

Revisión de las fuentes usadas para elicitar requerimientos

Leandro Antonelli¹, Alejandro Oliveros²

¹LIFIA, Facultad de Informática, Universidad Nacional de La Plata, lanto@sol.info.unlp.edu.ar

² Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires y Magíster de Ingeniería de Software, Facultad de Informática, UNLP, oliveros@fibertel.com.ar

Abstract

Los requerimientos son importantes en el desarrollo de software puesto que es inútil un sistema que no cumple con los requerimientos del cliente. Para descubrir estas necesidades se debe elegir a las fuentes adecuadas y relevar apropiadamente los requerimientos. Es importante también una vez establecidos los requerimientos rastrear sus fuentes para validar los requerimientos, resolver conflictos o ambigüedades. Este artículo ofrece un resumen de la literatura de fuentes de requerimientos: se sintetizan las fuentes utilizadas con sus características. Este resumen permite categorizar las fuentes para saber cual elegir en el proceso de elicitación y para poder realizar un modelo de ellas a fin de poder rastrearlas una vez concluida la especificación de requerimientos.

Keywords: requerimientos, traceability, fuentes

Incumbencia: pertenece al congreso en general, no es de ningún workshop

1. Introducción

Los requerimientos son de importancia en el desarrollo de un sistema de software puesto que la calidad del software depende de ellos. Garvin [Garvin 1984] define calidad como “un concepto complejo y multifacético que puede describirse desde cinco perspectivas: visión trascendental, del usuario, de manufactura, del producto y basada en el valor”. Tanto la visión del usuario como la de la manufactura relacionan calidad a los requerimientos. Davis [Davis 1993] muestra un diagrama del ciclo de vida del software en el cual los requerimientos se colocan como la materia prima para realizar las pruebas de aceptación del software. Esto significa que un producto de software es aceptado por el cliente (es de calidad para él) si cumple con los requerimientos establecidos. Boehm [Boehm 2001] establece el costo relativo de reparar errores a lo largo del ciclo de vida de desarrollo del software. Este costo tiene el valor más bajo en la etapa de requerimientos y crece en forma exponencial en las sucesivas etapas. Un error que no se soluciona en la etapa de requerimientos puede costar entre 100 y 200 veces más en la etapa de mantenimiento. Mizuno organizó los errores que se pueden producir a lo largo del desarrollo del software en un modelo de catarata de errores [Mizuno 1983]. El modelo muestra que una especificación de requerimientos incorrecta puede ocasionar errores ocultos dentro del sistema que son difíciles de hallar. Pero las palabras de Ackoff [Ackoff 1974] son las que mejor sintetizan la importancia de los requerimientos: “Fallamos más a menudo porque resolvemos el problema incorrecto, que porque obtenemos una solución inadecuada del problema correcto”.

Los requerimientos no son estáticos. La realidad cambia al igual que lo hace el dominio que alojará el sistema de software. Si el dominio cambia, también lo deben hacer los requerimientos, para que el software se adapte a la nueva situación.

Los cambios de los requerimientos se deben propagar por todos los productos del ciclo de vida de desarrollo del software hasta la aplicación. Para propagar estos cambios, es necesario rastrear las distintas transformaciones que sufren los requerimientos por todos los productos del desarrollo del software. Esto se conoce como rastreabilidad (traceability) de requerimientos: “habilidad de describir y seguir la vida de los requerimientos en ambas direcciones hacia delante y

hacia atrás (forward and backward traceability). Desde los orígenes, pasando por la especificación y desarrollo, hacia su posterior entrega y uso, y a través de todos los períodos de refinamiento e iteración de cualquiera de estas etapas.” [Gotel 1994].

