

TÉCNICAS PARA LA MEJORA DE LA CALIDAD EN LA INGENIERÍA DE REQUISITOS EN LAS EMPRESAS DE SOFTWARE DE ARGENTINA

María Laila Becker, Rafael Blanc, Carlos Casanova, Julián Escalante, Lourdes Pralong,
Laura Elena Ríos, Rossana Sosa Zitto

Facultad de Ciencia y Tecnología, Universidad Autónoma de Entre Ríos
rafaellujanblanc@yahoo.com.ar; carlos.casanova16@gmail.com; mdlpralong@gmail.com;
elenrios2@gmail.com; rossana.e.sosa.zitto@gmail.com

RESUMEN

La Ingeniería de Requisitos, es una fase crucial en el proceso de desarrollo de software, ya que ésta determina la calidad del diseño y la implementación y tiene un alto impacto en el éxito de los proyectos. Por tal motivo es necesario implementar estrategias que garanticen su éxito, entre ellas técnicas específicas para gestionar el conocimiento, que apunten a mejorar la calidad de los requisitos. Uno de los mayores inconvenientes que se presenta en esta fase es la brecha de interpretación que se produce entre el ingeniero de requisitos y el usuario del software lo que lleva a errores de traducción de necesidades. Para estructurar y categorizar la masa de información proveniente del proceso de educación de requisitos, es posible utilizar técnicas de representación del conocimiento a los efectos de facilitar la comprensión del problema manifestado por el usuario y reducir la brecha conceptual, entre los procesos de educación de requisitos y su modelado.

Palabras clave: Ingeniería de Software, Representación del conocimiento, Ingeniería del Conocimiento, Inteligencia Artificial.

CONTEXTO

El presente trabajo se desarrolla en el ámbito del Grupo de Investigación de Ingeniería de Software (GIISW), perteneciente a la Sede Concepción del Uruguay de la Facultad de Ciencia y Tecnología dependiente de la Universidad Autónoma de Entre Ríos.

1. INTRODUCCIÓN

Actualmente, la industria nacional del sector SSI (Software y Servicios Informáticos), al

igual que otras industrias competitivas, es uno de los sectores de la economía que muestra mayor nivel de dinamismo y, por ende, es considerado por el gobierno como un actor de importancia estratégica para el desarrollo del país en su conjunto. [1], [2].

Dentro de esta industria uno de los retos fundamentales de las empresas desarrolladoras de software es implementar aplicaciones que sean entregadas a tiempo, que no involucren presupuestos elevados y que satisfagan las necesidades del usuario, utilizando para su efecto metodologías y herramientas que guíen el proceso de desarrollo de Software [3].

Otra exigencia que ocupa a la industria de software es la calidad, este tema día a día genera mayor interés, así lo demuestra el informe presentado en 2017 por la OPSSI (Observatorio permanente de la industria del software y servicios informáticos). Uno de los principales motivos es que las certificaciones de calidad son uno de los requisitos para ingresar al régimen de promoción de la Ley de Software lo cual es un incentivo no menor para las empresas del sector, dados los beneficios que obtienen, tales como: descuentos de hasta el 60% del impuesto a las ganancias y acceso a bonos para pagar impuestos nacionales [1]–[4]. Sumado a esto, las certificaciones permiten acceder a mercados que exigen normas internacionales de calidad y aumentar sus oportunidades de negocios a través de la confianza que genera en sus clientes en relación a la seguridad y calidad de sus productos. [5].

Diversos estudios existen sobre los fracasos en la industria del software, tal es el caso del estudio presentado por la consultora internacional Standish Group, quien cita que tan solo el 32% de los proyectos de desarrollo de software, se pueden considerar exitosos, el 44% se entregaron fuera de plazo, excedieron

su presupuesto y no cubrieron la totalidad de las características y funcionalidad pactada, y el 24% de los proyectos fueron cancelados. En este mismo estudio se puntualiza que el principal factor para el fracaso de un proyecto de desarrollo de software radica en la mala calidad de los requisitos y la definición poco clara de los mismos [6].

