se creó gracias a la expansión que tuvo el mercado y a la generación de grandes y pequeñas empresas, las cuales requieren un instrumento para el desarrollo de sus proyectos. Ofrece recursos importantes tales como: Administración de requisitos, administración de casos de uso, administración de casos de prueba y error, planeamiento de liberaciones, administración de implementaciones, control de dependencia entre Implementaciones, matriz de rastreabilidad y rastreabilidad de los requisitos.

OSRMT (Open Source Requirements Management Tool)⁴

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.

JEREMIA⁵

Se trata exclusivamente de una aplicación cliente exclusivamente, lo cual no permite la posibilidad de trabajar en equipo. Ésta, ayuda durante el desarro-

llo del sistema, especialmente en el seguimiento de cambios de los requisitos a lo largo del ciclo de vida. Con JEREMIA es posible captar las necesidades, analizarlas y clasificarlas. Implementa un módulo orientado a la generación de la documentación posible de exportar en formato DocBook XML, la cual junto con los requisitos, se almacena en una base de datos en MySQL.

RAMBUTAN⁶

Esta herramienta está basada en XML, realmente consta de un conjunto de aplicaciones para el usuario final, ayudando a los analistas de sistemas en la recopilación y categorización de hechos en un documento de especificación de requisitos. Lo curioso es que tiene un cliente para palm (PDA), el cual se utiliza para recopilar los hechos en el lugar donde está ubicado el cliente mientras que la aplicación de escritorio recibe la información, edita y perfecciona. Ambas aplicaciones permiten al usuario introducir, modificar y visualizar los datos que componen un documento de especificación de requisitos.

Comparada con otras herramientas de gestión de requisitos, Rambutan ofrece las siguientes ventajas competitivas: Aplicación cliente para palm (PDA-class), portabilidad entre plataformas, es independiente de cualquier metodología de especificación de requisitos, y permite distribución libre.

Existen otras herramientas en estudios para la gestión de requisitos. A continuación se mencionan, algunas de las incluidas en el estudio comparativo presentado por El Consejo Internacional sobre la Ingeniería de Sistemas (INCOSE)⁷: CaliberRM, REM, SMART TRACE, SoftREQ, Analyst Real Team System

⁴Open Source Requirements Management Tool. <http://sourceforge.net/projects/osrmt>

⁵Jeremia, sistema de Gestión de Requisitos. <http://jeremia.sourceforge.net/>

⁶Arcle Technologies. <http://rambutan.sourceforge.net/>

⁷INCOSE, The International Council on Systems Engineering Requirements Management Tools Survey. <http://www.incose.org>



El desarrollo de software no es la excepción; en este caso dichas herramientas se han denominado CASE (Ingeniería De Software Asistida Por Computador).



La ingeniería de requisitos es una tarea que aún tiene mucho por explorar para optimizar sus tareas y cumplir a cabalidad los objetivos propuestos.



(ARTS), CARE 3.2, CORE 5.1, Cradle 5.2, Envision VIP, Gatherspace, IBM Rational RequisitePro, KollabNet Editor 2005, PACE, RaQuest 3.0, RMTrak, RTM, SLATE REquire 6.5, SoftREQ, UGS Teamcenter 2005, true-req product desktop, XTie-RT, Specification Analysis Tool (SAT), ECM, Banyan2.2, Contour, Projectcity 3.5, FeaturePlan 2.6, analyst pro, ChangeWare 2.0, aligned elements, Dassault Systemes CSE 4.0, Polaron ALM for Subversion 3.0, Telelogic DOORS, Accept 360.

CONCLUSIONES

La Ingeniería de requisitos no es la solución definitiva a los inconvenientes y/o problemas presentados en la crisis del software, pero ayuda en gran medida al descubrimiento y solución de falencias en etapas tempranas del desarrollo de proyectos software, reduciendo costos y tiempo en el ciclo de vida.

Las herramientas CASE agilizan y facilitan la opti-

mización de un producto software, ofreciendo apoyo permanente al grupo de desarrollo.

En el mercado existen herramientas CASE de apoyo a las diversas fases del proceso de desarrollo de software. Algunas, atadas a una metodología específica, otras totalmente independientes de la misma. En esta investigación se estudiaron instrumentos de soporte a la fase de ingeniería de requisitos. Contrario a lo que se creía al inicio del estudio, son bastantes; la mayoría de ellas son comerciales y presentan mayor funcionalidad, aunque debido a los altos costos de sus licencias son de difícil y/o limitado acceso.

La ingeniería de requisitos es una tarea que aún tiene mucho por explorar para optimizar sus tareas y cumplir a cabalidad los objetivos propuestos. Igualmente, es necesario realizar una evaluación de funcionalidad y rendimiento de las herramientas existentes, con el fin de depurarlas, ya que al aumentar su número se hace más difícil la elección para la gestión de recursos.