La rastreabilidad de requerimientos es una característica que deben poseer los requerimientos (la especificación de requerimientos de software) y su gerenciamiento [Katonya 1998] [Davis 1993] [Hilburn 1999]. Se necesita rastreabilidad de requerimientos tanto para el cliente como para el equipo de desarrollo [Wieringa 1995]. El cliente necesita conocer en que parte del software se implementa cada requerimiento para validarlos. Y por su parte, el gerente del proyecto necesita conocer como se operacionalizan los requerimientos para administrar el proceso de desarrollo.

Traceability se puede dar en ambos sentidos: hacia adelante (implementación) y hacia atrás (fuentes) [Davis 1993] [Gotel 1994]. Al gerente del proyecto le resulta útil traceability hacia las fuentes, puesto que permite conocer la justificación de los requerimientos y de las decisiones tomadas [Wieringa 1995].

Rastreabilidad de requerimientos permanece como un área de problema ampliamente reportada, a pesar de las muchas soluciones se han propuesto [Pinheiro 2000] [Katonya 1998] [Wieringa 1995] [Gotel 1994]. El problema se adjudica a una falla en los aspectos previos a la inclusión de los requerimientos en el documento de especificación de requerimientos. Este problema se conoce como rastreabilidad pre-especificación de requerimientos hacia atrás (pre-requirements specification backward traceability) [Gotel 1994].

Consideramos que es necesario realizar un análisis de las fuentes de requerimientos utilizadas en los desarrollos de software como primer paso para poder construir un modelo de traceability hacia esas fuentes.

Este artículo analiza la literatura y ofrece un resumen de las fuentes de requerimientos utilizadas y sus características. Esto permite realizar un posterior modelo de traceability hacia las fuentes. Este análisis también sirve de guía al ingeniero de requerimientos en el proceso de selección de la fuente adecuada para el proceso de elicitación de requerimientos.

Este trabajo está dividido en dos partes. En la sección 2 se muestra un resumen de la postura de cada autor respecto de las fuentes de requerimientos que utiliza y sus características (se adjunta un Anexo con un cuadro que resume la postura de cada autor). Y la sección 3 sintetiza las distintas fuentes con sus características.

2. Fuentes de requerimientos según los autores

En esta sección se presenta un resumen de las fuentes que utiliza cada autor para elicitar requerimientos. En el anexo I se sintetiza esta información en un cuadro.

Jitnah [Jitnah 1995] considera que la principal fuente de información son los usuarios de una organización, a pesar de que considera que no son adecuados para elicitar requerimientos y de que los requerimientos se obtienen en una forma inapropiada para los desarrolladores.

Valenti [Valenti 1998] utiliza en cambio a los expertos del dominio, de los cuales obtiene información en forma oral. Laguna [Laguna 2001] sigue la línea de Valenti, elicita de los expertos del dominio información oral que no tiene ningún tipo de estructura. Bostrom [Bostrom 1983] también coincide en que la principal fuente son los expertos del dominio y él propone un formato de entrevista para llevar a cabo el proceso de elicitación.

Finkelstein [Finkelstein 1994] elicita de las personas, pero no solamente de los usuarios o los expertos del dominio, él considera que se debe elicitar de los usuarios, los clientes o cualquier otro beneficiario. Finkelstein encuentra un inconveniente al elicitar de las personas (al igual que Jitnah) que es la falta de consenso entre ellos. Para ello, Finkelstein considera imprescindible el trabajo colaborativo para lograr consenso y negociar los requerimientos. Land [Land 2001] tiene una visión similar a la de Finkelstein y propone elicitar de las personas, ya sea como individuos o

grupos. Él propone como técnicas individuales: entrevistas estructuras y no estructuradas, análisis de protocolo, cuestionarios y encuestas. Y como técnicas colectivas propone: brainstorming, reuniones de inspección de software, aprendizaje de historias, sistemas de soporte de grupo y elicitación de conocimiento basado en eventos.