Para obtener calidad en los productos de software hay que implementar prácticas de calidad en el proceso de desarrollo [7]–[10]. La base de dicho proceso, son los requisitos ya que determinan la calidad del producto final, razón por la cual la Ingeniería de Requisitos (IR) es una etapa crucial dentro del proceso de desarrollo de software (SW) [11], dentro de esta etapa una de las dificultades que se presenta es la especificación clara del comportamiento que se espera del software, ya que se tiene muchas veces sólo una descripción textual del mismo.

La elicitación de requisitos, subactividad de la IR tiene un alto impacto en el diseño y en las fases posteriores del ciclo de vida de un producto software. Si esta se realiza correctamente contribuye a reducir cambios y errores en los requisitos. Esta etapa se considera como la primera en el proceso de abstracción, comprensión y concepción de las funcionalidades que un producto software debe cumplir. [12].

El objetivo de la Ingeniería de Requisitos es desarrollar una especificación completa, consistente y no ambigua de los requisitos, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema [13]. Tiene cuatro etapas fundamentales la elicitación, el análisis, la especificación y la validación. Durante ellas constantemente se requiere colaborar, coordinar y comunicar con los distintos stakeholders, entre sí, pero manteniendo distintas perspectivas y puntos de vista lo que plantea grandes retos de comunicación. [14]. Diversos problemas se presentan durante esta etapa, los cuales se deben a la carencia de un proceso adecuado de definición y entendimiento del problema y a la definición poco clara de las necesidades del cliente [15]. Pese a todos los estudios realizados, aún hoy los requisitos incompletos

siguen siendo uno de los principales factores de fracaso de los proyectos [16]. Según Yang el 56% de los errores encontrados en el software son generados en la fase de requisitos y el porcentaje total del esfuerzo del proyecto que se destina en esta fase no llega al 20%, independientemente del modelo de desarrollo utilizado. [17].

Para intentar minimizar el costo y maximizar la calidad final del producto software es necesario estabilizar la especificación de requisitos [18], por lo anterior existen un conjunto de técnicas para evaluación y selección de los requisitos que buscan disminuir los costos y tiempos asociados a los mismos [19]–[21].

Dada la naturaleza cambiante de los requisitos, es necesario gestionarlos durante todo el desarrollo del software y considerar que la producción de los requisitos se caracteriza por ser sustancialmente interactiva e iterativa [22].

Uno de los mayores inconvenientes que se presenta dentro de esta etapa es la brecha de interpretación que se produce entre el Ingeniero de requisitos y el cliente lo que lleva a posibles errores de traducción de necesidades por parte del Ingeniero y a una falta de comprensión de los requisitos iniciales y una inhabilidad para gestionar los cambios durante el desarrollo. Por tal motivo es importante realizar una buena traducción de los requisitos y para esto se utilizan una serie de técnicas de elicitación a fin de reflejar las necesidades del usuario para ser integradas al software [23], [24]. Los errores que se presentan se podrían evitar si se hace uso de buenas prácticas en el marco de la ingeniería de requisitos. [25]. Las técnicas de modelado o representación utilizadas normalmente en este marco no definen un proceso para tratar la complejidad contenida en el discurso de los interesados. Esto dota al proceso de análisis de un grado tal de imprecisión que hace que sea difícil llevar a cabo en forma efectiva esta actividad, dependiendo mucho de la experiencia de los analistas para seleccionar la técnica que mejor encaja en cada situación. Aunque muchos autores han desarrollado técnicas de modelado, al no contar con ninguna guía, la mayoría de los ingenieros prefieren utilizar las técnicas tradicionales. [18]. Pero al ser la IR una actividad que tiene un alto impacto

en el éxito o fracaso de los proyectos de desarrollo de software, es necesario implementar estrategias que garanticen el éxito de esta fase, entre ellos que apunten a aportar mejoras a la metodología, las técnicas, las herramientas utilizadas, así como también incrementar el compromiso y la participación de todos los involucrados, una tarea altamente compleja, por lo que es necesario utilizar técnicas específicas en la gestión del conocimiento a fin de mejorar la calidad de los requisitos, obtenidos en esta fase.

