

Revisión de herramientas para la gestión de requisitos: un aporte al aseguramiento de la calidad del software*¹

Review of tools for the management of requirements: a contribution to the Software Quality Assurance

Revisão de ferramentas para o gerenciamento de requisitos: uma contribuição para a garantia da qualidade do software

LUIS EDUARDO PELÁEZ VALENCIA², ALONSO TORO LAZO³, LUIS DAVID LEDESMA MARIÑO⁴, DANIEL EDUARDO RODRÍGUEZ FRANCO³, EDIER HUMBERTO LARGO BUENO³

Recibo: 31.01.2018 – Aprobación: 29.08.2018

* **Modelo para la citación de este artículo / Template for citation of this article / Modelo para a citação deste artigo:** PELÁEZ VALENCIA, Luis Eduardo; TORO LAZO, Alonso; LEDESMA MARIÑO, Luis David; RODRÍGUEZ FRANCO, Daniel Eduardo & LARGO BUENO, Edier Humberto (2018). Revisión de herramientas para la gestión de requisitos: un aporte al aseguramiento de la calidad del software. En: Ventana Informática No. 38 (ene-jun). Manizales (Colombia): Facultad de Ciencias e Ingeniería, Universidad de Manizales. p. 149-164. ISSN: 0123-9678. DOI: <https://doi.org/10.30554/ventanainform.38.2865.2018>

- 1 Artículo de investigación / Research article / Artigo de pesquisa
Proyecto / Project / Projeto: Modelo Automatizado para el aseguramiento de la calidad de los requerimientos de software / Automated Model for the assurance of the quality of software requirements / Modelo Automatizado para a garantia da qualidade dos requisitos de software
Institución / Institution / Instituição: Universidad Católica de Pereira (Pereira, Risaralda, Colombia).
Versión ampliada y ajustada de la ponencia presentada en el III Congreso Andino de Computación, Informática y Educación, CACIED 2017 [05-08/11/2017, San Juan de Pasto (Colombia): Universidad de Nariño] / Extended and adjusted version of the paper presented at the III Andean Congress of Computing, Information Technology and Education, CACIED 2017 [05-08/11/2017, San Juan de Pasto (Colombia): University of Nariño]. / Versão ampliada e ajustada do artigo apresentado no III Congresso Andino de Computação, Tecnologia da Informação e Educação, CACIED 2017 [05-08/11/2017, San Juan de Pasto (Colômbia): Universidade de Nariño].
- 2 Master en Ingeniería de software: Gestión y Desarrollo de Proyectos de Software, Magister en Ingeniería de Software / Master in Software Engineering: Management and Development of Software Projects, Master in Software Engineering / Mestrado em Engenharia de Software: Gestão e Desenvolvimento de Projetos de Software, Mestrado em Engenharia de Software. Profesor asociado / Associate Professor / Professor Associado, Universidad Católica de Pereira (Pereira, Risaralda, Colombia). luis.pelaez@ucp.edu.co. <https://orcid.org/0000-0002-4836-8336>
- 3 Master en Gestión y Desarrollo de Proyectos de Software / Master in Management and Development of Software Projects / Mestre em Gestão e Desenvolvimento de Projetos de Software. Docente y Director programa de Ingeniería de Sistemas y Telecomunicaciones / Professor and Director of the Systems and Telecommunications Engineering Program / Professor e diretor do programa de Engenharia de Sistemas e Telecomunicações, Universidad Católica de Pereira (Pereira, Risaralda, Colombia). alonso.toro@ucp.edu.co. <https://orcid.org/0000-0001-7593-8026>
- 4 Estudiantes de ingeniería de Sistemas y Telecomunicaciones / Students of Systems and Telecommunications Engineering / Estudantes de Engenharia de Sistemas e Telecomunicações, Universidad Católica de Pereira (Pereira, Risaralda, Colombia). luis.ledesma, daniel1.rodriguez, edier.largo@ucp.edu.co.

Resumen: *Dado que en un proyecto de software se puede llegar a identificar un considerable número de requisitos, el hacer uso de software especializado que agilice y facilite el trabajo con los mismos puede evitar inconvenientes al momento de gestionarlos. El no hacerlo, con frecuencia genera consecuencias negativas en el proyecto tales como retrasos en las entregas, aumento de los costos y difícil trazabilidad. Lo anterior, permite comprender la importancia de las herramientas para la gestión de requisitos en el desarrollo de proyectos software y su aporte en el aseguramiento de la calidad. El objetivo de este artículo es dar a conocer una revisión sistemática de herramientas empleadas para la gestión de requisitos que permita establecer las principales características de un modelo automatizado para el Aseguramiento de la Calidad de los Requisitos, trabajo futuro de la investigación. Finalmente, se presentan los resultados y principales conclusiones obtenidas, entre las que se destaca la predominancia en el uso de herramientas libres por las Mipymes desarrolladoras, debido a los costos de licenciamiento.*

Palabras clave: Herramientas, Gestión de requisitos, Aseguramiento de la calidad de los requisitos.

Abstract: *Since a project can identify a large number of requirements, make use of specialized software that speeds up and facilitates the work with them can avoid inconveniences when managing them. Failure to do so often result in negative project consequences such as delays in deliveries, increased costs and difficult traceability. The above, allows understanding the importance of the tools for the management of requirements in the development of software projects and their contribution in the quality assurance. The objective of this article is to present a systematic review of tools used for the management of requirements to establish the main characteristics of an automated model for Requirements Quality Assurance, future work of research. Finally, the results and main conclusions are presented, among which the predominance in the use of free tools by the development is SMEs, due to licensing costs.*

Keywords: Tools, Requirement management, Requirements Quality Assurance.