France [France 1995] sigue la línea de elicitar desde los expertos del dominio. Él utiliza entrevistas grupales en las que participan expertos con diferentes perspectivas. Sin embargo, no se limita a expertos, también considera incluir personal de áreas de producción y marketing. Sugiere entrevistas de una hora en la cual participen cuatro o cinco personas. A diferencia de Valenti y Laguna quienes elicitan información oral, France alienta a que los expertos modelen los conceptos utilizando su propia notación. Esto permite lograr un mejor canal de comunicación puesto que los desarrolladores aprenden el lenguaje del dominio.

Leite [Leite 1995] extrae requerimientos de los clientes del macrosistema a través de entrevistas [Leite 1996]. Esta información es utilizada para construir el Client Oriented Requirements Baseline, el cual está compuesto por el LEL y Escenarios. LEL tiene como objetivo capturar el lenguaje del dominio, mientras que los escenarios se ocupan de los aspectos dinámicos.

Miura reporta tres casos de estudio en los que desarrolla entrevistas con los clientes y usuarios [Miura 1995]. En cada caso realizó 3, 4 y 5 entrevistas, de aproximadamente 5, 12 y 4 horas de duración. Como resultado se produjo 8, 7 y 15 hojas de reporte tamaño A4.

Goguen coincide en elicitar de las personas. Sin embargo, tiene una visión más amplia de la información que se debe obtener, puesto que no captura solamente lo que expresan oralmente y se ocupa de lo que expresan tácitamente [Goguen 1993]. Instille [Instille 2002] sigue la línea de Goguen. Si bien plantea que las necesidades se extraen de los usuarios personalmente a través de entrevistas o focus group, considera que los usuarios saben más de lo que pueden decir en una o varias entrevistas. Por lo tanto, hace falta realizar observación de los mismos a través de visitas a los sitios de trabajo o por medio del análisis de fotografías y videos. Macaulay [Macaulay 1996] también sigue la línea de Goguen, si bien elicita requerimientos de los usuarios a través de entrevistas y cuestionarios, aconseja realizar observación.

Ransley [Ransley 2003] propone elicitar de los dueños o stakeholders, puesto que son ellos quienes imponen restricciones o necesitan el sistema. Pero reconoce que estas restricciones pueden venir de fuentes escritas, por ejemplo una regulación (ley provincial o nacional, por ejemplo) o una business rule. Además plantea distintos niveles de fuentes, puesto que una puede referenciar a otra en un estado más puro. Por ejemplo, una persona o documento puede referenciar a un memo, a una reunión o una sesión de casos de usos. Rook [Rook 1988] tiene una concepción similar a la de Ransley. Rook propone elicitar desde los expertos del dominio, doctrinas u otras fuentes de procedimientos, metodologías o lineamientos.

Drake [Drake 1997] plantea elicitar de documentos y personas. Recomienda comenzar con los documentos para luego identificar las fuentes de los mismos. Drake propone una jerarquía de fuentes que contiene en un primer nivel a las personas, material crudo y documentación de referencia. Subclasifica a las personas en clientes y staff (junior y senior). Como documentación de referencia menciona instrucciones, libros y artículos.

Lubars en [Lubars 1997] describe un caso real que se ajusta al modelo propuesto por Drake. Lubars realizó un trabajo para el gobierno norteamericano en el cual comenzaron elicitando de un documento de 529 páginas escrito en inglés con algunos diagramas de página completa y muchas tablas. El documento no contenía ningún diagrama de flujo de datos, diagramas entidad-relación o alguna otra forma de notación abstracta comúnmente utilizada en ingeniería de software. Además del documento, Lubars obtuvo información de un curso de entrenamiento dictado por un experto del dominio. Tanto el instructor del curso como otros miembros estuvieron disponibles a lo largo del proceso de captura de requerimientos para brindar más información.

Blyth [Blyth 1993] incorpora el sistema de software a las fuentes de requerimientos. Propone elicitar de los dueños de los problemas requerimientos en lenguaje natural sobre mejoras al sistema actual. Van Lamsweerde [Van Lamsweerde 2000] también propone elicitar en función de un sistema de software existente y entrevistar a los stakeholders.