La ingeniería de conocimiento es una disciplina que es parte de la Inteligencia Artificial, cuyo fin es el diseño y desarrollo de sistemas expertos. Un Sistema Experto (SE), es un software que imita el comportamiento de un experto humano en la solución de un problema. Pueden almacenar conocimientos de expertos para un campo determinado y solucionar un problema mediante deducción lógica de conclusiones.

Diversos estudios avalan la existencia de problemas para satisfacer la necesidad de estructurar y categorizar la masa de información proveniente del proceso de educación de requisitos a los efectos de facilitar la comprensión del problema manifestado por el usuario. En otros términos, formalizar los requisitos mediante técnicas de representación del conocimiento [18].

Para dar solución a los problemas planteados se han llevado adelante los estudios donde se analiza la aplicación de la Ingeniería del Conocimiento y de la gestión del mismo, en el campo de desarrollo de software. Ésta ha sido de gran utilidad dada la naturaleza misma de la Ingeniería de Software como una disciplina intensa en conocimiento [26]. Se ha analizado también el grado de concordancia entre los atributos de calidad de los conocimientos y de los requisitos, se concluye que ambos son coincidentes. Los mismos son exactitud, consistencia, certidumbre y fácil de usar [27].

Entre las técnicas más utilizadas para la representación de conocimiento se encuentran los mapas conceptuales, los cuales representan el conocimiento conceptual de un dominio. Los mismos se basan en la utilización de grafos que representan conceptos, objetos y relaciones

entre ellos [28]. El eliminar ambigüedades y omisiones en los requisitos contribuye directamente a mejorar la calidad de los mismos y como consecuencia obtener un software de mayor calidad. La selección de esta técnica se debe a que los mapas conceptuales confeccionados por expertos tienden a exhibir altos niveles de acuerdo [29], [30]. Además, la utilización de Mapas Conceptuales tiene como fortaleza la generación de modelos de conocimiento [31].

En el proceso de Ingeniería de Conocimiento involucrando un experto, un investigador puede proyectar el software en una pantalla y trabajar junto al experto en la confección del diagrama, combinando de esta manera la Ingeniería de Conocimiento con la representación del conocimiento. Los enlaces pueden conectar también a otros mapas conceptuales, formando así un modelo de conocimiento. De esta manera los mapas conceptuales pueden constituir modelos de conocimiento estructurado y servir como repositorios de conocimiento de expertos. También pueden servir como interfaces para sistemas inteligentes para la construcción de herramientas de soporte o entrenamiento. Adicionalmente se explorarán técnicas alternativas.

Considerando este tipo de enfoques basados en inteligencia artificial, orientados a propiciar la calidad del proceso de elicitación de requerimientos, cabe destacar el trabajo de [32], cuyo objetivo es obtener una mejor representación de la información, partiendo de la automatización del proceso de generación de un mapa conceptual. De esta manera, se provee un mejor soporte para la interpretación y la toma de decisiones a partir de datos recolectados en el desarrollo del proyecto. Esta estrategia constituye una buena herramienta de gestión del conocimiento, ya que los conceptos pueden ser capturados o consultados, permitiendo el descubrimiento automático de conexiones implícitas para generar nuevos mapas. Los ingenieros del conocimiento extraen el conocimiento de los expertos humanos en una determinada área, y codifican dicho conocimiento de manera que pueda ser procesado por un sistema software [18].

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

2.1. Relevamiento del estado de la industria en cuanto a IR

Se trabajará mediante estudios de casos a fin de hacer un análisis descriptivo de la situación actual de la ingeniería de requisitos en las empresas de software. Como primer parámetro de selección de empresas para la muestra del estudio se seleccionarán firmas que desarrollan software a medida a fin de que tengan la etapa de toma de requisitos, que cuenten con una persona o departamento que cumpla la función de ingeniero de requisitos y que tengan al menos diez empleados a fin de asegurar una estructura de funciones de desarrollo, entre otros criterios. Finalmente se tendrá en cuenta que las firmas tengan al menos cinco años de funcionamiento en el mercado a fin de poder hacer un análisis de la trayectoria de las mismas en cuanto a los requisitos y los problemas y aprendizajes logrados en este aspecto. Se les realizará entrevistas en profundidad sobre las dimensiones abarcadas dentro del aspecto teórico a fin de identificar su comportamiento en cuanto a los requisitos y las consecuencias de ellos en sus resultados en el desarrollo de software.