Resumo: *Dado que em um projeto de software é possível identificar um número considerável de requisitos, o uso de software especializado que acelera e facilita o trabalho com eles pode evitar*

inconvenientes ao gerenciá-los. A falta de fazê-lo muitas vezes resulta em consequências negativas do projeto, como atrasos nas entregas, aumento dos custos e rastreabilidade difícil. O acima permite compreender a importância das ferramentas para o gerenciamento de requisitos no desenvolvimento de projetos de software e sua contribuição na garantia de qualidade. O objetivo deste artigo é apresentar uma revisão sistemática das ferramentas utilizadas para o gerenciamento de requisitos para estabelecer as principais características de um modelo automatizado para Garantia da Qualidade dos Requisitos, trabalho futuro de pesquisa. Finalmente, os resultados e principais conclusões são apresentados, entre os quais predomina a utilização de ferramentas gratuitas pelas PMEs de desenvolvimento, devido aos custos de licenciamento.

Palavras-chave: Ferramentas, Gerenciamento de Requisitos, Garantia da qualidade de requisitos.

Introducción

Una problemática de la industria del software es el deficiente proceso de desarrollo y gestión de requisitos que se aplica y su efecto en el producto y su calidad. Esta industria representa un papel significativo debido a que el software -como producto intangible- educa, genera empleo, brinda bienestar, satisface necesidades y, especialmente, contribuye al desarrollo tecnológico e integral de la sociedad. Como lo indica el Ministerio de Comercio, Industria y Turismo de Colombia, citado por Lopera (2012, 3), «*es un gran componente de conocimiento, por lo que requiere un alto desempeño en investigación, desarrollo tecnológico y formación de personas capaces de producir conocimiento y soluciones acordes a las necesidades de todo tipo de organizaciones que lo requieran*».

Para Torres (2009, 5), en la fase de Requisitos se genera el 56% del total de errores encontrados en los productos software (Figura 1), a la cual, según Yang et al. (2008, 61), Se le destina menos del 20% del esfuerzo del proyecto, independientemente del modelo de desarrollo (iterativo, en cascada, etc.) o del tipo de proyecto (nuevo, de mantenimiento). Del mismo modo, Cristiá (2011, 6), citando un estudio de *Standish Group*, relaciona las principales causas de fracaso en los proyectos de software: «*Requerimientos incompletos (13,1 %), Falta de compromiso del usuario (12,4 %), Falta de recursos (10,6 %), Expectativas no realistas (9,9 %), Falta de soporte ejecutivo (9,3 %), Requerimientos y especificaciones*

cambiantes (8,7 %), Falta de planeamiento (8,1 %), Fin de la necesidad del sistema (7,5 %)». En coherencia con ello, Toro & Gálvez (2017, 7), afirman: «Al ser la parte inicial de un proyecto de desarrollo de software, los requisitos tienden a ser la parte más incomprendida de la Ingeniería de Software y, sin embargo, es la más crucial. Gran parte de las fallas en los proyectos de software se deben a una pobre definición de requisitos, encontrándose dentro de las principales razones más comunes de fracaso de dichos proyectos».

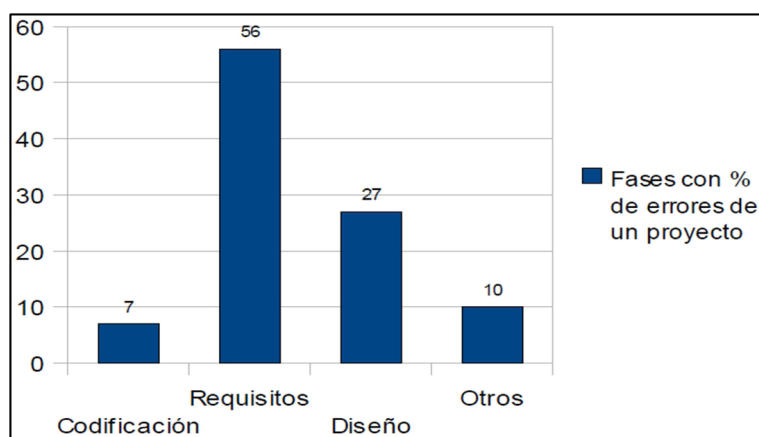


Figura 1. Proporción de errores en cada una de las fases de desarrollo (Torres, 2009, 5).

El costo de una adecuada recolección de requisitos y el análisis del sistema es menor al ocasionado por la obligación posterior de reparar productos deficientes y de baja calidad, que con frecuencia lleva al fracaso del proyecto, lo cual corrobora que «*El fundamento básico de cualquier software recae sobre su proceso de ingeniería de requisitos*»; *que el éxito o fallo del software depende casi siempre de qué tan bien se hayan capturado, entendido y usado los requisitos como base para el desarrollo; y que la Ingeniería de Requisitos es la fase de un proyecto software donde se definen las propiedades y la estructura del mismo, comprendiendo a la vez el desarrollo y gestión de requisitos*» (Inteco, citado por Toro & Peláez, 2016, 119).