Holibaugh [Holibaugh 1992] sigue la línea de utilizar el software existente. Comienza analizando el software, luego entrevista a los expertos del dominio para obtener información de alto nivel y verificar la información de los sistemas. Sin embargo, Holibaugh no sólo analiza software, sino que lo hace con cualquier artefacto del desarrollo de software, aunque también utiliza cualquier material de referencia y de entrenamiento del dominio.

Iscoe [Iscoe 1993] propone obtener conocimiento de sistemas y expertos como lo hacen Blyth, Van Lamsweerde y Holibaugh. Pero, a diferencia de ellos, propone primero entrevistar a los expertos (aunque reconoce que es un proceso laborioso, costoso y propenso a errores) para luego realizar ingeniería inversa de todas las aplicaciones (recurriendo a los expertos cuando sea necesario para resolver conflictos y proveer información faltante).

Arango [Arango 1994] recomienda elicitar desde los expertos y de las aplicaciones. Coincide con Holibaugh en elicitar desde los expertos información general y recurrir a las aplicaciones para obtener información detallada. Lam [Lam 1997] propone abordar los requerimientos desde dos perspectivas: examinar la documentación del sistema y hablar con los expertos del dominio.

Chung [Chung 1995] reporta una experiencia en donde elicató requerimientos no funcionales. En una de sus prácticas elicató desde documentos de desarrollo de software: documentos de requerimientos, diseño e implementación. En otra se basó en información de carga de trabajo (workload) e información de la organización (su operación y sus sistemas). En un tercer trabajo, sólo trabajaron con información pública y general (por ejemplo reportes anuales).

Hoffmann [Hoffmann 2001] tiene la visión más amplia y él aconseja elicitar de cualquiera sea la fuente. Reconoce que típicamente se elicita de expertos, repositorios o de aplicaciones de software.

Loucopoulus [Loucopoulus 1995] clasifica las fuentes en: expertos del dominio, literatura sobre el dominio, software existente sobre el dominio, software sobre otro dominio, estándares nacionales e internacionales y otros stakeholders. Reconoce que las personas presentan varias dificultades. Pueden no tener una idea clara o no saber como describir lo que necesitan. Pueden utilizar una terminología orientada al dominio que los analistas no comprenden por utilizar un lenguaje orientado a la computadora. O bien les puede disgustar la idea de tener que usar un nuevo sistema de software.

3. Resumen de las fuentes

La fuente mencionada por casi todos los autores son las personas [Land 2001] [Goguen 1993], si bien hay diferencias en cuanto a los distintos roles que desempeñan. Pueden ser el:

- cliente o dueño del problema [Finkelstein 1994] [Leite 1995] [Miura 1995] [Ransley 2003] [Drake 1997] [Blyth 1993];
- los usuarios del sistema informatizado [Jitnah 1995] [Finkelstein 1994] [Miura 1995] [Instille 2002] [Macaulay 1996];
- los expertos del dominio [Valenti 1998] [Laguna 2001] [Bostrom 1983] [France 1995] [Rook 1988] [Iscoe 1993] [Arango 1994] [Lam 1997] [Hoffmann 2001];
- cualquier otro stakeholder [Drake 1997] [Van Lamsweerde 2000] [Ransley 2003] [Finkelstein 1994].

Sin embargo, las personas no son las fuentes más adecuadas [Loucopoulus 95] [Jitnah 1995]. En primer lugar porque las personas y los ingenieros de requerimientos utilizan distintos lenguajes. Los stakeholders utilizan un lenguaje orientado al dominio mientras que los ingenieros de

requerimientos utilizan un lenguaje orientado al sistema informatizado [Loucopoulus 1995]. Además, a las personas les puede resultar difícil expresar las ideas [Loucopoulus 1995]. Ellos tienen un lenguaje tácito que el ingeniero de requerimientos debe interpretar [Goguen 1993] [Instille 2002]. Otro motivo de la dificultad de elicitar de las personas es la falta de consenso [Finkelstein 1994], cada uno de los stakeholder puede tener requerimientos que entran en conflicto con otros. En último lugar, puede ser difícil elicitar de las personas porque no tengan interés o simplemente aversión al nuevo sistema [Loucopoulus 1995].