2.2. Metodologías basadas en Conocimiento para la educación de requisitos

Las entrevistas permitirán detectar patrones comunes y distintivos de cada caso en cuanto al tratamiento y seguimiento de los requisitos. Se diagramarán los distintos procesos existentes en los casos a fin de determinar que técnicas de la ingeniería de conocimiento están presentes y cuáles no, a fin de detectar falencias y fortalezas en los diferentes procesos hallados.

3. RESULTADOS ESPERADOS

Se esperan detectar patrones comunes y distintivos de cada caso de estudio en cuanto al tratamiento y seguimiento de los requisitos. Además, se pretende desarrollar una herramienta informática para el soporte integral de la ingeniería de requisitos basada en la Ingeniería de conocimiento, que será implementada en una empresa de la muestra a fin de determinar si produce mejora en

tratamientos de requisitos y por lo tanto eleva la calidad del software desarrollado por la firma.

4. FORMACIÓN DE RRHH

El impacto sobre las capacidades institucionales estará garantizado por el contacto con los grupos de investigación del país y el exterior más consolidados.

El proyecto permitirá dar continuidad a la línea de investigación en la temática calidad de software iniciada en el año 2013 y contribuirá a la formación inicial como investigadoras de la Ing. Elena Ríos y la Lic. Lourdes Pralong.

Además, participan en el proyecto dos becarios alumnos de la Licenciatura en Sistemas de Información que inician su formación en la investigación.

5. BIBLIOGRAFÍA

- [1] F. Barletta, M. Pereira, V. Robert, G. Yoguel, Argentina: Dinámica reciente del sector de software y servicios informáticos. 2013.
- [2] R. L. Blanc, L. Lepratte, D. R. Hegglin, R. A. Pietroboni, Empresas de software en Entre Ríos: innovación, exportación, calidad y empleo, *XIX Reunión Anual Red Pymes Mercosur* 2014.
- [3] F. Barletta, M. Pereira, and G. Yoguel, "Impacto de la Política de Apoyo a la Industria de Software y Servicios Informáticos," 2014.
- [4] A. J. Pérez Puletti, "El sector de software y servicios informáticos de Argentina entre 2000 y 2012," 2014.
- [5] R. Sosa Zitto, R. Blanc, L. Pralong, C. Álvarez, and S. Galáz, "El desafío de producir software de calidad aplicando prácticas de CMMI para las pymes de Concepción del Uruguay, Entre Ríos," in *CoNaIISI* 2013.
- [6] T. Clancy, "The standish group chaos report," 2014.
- [7] B. A. Nuseibeh and S. M. Easterbrook, "Requirements Engineering: A Roadmap," in *ICSE 2000 Proceedings of the Conference on The Future of Software Engineering*, 2010.
- [8] A. F. Aguirre, C. Pardo, W. L. Pantoja, M. F. Mejía, and F. J. Pino, "Reporte de Experiencias de la Aplicación de Competisoft en Cinco Mipymes colombianas," *Rev. EIA*, vol. 7. 2013.
- [9] G. Pantaleo, *Calidad en el Desarrollo de Software*. Alfaomega Grupo Editor, 2011.