Para lograr un producto software de alta calidad, los requisitos deben ser completos, correctos, realizables y verificables, señalan Wieggers y Beatty, citados por Salazar et al. (2012, 83), condiciones que en ocasiones no se cumplen. «*Por esto, la gestión de requisitos en la*

Ingeniería de Software se ha convertido en una de las principales estrategias para garantizar desde las primeras etapas del proceso de desarrollo de software, la calidad de las aplicaciones resultantes» (García y Puello, citados por Toro & Gálvez, 2017, 1), cobrando importancia las herramientas para el tratamiento, administración y desarrollo de requerimientos, pues *«precisarlos es una parte esencial de la fórmula para los proyectos de software exitosos»* (McEwen, citado por Camacho, 2005, 15).

«En la mayoría de las prácticas, los requisitos de software se especifican utilizando un lenguaje natural en particular debido a su flexibilidad y simplicidad. Sin embargo, dicho lenguaje natural con frecuencia hace que la especificación sea propensa a la ambigüedad e incompletitud. Para evitar lo anterior y reducir las falencias propias atribuidas al uso del lenguaje natural, se propone en diferentes trabajos el uso de métodos, lenguajes, herramientas y técnicas para llevar a cabo una especificación formal de los requisitos, asegurando que su implementación podría reducir significativamente su ambigüedad e incompletitud de una forma más rápida y anticipada, minimizando el número de errores durante el proceso» (Toro & Gálvez, 2016, 112).

Se ha establecido que la eficiencia en el uso de metodologías de desarrollo de software radica, en gran medida, en contar con herramientas para la gestión de requisitos. Sin embargo, *«ninguna de las herramientas existentes logra apoyar por completo la metodología»* (Gómez, Urrego & González, 2009, 60), debido a sus limitaciones dependiendo del tipo de sistema, por ejemplo, cuando de sistemas embebidos se trata.

El presente trabajo pretende mostrar un análisis de las herramientas de gestión de requisitos de software que se encuentran en el mercado, con la intención de determinar sus principales entradas, ventajas y falencias, que ayudarán a la formulación del modelo. Para ello, se ha estructurado de la siguiente manera: en la primera sección, el lector encontrará una exploración de las herramientas para la gestión de requerimientos. Luego, en la segunda sección, hay una caracterización de las herramientas exploradas y las pruebas realizadas sobre las mismas. En la sección tres, se realiza un análisis comparativo de las herramientas y, finalmente, en la última sección, se presentan las conclusiones parciales y trabajo futuro.

1. Fundamento Teórico

1.1 Ingeniería de requisitos

Tratar la Ingeniería de Software implica considerar la ingeniería de requisitos, *«tratada como una disciplina que tiene por propósito “desarrollar una especificación de requerimientos completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema”»* (Toro & Peláez, 2016, 118), por tanto, cualquier proyecto de desarrollo de software debe satisfacer las características alineándose con el propósito de la Ingeniería de Requisitos.

«La ingeniería de requisitos o Ingeniería de requerimientos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado. La ingeniería de requisito toma en cuenta los diversos requisitos de las partes interesadas que pueden entrar en conflicto entre ellos. Muchas veces se habla de requerimientos en vez de requisitos, esto se debe a una mala traducción del inglés. El propósito de esta ingeniería es hacer que los requisitos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones, etc.» (López, 2015, 13).

No sobra indicar que los requisitos de un sistema son de dos tipos:

- *Funcionales*: según Charry & Umbarila (2017, 207) ellos *«ordenan qué va a hacer el sistema “el qué” de la funcionalidad a desarrollarse en el sistema»*, sin considerar sus limitaciones físicas.
- *No funcionales*: de acuerdo con Buitrón, Flores & Pino (2018, 143), definen las propiedades ambientales y las restricciones de implementación relacionadas con el desempeño del producto, debiendo considerarse como parte del proceso de desarrollo de software debido a su relación con la usabilidad, flexibilidad, operatividad y seguridad del producto solicitado, e inclusive el desempeño, la reusabilidad y la confiabilidad del software.

«Para (Sommerville, 2005, pág. 108) “el proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requisitos (RE)”. También menciona que “la Ingeniería de Requisitos es el proceso de desarrollar una especificación de software. Las

especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema". (Ibídem, Pág. 82). Es evidente entonces la relevancia que tiene la especificación de requisitos dentro de la Ingeniería de requisitos, pues permite que el desarrollador entienda y de solución a las necesidades expresadas por el cliente» (Toro & Gálvez, 2017, 28).

1.2 Gestión de requerimientos

«La Gestión de Requerimientos no se ejecuta en algún momento en específico del proyecto, sino que abarca todo lo referente a la planeación, monitorización del progreso y gestión de cambios durante todo el ciclo de vida de este» (Murcia, 2014, 25). Al existir diversos tipos de proyectos, «la gestión cambia constantemente para identificar lo que el producto necesita para ser apropiado, competitivo y útil» (Murcia, 2014, 28), y generar mejores resultados.