También son objeto de críticas las técnicas utilizadas para obtener información de las personas y a los productos obtenidos de ellas. Jitnah [Jitnah 1995] sostiene que se obtiene información en forma inapropiada. Mientras que Iscoe [Iscoe 1993] considera que las entrevistas, la principal técnica de elicitación, como un proceso laborioso, costoso y propenso a errores.

Como los stakeholders y el ingeniero de requerimientos utilizan distintos lenguajes, varios autores [Laguna 2001] [Valenti 1998] [Blyth 1993] recomiendan utilizar uno comprensible por ambos en lugar de introducir formalismos de algún tipo. Ellos recomiendan utilizar información oral, en general totalmente libre y desestructurada. Otros autores aconsejan utilizar el lenguaje del stakeholder [France 1995] [Leite 1995]. Leite plantea capturar el lenguaje a través de la conformación del LEL.

A los stakeholders les puede resultar difícil expresarse oralmente, por lo que se debe recurrir a su lenguaje tácito. Se puede obtener esta información tácita a través de la observación de las conductas de las personas [Goguen 1993] [Macaulay 1996] [Instille 2002]. Este proceso se puede realizar “in situ” pero no es la única alternativa. También es posible analizar filmaciones de video o fotografías [Instille 2002].

Una única persona no siempre puede determinar los requerimientos, puede ser necesaria la aprobación conjunta o el consenso de todos ellos [Finkelstein 1994] [Land 2001].

Además de las personas, la segunda fuente para obtención de requerimientos la conforman los productos que pudieran producir las mismas personas. Ransley [Ransley 2003] enmarca dentro de esta categoría a las regulaciones y reglas de negocios. Por su parte, Holibaugh [Holibaugh 1993] se refiere a material de entrenamiento de referencia del dominio (domain reference material training). Rook [Rook 1988] menciona doctrinas u otras fuentes de procedimientos, metodologías o lineamientos. Drake [Drake 1997] utiliza documentación de referencias como ser instrucciones, libros y artículos.

La tercera y última fuente de requerimientos las constituyen los productos provenientes del proceso de desarrollos de software previos. Algunos autores consideran a sistemas de software como fuente principal y comienzan elicitando de ellos [Holibaugh 1993] [Van Lamsweerde 2000]. Otros la utilizan los sistemas de software como información de soporte [Blyth 1993] [Iscoe 1993] [Arango 1994] [Hoffmann 2001].

Sin embargo, no solamente se puede hacer ingeniería reversa de un sistema ya desarrollado, se pueden utilizar otros productos de desarrollos previos con el objetivo de reutilizar requerimientos o análisis de dominios previos [Holibaugh 1993] [Lam 1995] [Chung 1997].

Cabe mencionar que hay distintos niveles entre las fuentes. Ransley [Ransley 2003] hace esta distinción. Él establece los datos fuentes (source data) que pueden ser memos de gerentes, minutas de reuniones, casos de uso obtenidos a partir de sesiones de trabajo. Y por otro lado, el material de fuente completa (full source material), lo cual refiere a los gerentes que escribieron el memo, a las reuniones de la cual se elaboró las minutas o que permitieron obtener los casos de uso. Drake [Drake 1997] tiene un enfoque similar. Él comienza su elicitación de requerimientos analizando los documentos para luego rastrear los creadores de estas fuentes.