- [10] R. Sosa Zitto and R. Blanc, "Buenas prácticas de Scrum para alcanzar niveles de calidad en pymes de desarrollo de software," in XVI Workshop de Investigadores en Ciencias de la Computación, 2014, pp. 490–494.
- [11] C. M. Zapata, G. L. Giraldo, and J. E. Mesa, "Una propuesta de metaontología para la educación de requisitos," *Ingeniare. Rev. Chil. Ing.*, vol. 18, no. 26–37, 2010.
- [12] M. A. C. Díaz and J. L. Carrillo-Medina, "Desarrollo de un Sistema Experto para la Especificación de Requerimientos," *Univ. Las Fuerzas armadas extensión Latacunga*, vol. 8, p. 6, 2014.
- [13] E. Kheirkhah and A. Deraman, "Important factors in selecting requirements engineering techniques," in *Proceedings - International Symposium on Information Technology 2008, ITSIm*, 2008, vol. 3.
- [14] J. C. Ramírez Leal, W. J. Giraldo Orozco, and R. Anaya Hernández, "Una propuesta metodológica para mejorar la comunicación en IR," *Rev. EIA*, vol. 13, no. 26, 2017.
- [15] L. Londoño, R. Anaya, and M. Tabares, "Análisis de la Ingeniería de Requisitos Orientada por Aspectos según la Industria del Software," *Rev. EIA*, no. c, pp. 43–52, 2008.
- [16] M. Cristiá, "Introducción a la Ingeniería de Requerimientos," 2011.
- [17] Y. Yang, M. He, M. Li, Q. Wang, and B. Boehm, "Phase distribution of software development effort," in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 2008, pp. 61–69.
- [18] P. Pytel et al., "Propuesta de Aplicación de Técnicas de Representación de Conocimiento en el Análisis de Requisitos Software," *Actas 1er Seminario Argentina-Brasil de TIC*, 2011.
- [19] R. Belschner, "Evaluation of Real-Time Requirements by Simulation Based Analysis," in *IFAC Proceedings Volumes*, 1996.
- [20] J. Karlsson and K. Ryan, "A Cost-Value Approach for Prioritizing Requirements," *IEEE Softw.*, vol. 14, no. 5, pp. 67–74, 1997.
- [21] Ho-Won Jung, "Optimizing value and cost in requirements analysis," *IEEE* 2002.
- [22] P. (Pericles) Loucopoulos, V. Karakostas, and B. Καρακώστας, *System requirements engineering*. McGraw-Hill Book Co, 1995.
- [23] V. Ambriola and V. Gervasi, "An environment for cooperative construction of natural-language requirement bases," in *Proceedings 8th Conference on Software Engineering Environments*, 2002, pp. 124–130.
- [24] J. Misra, "Terminological inconsistency analysis of natural language requirements," *Inf. Softw. Technol.* 2016.
- [25] L. E. P. Valencia, A. T. Lazo, and L. C. Benjumea, "La relación existente entre la calidad del software y el uso de modelos internacionales vs el uso de modelos autóctonos: el caso de las pymes en el eje cafetero-Colombia," *Informática*, vol. 1, no. 1, 2013.
- [26] J. Capote, C. J. Llanten Astaiza, C. J. Pardo Calvache, A. de J. González Ramírez, and C. A. Collazos, "Gestión del conocimiento como apoyo para la mejora de procesos software en las micro, pequeñas y medianas empresas," *Rev. Ing. e Investig.*, vol. 28. 2008.
- [27] C. Hinojosa, G. Raura, and E. R. Fonseca, "El Aporte de la Gestión del Conocimiento a la Ingeniería de Requisitos," *Univ. Las Fuerzas armadas extensión Latacunga*, vol. 8, p. 102, 2014.
- [28] A. Sebastián and G. D. S. Hadad, "Mejoras a un modelo léxico mediante mapas conceptuales," 2015.
- [29] S. E. Gordon, "Implications of Cognitive Theory for Knowledge Acquisition," in *The Psychology of Expertise*, New York, NY: Springer New York, 1992, pp. 99–120.
- [30] K. Hoffman, RR and Coffey, JW and Ford, "A case study in the research paradigm of human-centered computing: Local expertise in weather forecasting," 2000.
- [31] R. R. Hoffman and G. Lintern, "Eliciting and Representing the Knowledge of Experts," in *The Cambridge Handbook of Expertise and Expert Performance*, 2006.
- [32] W. M. Wang, C. F. Cheung, W. B. Lee, and S. K. Kwok, "Self-associated concept mapping for representation, elicitation and inference of knowledge," *Knowledge-Based Syst.*, vol. 21, no. 1, pp. 52–61, 2008.