La planeación varía su gestión de acuerdo con los resultados obtenidos en etapas anteriores, los cuales deberían ser aprobados antes de realizar cualquier actividad acorde por el experto del negocio, con el fin de establecer si las soluciones realmente satisfacen las necesidades de los usuarios. La gestión, adicionalmente, permite *«un correcto manejo de las versiones de los requerimientos, de los documentos generados y de los modelos utilizados»* (Murcia, 2014, 25), pues un error en el manejo de versiones, podría afectar el tiempo, costo, cronograma, diseño, o en el peor de los casos la cancelación del proyecto, ya que *«aún se están cometiendo errores fatales, que comprometen el correcto desempeño del proyecto, y en algunos casos, la propia continuidad del mismo»* (Murcia, 2014, 14).

2. Metodología

La metodología, soportada en un estudio de exploración documental de la literatura de los últimos 10 años, bases de datos especializadas y contacto directo con los fabricantes o distribuidores de las herramientas, contempla cuatro fases.

2.1 Fase 1: Exploración del estado del arte

Consiste en realizar un levantamiento de información en diferentes fuentes físicas y electrónicas, relacionado principalmente con las teorías

sobre ingeniería del software e ingeniería de requisitos, así mismo, sobre las herramientas de gestión de requisitos que acompañan el desarrollo de un proyecto de software. Lo anterior, con la intención de contextualizar la situación problemática y justificación del proyecto de investigación.

2.2 Fase 2: Caracterización de las herramientas

Se clasifican y describen las herramientas más usadas en el entorno mundial por las compañías desarrolladoras de software a la hora de desarrollar y gestionar los requerimientos. Esta caracterización se realiza considerando factores como tipo de licenciamiento (de pago y libres), plataforma, facilidad de soporte o contacto con el fabricante, fecha última actualización, vigencia en el mercado, posibilidad de prueba, entre otras. Además, se determinan sus principales características, funcionalidades y limitaciones, de tal manera que se puedan identificar las principales entradas de información que requieren para su funcionamiento, ya que el proyecto pretende, a futuro, aportar en la formulación de un modelo automatizado para el aseguramiento de la calidad de los requerimientos.

Posteriormente, se realizan pruebas sobre varias de las herramientas identificadas para estudiar su funcionamiento, necesidades de información y restricciones, con el ánimo de determinar los elementos de entrada de un requerimiento solicitados por las mismas. Dentro del desarrollo metodológico, es precisamente esta fase en la que se centrará en adelante el propósito de este documento.

2.3 Fase 3: Análisis comparativo de las herramientas

Se realiza una comparación de las herramientas utilizadas principalmente en implementar una especificación de requisitos de software a través de diferentes plataformas, identificando los elementos de entrada requeridos, las restricciones, los detalles del proceso a seguir, los atributos que permite definir, las ventajas y desventajas, entre otros.

Dado que la mayoría de ellas permiten ejecutar otras funciones relacionadas con la ingeniería de software, se enfatiza en la gestión de requerimientos. Esta fase pretende además, entregar a la sociedad elementos diferenciadores de las múltiples herramientas, de tal manera que las personas y empresas que hagan uso de herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora), puedan tomar decisiones con facilidad a la hora de elegir una de ellas.

2.4 Fase 4: Conclusiones y trabajo futuro

En esta fase del proyecto, se plantean las conclusiones iniciales de acuerdo con los hallazgos y resultados del trabajo realizado con las herramientas, de tal manera que las mismas puedan contribuir a las etapas subsecuentes del proyecto de investigación. Estas, se relacionan con el diseño conceptual de un modelo para el aseguramiento de la calidad de los requerimientos y el desarrollo de un sistema automatizado para la gestión de los mismos.

3. Resultados y discusión

3.1 Descripción de resultados

3.1.1 Herramientas para la gestión de requerimientos. Una de las etapas más importantes es el desarrollo de los requerimientos, dado que allí se conoce a profundidad el problema a resolver y, por defecto, se precisa el parámetro bajo el cual se cumplirán o no las expectativas del cliente. Se vislumbra como hipótesis que ninguna de estas herramientas satisface totalmente las necesidades de las MiPymes desarrolladoras de software, pues cada una presenta características, funcionalidades y limitaciones diferentes que no permiten lograr un cubrimiento completo de sus necesidades⁵, por lo que debe considerarse su diversidad.

El comparativo entre algunas herramientas para la gestión de requerimientos, tanto de pago (*Case complete, Enterprise Architect, Jama software, Visual paradigm, Caliber, Rational IBM, Visure Requirements, Case Spec*) como de uso libre (*REM, Rmtoo, Let's req, REMAS, RETO, DRES, Gatherspace, TopTeam Analyst*), se consolida en la Tabla 1, con sus principales diferencias, aspectos de mayor relevancia, ventajas y falencias, permitiendo encontrar que las herramientas de pago contienen mejores funcionalidades y son más robustas, aunque el alto costo de sus licencias dificulta su adquisición por las MiPymes, que generalmente solo funcionan en plataformas Windows.

5 El alto costos de licencias y la capacidad limitada de algunas herramientas de uso libre, pueden afectar a varias empresas de la industria del software pues, según Supersociedades (2015, 6), no todas están en la capacidad financiera y operativa para implementar algunos procesos y/o modelos que les permitan asegurar la calidad, de sus procesos y productos, al ser pymes en su mayoría.

Además, «*Quispe et al., descubre que algunas de las características principales de las pequeñas empresas son: • El equipo de trabajo es poco calificado [...]. • El proceso de desarrollo y gestión [...] se basa en técnicas informales. • No se encuentra una estructura organizacional clara y no se definen roles dentro de la organización*» (De la Cruz & Castro, 2015, 6).