4. Conclusiones

La primera conclusión que se puede extraer del análisis realizado es la escasa presencia de trabajos específicos sobre las fuentes ha utilizar para la elaboración de los requerimientos, sus características y las técnicas a aplicar en el proceso de elicitación de requerimientos. En segundo lugar se puede concluir que la temática de las fuentes de obtención de los requerimientos aparece en múltiples trabajos del área. Efectivamente una revisión intensa de las fuentes ha permitido detectar un número considerable de trabajos que analizan la problemática de las fuentes los que, aunque no consideramos agotada la búsqueda, podemos concluir que en general los tratamientos del tema son de escasa profundidad y se les concede poca relevancia. En ese sentido, este trabajo ofrece una guía al ingeniero de requerimientos sobre la literatura acerca de qué fuentes utilizar y cómo abordarlas.

A pesar de las dificultades que ocasiona la elicitación de requerimientos a partir de las personas, este recurso constituye la fuente más comúnmente utilizada, esto es confirmado por los survey realizados en [Antonelli 2002] y [Antonelli 2003]. El material producido por las personas es la segunda de las fuentes y por último el conocimiento encerrado en desarrollos de software anteriores. Las menciones a estas fuentes son poco precisas y generalmente apelan a mecanismos de interacción que podríamos denominar “generales”, tales como entrevistas, reuniones, brainstorming, etc.

El conocimiento disponible en el terreno de las fuentes de obtención de los requerimientos requiere mayor precisión en su identificación, caracterización y clasificación. También es necesario establecer cuáles son los soportes reales de ese conocimiento y la forma en que reflejan las relaciones entre las distintas áreas de conocimiento que debe capturar el proceso de elicitación.

5. Futuros trabajos

Como se mencionó el objetivo final consiste en establecer la rastreabilidad desde la Especificación de Requerimientos de Software (SRS) hasta las fuentes de los requerimientos (*backward traceability*). El primer paso, actualmente en curso, consiste en construir un modelo de esas fuentes y las relaciones entre ellas, sus soportes y con la SRS. Sobre la base de ese modelo se construirá un enfoque de traceability que, en primer lugar, refleje la relación estática entre las fuentes y la SRS y, posteriormente incorpore la dinámica de la traceability en el proceso de desarrollo. Uno de los objetivos de la *backward traceability* en el momento del desarrollo del software es soportar la validación de los requerimientos.

Por último, una vez liberado el sistema software, la *backward traceability* debe estar en condiciones de soportar la evolución, así como en el momento de encarar procesos de reemplazo o reingeniería del producto software.

Referencias

[Ackoff 1974] Ackoff, R.: Redesigning The Future, Wiley publishing, (1974)

[Antonelli 2002] Antonelli, L., Oliveros, A.: “Fuentes utilizadas por desarrolladores de software en Argentina para elicitar requerimientos”, V Workshop on Requirements Engineering, WER’2002, ISBN 84-96023-01-X, Valencia, España, 11 y 12 de noviembre, (2002), 1-19.

[Antonelli 2003] Antonelli, L., Oliveros, A., Rossi, G.: “Traceability en la elicitación y especificación de requerimientos”, Tesis de Magíster, Facultad de Informática, UNLP, Argentina, (2003).

[Arango 1994] Arango, G.: “A brief introduction to domain analysis” Proceedings of the 1994 ACM symposium on Applied computing, Phoenix, Arizona, United States, ISBN:0-89791-647-6, ACM Press, (1994), 42 – 46.

[Blyth 1993] Blyth, A. J. C., Chudge, J., Dobson, J. E., Strens, M. R.: “ORDIT: a new methodology to assist in the process of eliciting and modelling organizational requirements”, Conference on Supporting Group Work, Proceedings of the conference on Organizational

computing systems, Milpitas, California, United States, ISBN:0-89791-627-1, ACM Press, (1993), 216 – 227.

[Boehm 2001] Boehm, B., Basili, V. R.: “Software defect reduction top 10 list”, IEEE Computer, Vol. 34, No. 1, Enero, (2001).