BIBLIOGRAFÍA

- Arcle Technologies. <http://rambutan.sourceforge.net/>
- BOOCH, G., JACOBSON, I. y RUMBAUGH, J. 2006. El Lenguaje Unificado de Modelado, Segunda edición. Madrid: Pearson Educación.
2000. El Proceso Unificado de Desarrollo de Software. Primera edición. México: Pearson Educación.
- CALLEJAS, M. 2005. La Ingeniería de Software Libre y sus Herramientas Aplicadas a Proyectos Informáticos. <http://www.itba.edu.ar/capis/rtis/Art%EDculos.html>
- DÁVILA, M. 2007. Ingeniería del Software Libre. Revista ACIS. Edición N° 102: Ingeniería de Software. www.acis.org.co/fileadmin/Revista_102/uno.pdf.
- GINESTÁ, M. y GONZÁLEZ A. 2005. Ingeniería de software en entornos de software libre. Primera edición. Barcelona: UOC. <http://www.incose.org>.
- JEREMIA, < <http://jeremia.sourceforge.net/> >
- LAUESEN, S. 2002. Software Requirements: Styles and Techniques. Addison-Wesley.
- Open Source Requirements Management Tool. [Consultado Junio de 2008]. Disponible en < <http://sourceforge.net/projects/os-rmt> >
- PALACIO, B. 2006. Compendio de Ingeniería del Software. [Versión electrónica]. Recuperado de http://www.navegapolis.net/files/cis/CIS_1_05.ppt
- PFLEEGER, S. 2002. Ingeniería de Software: Teoría y práctica. Primera edición. Brasil: Prentice Hall.
- PRESSMAN, R. 2006. Ingeniería del Software: Un enfoque práctico. Sexta edición. México: McGraw Hill.
- SCHACH, S. 2006. Ingeniería de Software clásica y orientada a objetos. Sexta edición. México: McGraw Hill.
- Visure Solutions, The IRqA Company. [Consultado Junio de 2008]. Disponible en < <http://www.visuresolutions.com> >
- VITTA F. G., 2005. El Bazar. Lecciones sobre ingeniería de software desde el Software Libre. [Versión electrónica]. Recuperado de www.csol.org/cc60v/2005/InformesAlumnos/Inf-gvitta-bazar.pdf

Por: **CASTILLO, Dilia**¹
ROMERO, John²
CHAVES, Carlos³

¿Cómo planear financiera y tributariamente una pyme?

Resumen

Las Pyme son las unidades económicas más pequeñas, y en ellas germinan las grandes ideas empresariales. Nacen en el núcleo familiar y van creciendo a medida que el mercado demanda sus productos o servicios. En este nivel, su capital de trabajo es igual a los aportes del dueño o dueños junto con la escasa inversión en tecnología, investigación y desarrollo, y capital intelectual entre otras. En este trabajo se han recopilado algunas de las prácticas para la implementación de la cadena valor que permita el crecimiento de las Pyme. La planeación financiera y tributaria se presenta dividida en tres etapas teniendo en cuenta: el tamaño, el valor del activo y el número de empleados. De igual manera, es subdividido en los siguientes subtemas de proyección: aporte social, constitución de la empresa, objetivo de creación, estructura de capital, los ingresos, los costos y gastos, los impuestos como gasto, los impuestos como beneficio fiscal, herramientas financieras para la permanente evaluación, la toma de decisiones y continuidad de la empresa, evitando así el cierre definitivo.

Palabras clave: Proyección de la gestión, prospectiva financiera, pequeñas empresas, mejores prácticas para la pequeña empresa

Abstract

The PYMES are the smallest economic units where the greatest ideas are germinated. They are born in the familiar nucleus and grow as the market demands their products or services. At this level the company's working capital is equal to the owner (s) contributions, different from the investment in technology, research, development and intellectual capital, among others. In this work there have been implemented some of the practices to follow in the implementation of the value change for the Pyme's growth. Financial and tax planning is divided into three stages, taking into account the size according to the asset value and the number of employees. In the same way, it is subdivided into the following projection subthemes: social contribution, company constitution, creation objective, capital structure, income, costs and expenses, taxes as expenses, taxes as fiscal benefits, financial tools for continuous evaluation, as a tool for decision-making and continuity of the company, avoiding its definitive closure.

Keywords: Management Projection, financial prospective, small businesses, best practices for small business.

¹M.Sc. en Administración con énfasis en Finanzas, UNAB-ITESM, Esp. en Revisoría Fiscal de Universidad Jorge Tadeo Lozano, Contador Público de la UPTC. Asesor Contable JDC. Docente Contaduría Pública UPTC. e-mail: dilcast@hotmail.com

²M.Sc. (c) en Derecho con Énfasis en Tributación, Universidad Externado de Colombia. Esp. en Gerencia Tributaria, Universidad la Gran Colombia, Contador Público Universidad Nacional de Colombia. Docente UPTC. e-mail: tayro_jh@yahoo.com

³M.Sc. en Administración con énfasis en Finanzas UNAB-ITESM. Esp. en Gerencia de Mercados de la UNIBOYACA, Esp. en Administración de la Escuela de Administración de Negocios y Administrador de Empresas, UPTC. Docente UPTC de Chiquinquirá. e-mail: carloschavesm@gmail.com