Tabla 1. Tabla comparativa entre herramientas para gestión de requerimientos

Herramienta	Tipo*	Plataforma	Ventajas	Desventajas
Enterprise Architect	De pago	Windows	Modelado completo para el ciclo de vida. Repositorio escalable para el trabajo en equipo. Veloz, estable y efectivo. Genera documentación. Ingeniería inversa del código fuente. Permite trazabilidad.	Solo funciona en plataforma Windows.
Visual Paradigm	De pago	Windows, Linux, MAC OS	Genera modelos VP-UML instantáneamente a partir de código binario .Net. Genera documentación en formatos HTML y PDF, de todo el trabajo sin necesidad de herramientas externas. Permite trazabilidad. Integración con otras plataformas.	Imágenes y reportes generados, no son de muy buena calidad.
Case Complete	De pago	Windows	Interfaz amigable. Permite trazabilidad entre los requerimientos. Rápido y de fácil manejo.	Solo funciona en plataforma Windows.
Jama Software	De pago	Windows	Dispone de requisitos de calidad en un entorno compartido. Implica a los interesados en el proyecto y fomentar la revisión y validación. Optimiza la gestión de requisitos óptima. Gestión de pruebas. Interfaz con una experiencia de usuario excelente. Adaptación a cualquier tipología y ciclo de vida de requisitos. Genera todo tipo de vistas y reportes. Integración con otras plataformas.	Solo funciona en plataforma Windows.
REM	Libre	Windows	Permite exportar el documento HTML.	Última actualización en 2004. Presenta un error que no afecta su funcionalidad. Interfaz poco amigable.
DRES	Libre	Windows, Linux, MAC OS	Es de código abierto que permite contribuciones para mejora. Funciona en entorno web, razón por la cual no presenta restricciones de plataforma.	Última actualización no reciente. Al ejecutar en un servidor XAMPP genera error en su código fuente.
REMAS	Libre	Windows, Linux, MAC OS	Permite la gestión de requerimientos del lado de la web. Es de código abierto. Permite administración de versiones. Basada en Java y cuenta con un repositorio SVN.	Interfaz poco amigable. Ejecución compleja.

* Para las herramientas de pago (las primeras cuatro de la tabla), se encuentra información sobre los tipos de licenciamiento y costos en: <http://www.sparxsystems.com.ar/products/ea/purchase.html>, <https://www.visual-paradigm.com/shop/pricelist.jsp>, <http://casecomplete.com/pricing> y <https://www.jamasoftware.com/platform/pricing/>, respectivamente.

3.1.2 Pruebas a las herramientas. Algunas de las herramientas probadas se presentan a continuación:

- *Case Complete* (Figura 2), solicita ingresar algunos detalles del requerimiento a especificar, tales como nombre, prioridad, descrip-

ción, tipo, responsable, estado, documentos relacionados y notas, entre otros atributos. De igual forma, se presenta una trazabilidad del requerimiento que es muy importante para gestionar el control de versiones, los requerimientos asociados a este y las consecuencias de su modificación en cualquier momento.

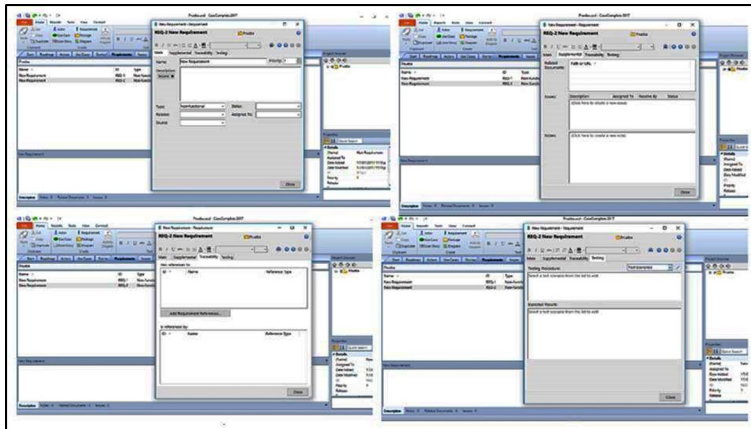


Figura 2. Datos de entrada solicitados por Case complete

- *Enterprise Architect* (Figura 3), es una herramienta de pago que acompaña la gestión de requerimientos y las demás fases del desarrollo de software, por su variedad de atributos: tipo, estado, alias, palabras clave, autor, dificultad del requerimiento, prioridad, versión, fase y una funcionalidad para la especificación del requerimiento relacionada con reglas, restricciones y otros requerimientos.