[Bostrom 1983] Bostrom, R. P.: "Achieving Excellence in Communication: A Key to Developing Complete, Accurate and Shared Information Requirements", Proceedings of the Twentieth Annual Computer Personnel Research Conference, edited by Elias M. Awad, New York, ISBN, O-89791-122-9, Vol II, (1983), 1-13.

[Chung 1995] Chung, L., Nixon, B. A.: “Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach”, International Conference on Software Engineering, Proceedings of the 17th international conference on Software engineering, Seattle, Washington, United States, ISBN:0-89791-708-1, ACM Press, (1995), 25 – 37.

[Davis 1993] Davis, A. Software Requirements Objects, functions and states, PrenticeHall international inc, 1993.

[Drake 1997] Drake, J. M., Xie, W. W., Tsai, W. T., Zualkernan, I. A.: “Approach and Case Study of Requirement Analysis Where End Users Take an Active Role”, International Conference on Software Engineering, Proceedings of the 15th international conference on Software Engineering, Baltimore, Maryland, United States, Los Alamitos, CA, USA, ISBN:0-89791-588-7, IEEE Computer Society Press, (1997), 177 – 186.

[Finkelstein 1994] Finkelstein, A.: “Requirements Engineering: a review and research agenda”, Proceeding 1st Asian & Pacific Software Engineering Conference, IEEE Computer Science Press, (1994), 10-19.

[France 1995] France, R. B., Horton, T. B.: “Applying Domain Analysis and Modeling: An Industrial Experience”, Symposium on Software Reusability, Proceedings of the 1995 Symposium on Software reusability, Seattle, Washington, United States, ISSN:0163-5948, ACM Press, (1995), 206 – 214.

[Garvin 1984] Garvin, D.: “What does product quality really mean?”, Sloan Management Review, Vol 26, No 1, (1984), 25-33.

[Goguen 1993] Goguen, J. A., Linde, C.: “Techniques for requirements elicitation”, Proceedings IEEE International Symposium on Requirements Engineering, San Diego, CA, IEEE, New York, January (1993), 152-164.

[Gotel 1994] Gotel, O.C.Z., Finkelstein, A.C.W.: “An Analysis of the Requirements Traceability Problem”, International Conference on Requirements Engineering, ICRE'94, Los Alamitos, California, Abril, (1994), 94-101.

[Hilburn 1999] Hilburn, T.B., Hirmanpour, I., Khajenoori, S., Turner, R., Qasem, A.: “A Software Engineering Body of Knowledge”, Version 1.0, SEI, Technical Report, CMU/SEI-99-TR-004, ESC-TR-99-004 (1999).

[Hoffmann 2001] Hoffmann, H. F., Lehner, F., “Requirements Engineering as a success factor in software projects”, IEEE Software, July/Agust, Vol 18, Number 4, IEEE computer society, (2001), 58-66.

[Holibaugh 1993] Holibaugh, R.: Object-Oriented Domain Analysis Method (JODA), Special Report, Version 3.1, Joint Integrated Avionics Working Group (JIAWG), CMU/SEI-92-SR-3, November (1993).

[IEEE 1990] IEEE-Std-610 (1990).

[Instille 2002] Intille, S., Kukla, C., Ma, X.: “Eliciting user preferences using image-based experience sampling and reflection”, Conference on Human Factors and Computing Systems, Minneapolis, Minnesota, USA, ACM Press, ISBN:1-58113-454-1, (2002), 738 - 739

[Iscoe 1993] Iscoe, N.: "Domain Modeling Overview & Ongoing Research at EDS", Proceedings of the 15th international conference on Software Engineering, IEEE, 0270-5257/93, May (1997).

[Jitnah 1995] Jitnah, D., Han, J., Steele, P. "Software Requirements Engineering: An Overview", Technical Report 95-04, Peninsula School of Computing and Information Technology, Monash University, Melbourne, Australia, September (1995).