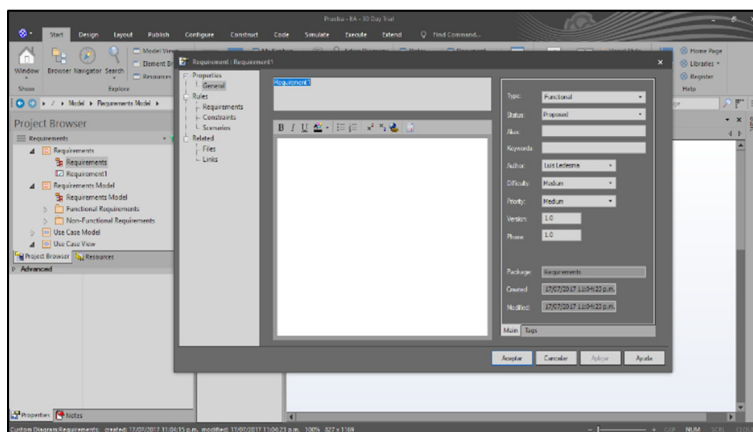


Figura 3. Datos solicitados por Enterprise Architect

- *Visual Paradigm* tiene un conjunto de ayudas para el desarrollo de programas informáticos (planificación, análisis y diseño, generación del código fuente y documentación), soporta el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Entre sus principales ventajas está funcionar en plataformas *Windows* y *Linux*, con diversas versiones (*Enterprise, Community Edition, Professional*), siendo una de las más completas y enfocadas a la gestión de requerimientos, que permite generar documentación, trazabilidad a los requerimientos, realizar gráficas y diagramas de requerimientos, vincular a los miembros del equipo para agilizar, compartir y realizar seguimiento a los requerimientos. De igual forma, tiene unos atributos clave a la hora de crear los requerimientos: id, nombre, tipo, riesgo, descripción (Figura 4).

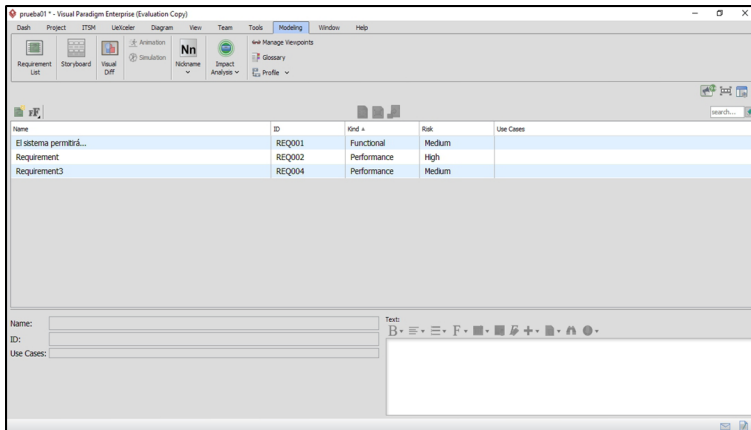


Figura 4. Datos solicitados por Visual Paradigm

3.2 Análisis y discusión de resultados

Los resultados obtenidos permiten describir los logros alcanzados en cada una de las etapas de la metodología propuesta, la primera de ellas tuvo como propósito realizar un levantamiento de estado de arte de los requisitos, además de una descripción y caracterización de las herramientas de desarrollo y gestión de requisitos que se encuentran en el mercado. En la segunda etapa se clasificaron y describieron las herramientas usadas por las empresas desarrolladoras de software, haciendo una diferenciación entre las de pago y aquellas de uso libre. Se determinan también sus principales características, funcionalidades y limitaciones, así como algunas pruebas que arrojasen información

de primera mano sobre las entradas de información requeridas por las mismas. Finalmente, en la tercera etapa se realizó un análisis comparativo entre las herramientas estudiadas.

En el anterior análisis, es evidente que las herramientas de pago son mucho más robustas y permiten implementar mayor número de funciones, sin embargo, representan costos elevados que hacen más difícil su adquisición por parte de pequeñas y medianas empresas. De igual forma, se puede apreciar que la totalidad de ellas solo pueden ser usadas en plataforma Windows.

Por otro lado, las herramientas libres tienen funciones y características importantes a la hora de crear requerimientos, las mismas pueden ser elegidas por personas o empresas que no cuenten con capital suficiente para este tema o recién comiencen en el ámbito laboral, pues con sus capacidades pueden satisfacer ampliamente las necesidades de pequeñas empresas desarrolladoras de software al momento de gestionar sus requerimientos.

El ejercicio permite también determinar los datos de entrada que son comunes entre las herramientas, tales como nombre del requerimiento, id, versión, fecha de creación, descripción; así como los de obligatorio diligenciamiento, tales como prioridad, tipo, estado, restricciones, entre otros. Esto contribuye significativamente al análisis de las necesidades principales para el modelo automatizado que permita asegurar la calidad de los requerimientos, planteado como trabajo futuro.

Otros aspectos como la trazabilidad, integración con otras plataformas, documentación relacionada requerida y validación de la especificación obtenida son vitales para la investigación, que permiten dar una idea de la completitud con la que un sistema de gestión de requerimientos debe operar.

Existe cantidad de herramientas, tanto de pago como libres, que pueden ser usadas en la gestión de requisitos. Pese a ello, son pocas las empresas que las utilizan como parte de un proceso adecuado de aseguramiento de la calidad. Hacerlo podría disminuir considerablemente el porcentaje de requerimientos incompletos en los proyectos, los cambios en las especificaciones y, seguramente, el número de proyectos que fracasan según lo expresado por Cristiá (2011, 6), disminuyendo a su vez los costos que obedecen a la reparación de productos deficientes o de poca calidad debidos a un pobre desarrollo y gestión de los requerimientos. En la misma línea, se coincide con Supersociedades (2015, 6) y De la Cruz & Castro (2014, 6) sobre

las características de las empresas desarrolladoras de software de Colombia, y sus limitaciones para asegurar la calidad de sus procesos y productos.

También, una vez presentado este análisis sobre la revisión sistemática de herramientas empleadas para la gestión de requisitos, es posible expresar el cumplimiento del objetivo de este trabajo, pues permite establecer los principales atributos y características a considerar en un modelo automatizado para el Aseguramiento de la Calidad de los Requisitos.

Finalmente, el análisis realizado entrega información relevante para la siguiente etapa del proyecto, relacionada con el diseño conceptual del modelo antes mencionado y el desarrollo de un sistema automatizado para la gestión de los requerimientos.

4. Conclusiones

Del análisis literario expuesto se puede indicar que la Ingeniería de requisitos y, como parte de ella la gestión de los requerimientos, no es la solución definitiva al problema de producir software de baja calidad, pero ayuda en gran medida al descubrimiento y solución de falencias en etapas tempranas del desarrollo de proyectos de software, reduciendo costos y tiempo en el ciclo de vida.

Por lo tanto, las herramientas CASE agilizan y facilitan la gestión de requerimientos software, ofreciendo apoyo permanente al equipo de trabajo encargado del desarrollo del proyecto. El trabajo realizado permite determinar que en el mercado se pueden encontrar muchas herramientas gratuitas y de pago, las cuales brindan un abanico de posibilidades a las empresas y/o personas, siendo además un complemento muy importante al proceso de desarrollo de software, siempre y cuando se consideren las limitaciones que cada una presenta.

Sin embargo, la mayoría de herramientas de pago que se encuentran en el mercado presentan altos costos en el licenciamiento (*Enterprise Architect*: 299-699 USD, *Visual Paradigm*: 99-1999 USD, *Case Complete*: 699 y 28999 USD, por mencionar algunas), lo que hace difícil su adquisición para muchas de las pequeñas y medianas empresas desarrolladoras de software, razón por la cual, en su gran mayoría, utilizan herramientas gratuitas o no utilizan ninguna de ellas.

Con lo anterior, la gestión de requerimientos es una tarea que aún tiene mucho por explorar, pues requiere optimizar y definir elementos de entradas fundamentales y concisos para cumplir con las expectativas del cliente y satisfacer la necesidades del proyecto.

Referencias bibliográficas

- BITRÓN, Sandra L.; FLORES-RÍOS, Brenda L. & PINO, Francisco J. (2018). Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders [en línea]. En: *Ingeniare: Revista chilena de ingeniería*, Vol. 26, No. 1 (ene-mar). Arica (Chile): Universidad de Tarapacá. p. 142-156. E-ISSN: 0718-3291 <<http://dx.doi.org/10.4067/S0718-33052018000100142>>, <http://www.ingeniare.cl/index.php?option=com_ingeniare&view=d&doc=95/art14.pdf&aid=616&vid=95&lang=es> [consulta: 20/05/2018]
- CAMACHO ZAMBRANO, Antonio Nicolás (2005). Herramienta para el análisis de requerimientos dentro de la pequeña empresa desarrolladora de software en Bogotá [en línea]. Proyecto de grado (Ingeniero de Sistemas). Bogotá (Colombia): Pontificia Universidad Javeriana, Facultad de Ingeniería. 147 p. <<http://javeriana.edu.co/biblos/tesis/ingenieria/Tesis189.pdf>> [consulta: 10/12/2017]
- CHARRY RAMÍREZ, Paola & UMBARILA TORRES, Ricardo (2017). Modelado y gestión de la información. Tecnología, Investigación y Academia, TIA, Vol. 5, No. 2 (jul-dic). Bogotá (Colombia): Universidad Distrital 'Francisco José de Caldas'. p. 199-213. ISSN: 2344-8288 <<https://revistas.udistrital.edu.co/ojs/index.php/tia/article/download/10260/pdf>> [consulta: 20/05/2018]
- CRISTIÁ, Maximiliano (2011). Introducción a la Ingeniería de Requerimientos [en línea]. Apuntes de clase: Cursos Ingeniería de Software I y II. Rosario (Argentina): Universidad Nacional de Rosario, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Departamento de Ciencias de la Computación. 58 p. <<https://www.fceia.unr.edu.ar/~mcristia/publicaciones/ingreq-a.pdf>> [consulta: 20/06/2014]
- DE LA CRUZ LONDOÑO, Cristian Andrés & CASTRO GUEVARA, Gustavo Andrés (2015). Metodología para la adquisición y gestión de requerimientos en el desarrollo de software para pequeñas y medianas empresas (pymes) del departamento de Risaralda [en línea]. Tesis (Magister en ingeniería de sistemas y computación). Pereira (Colombia): Universidad Tecnológica de Pereira, Facultad de Ingeniería. 217 p. <<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/5589/0053D278.pdf?sequence=1>> [consulta: 25/04/2018]
- GÓMEZ MALDONADO, Jorge Eduardo; URREGO GIRALDO, Germán & GONZÁLEZ PALACIO, Liliana (2009). REASEM: Herramienta para la gestión de requisitos [en línea]. En: *Avances en Sistemas e Informática*, Vol. 6, No. 2. Bogotá (Colombia): Universidad Nacional de Colombia. p. 59-67. e-ISSN: 1909-0056. <<https://revistas.unal.edu.co/index.php/avances/article/view/20334>> [consulta: 12/12/2017]
- LOPERA CARMONA, Diana Catalina (2012). Análisis estratégico de la industria colombiana de software a partir de la simulación de escenarios de competencia utilizando Dinámica de Sistemas [en línea]. Trabajo final de Maestría (Magister en Ingeniería Administrativa). Medellín (Colombia): Universidad Nacional de Colombia, Facultad de Minas. 57 p. <<http://www.bdigital.unal.edu.co/8466/1/39175215.2012.pdf>> [consulta: 23/05/2018]
- LÓPEZ GARCÍA, Estefanía (2015). Ingeniería de requisitos [en línea]. En: *MoleQla: revista de Ciencias*, No. 19 (sep). Sevilla (España): Universidad Pablo de Olavide. p. 13-14. e-ISSN: 2173-0903 <https://www.upo.es/moleqla/export/sites/moleqla/documentos/Numero19/Nxmero_19.pdf> [consulta: 12/12/2017]
- MURCIA TORRES, Juan Darío (2014). Guía para la integración de métodos formales de ingeniería [en línea]. Trabajo de grado (Ingeniero de Sistemas). Bogotá (Colombia): Pontificia Universidad Javeriana, Facultad de Ingeniería. 62 p. + Anexos. <<https://repositorio.javeriana.edu.co/bitstream/handle/10554/12636/MurciaTorresJuanDario2014.pdf?sequence=1&isAllowed=y>> [consulta: 08/12/2017]
- SALAZAR OSORIO, Holmes Giovanni; RENGIFO ROMERO, Harvin Jessid; MACHUCA VILLEGAS, Liliana Esther & ARANDA BUENO, Jesús Alexander (2012). Una guía general para la especificación y verificación formal de requerimientos usando Event-B™ y Rodin™ [en línea]. En: *Revista Educación en Ingeniería*, Vol. 7, No. 14. Bogotá (Colombia): Asociación Colombiana de Facultades de Ingeniería. ACOFI. p. 82-91. ISSN: 1900-8260 <<http://dx.doi.org/10.26507/rei.v7n14.230>>, <<https://www.educacioningenieria.org/index.php/edi/article/view/230>> [consulta: 15/12/2017]
- SUPERINTENDENCIA DE SOCIEDADES, SUPERSOCIEDADES (2015). Desempeño del sector software 2012-2014: Informe [en línea]. Bogotá (Colombia): Superintendencia de Sociedades, Delegatura para Asuntos Económicos

- y Contables, Grupo de Estudios Económicos y Financieros. 27 p. <<http://www.supersociedades.gov.co/noticias/Documents/2015/EE1-%20Estudio%20Sectorial%20Software-%202015%20VII%2030.pdf>> [consulta: 10/11/2017]
- TORO LAZO, Alonso & GÁLVEZ BOTERO, Juan Guillermo (2016). Especificación de requisitos de software: una mirada desde la revisión teórica de antecedentes [en línea]. En: Entre Ciencia e Ingeniería, Vol. 10, No. 19 (ene-jun). Pereira (Colombia): Universidad Católica de Pereira. p. 108-113. ISSN 1909-8367 <<http://biblioteca.ucp.edu.co/ojs/index.php/entrecei/article/view/2803/2785>> [consulta: 12/02/2018]
- TORO LAZO, Alonso & GÁLVEZ BOTERO, Juan Guillermo (2017). Procedimiento para especificar y validar requisitos de software en Mipymes desarrolladoras de software de la ciudad de Pereira, basado en estudios previos en la región [en línea]. Tesis (Magíster en Gestión y Desarrollo de Proyectos de Software). Manizales (Colombia): Universidad Autónoma de Manizales, Facultad de Ingenierías. 111 p. <http://repositorio.autonoma.edu.co/jspui/bitstream/11182/1116/1/INFORME%20FINAL%20PROYECTO%20MGDPS%20ALT_JGGB_04-03-2017.pdf> [consulta: 25/04/2018]
- TORO, Alonso & PELÁEZ, Luis Eduardo (2016). Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira [en línea]. En: Entre Ciencia e Ingeniería, Vol. 10, No. 20 (jul-dic). Pereira (Colombia): Universidad Católica de Pereira. p. 117-123. ISSN: 1909-8367 <<http://biblioteca.ucp.edu.co/OJS/index.php/entrecei/article/view/3032/3364>> [consulta: 12/02/2018]
- TORRES DE PAZ, Rosa María (2009). El proceso de Ingeniería de Requisitos en el ciclo global del software [en línea]. Trabajo de investigación (Doctora en Ingeniería en Informática). Sevilla (España): Universidad de Sevilla, Departamento de Lenguajes y Sistemas Informáticos. 65 p. <<http://www.lsi.us.es/docs/doctorado/memorias/memo-inv-rosa-m-torres.pdf>> [consulta: 18/05/2014]
- YANG, Ye; HE, Mei; LI, Mingshu; WANG, Qing & BOEHM, Barry (2008). Phase distribution of software Development Effort [online]. In: Second ACM-IEEE international symposium on Empirical software engineering and measurement, ESEM'08 (09-10/10/2008), Kaiserslautern (Germany): ACM – IEEE. ROMBACH, Dieter; ELBAUM, Sebastian & MÜNCH, Jürgen (eds.). Proceedings of the ESEM'08. New York (NY, USA): ACM. p. 61-69. ISBN: 978-1-59593-971-5 <<https://doi.org/10.1145/1414004.1414016>>, <<https://dl.acm.org/citation.cfm?doid=1414004.1414016>> [consult: 12/02/2018]