[Katonya 1998] Katonya, G., Sommerville, I.: Requirements Engineering Processes and Techniques, Chichester, John Wiley & Sons (1998).

[Laguna 2001] Laguna, M. A., Marqués, J. M., García, F. J.: "A user requirements elicitation tool", ACM Press, Volume 26, Issue 2 March, ISSN:0163-5948, (2001), 35 - 37.

[Lam 1997] Lam, W., McDermid, J. A.: "Ten Steps Towards Systematic Requirements Reuse", Symposium on Software Reusability, Proceedings of the 1997 symposium on Software reusability, Boston, Massachusetts, United States, ACM Press, ISSN:0163-5948, (1997), 54 - 64.

[Land 2001] Land, L. P. W., Aurum, A., Handzic, M.: "Capturing Implicit Software Engineering Knowledge", Proceedings of the 13th Australian Software Engineering Conference (ASWEC'01), 1530-0803/01 IEEE, (2001).

[Leite 1995] Leite, J. C. S. P., Oliveira, A. P. A.: "A Client Oriented Requirements Baseline", Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE '95), 0-8186-7017-7/95, IEEE, (1995).

[Leite 1996] Leite, J. C. S. P., Gilvaz, A. P. P.: "Requirements Elicitation Driven by Interviews: The Use of Viewpoints", Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD '96), 1063-6765/96 IEEE, (1996).

[Loucopoulos 95] Loucopoulos, P., Karakostas, V.: "System Requirements Engineering", McGraw-Hill International Series in Software Engineering, ISBN 0-07-707843-8, (1995).

[Lubars 1997] Lubars, M., Potts, C., Richter, C.: "Developing initial OOA models" IEEE Computer Society Press, ISBN:0-89791-588-7, Los Alamitos, CA, USA, (1997), 255 - 264.

[Macaulay 1996] Macaulay, L., "Requirements for Requirements Engineering Techniques", Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96), 0-8186-7252-8/96 IEEE, (1996).

[Miura 1995] Miura, N., Kaiya, H., Saeki, M.: "Building the Structure of Specification Documents from Utterances of Requirements Elicitation Meetings", Proceedings of the 1995 Asia Pacific Software Engineering Conference (APSEC '95), 0-8186-7171-8/95 IEEE, (1995).

[Mizuno 1983] Mizuno Y.: "Software Quality Improvement", IEEE Computer, Vol. 16, No. 3, March (1983), 66-72.

[Pinheiro 2000] Pinheiro, F. A. C.: "Formal and Informal Aspects of Requirements Tracing", III Workshop de Engenharia de Requisitos, WER'2000, Rio de Janeiro, Brasil, Julio (2000)

[Ransley 2003] Ransley, P., "White Paper: A Primer on Requirements Engineering & Management", Version 0.2, Beaver Computer Consultants Ltd's Requirements Engineering & Management Services Capability, www.beaver-consulting.co.uk/requirement_primer.pdf, Mayo (2003).

[Rook 1988] Rook, F. W., Croghan, J. W.: "Knowledge acquisition for AI system requirements analysis: A systems engineering methodology", International conference on Industrial and engineering applications of artificial intelligence and expert systems, Proceedings of the first international conference on Industrial and engineering applications of artificial intelligence and expert systems, Volume 2, Tullahoma, Tennessee, United States, ISBN:0-89791-271-3, ACM Press, (1988) 798 - 804.

[Valenti 1998] Valenti, S., Panti, M., Cucchiarelli, A.: "Overcoming communication obstacles in user-analyst interaction for functional requirements elicitation", ACM SIGSOFT Software Engineering Notes, Volume 23, Issue 1 January, ACM Press, ISSN:0163-5948, (1998), 50 - 55.

[Van Lamsweerde 2000] Van Lamsweerde, A.: “Requirements engineering in the year 00: a research perspective”, International Conference on Software Engineering archive, Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, ACM Press, ISBN:1-58113-206-9, (2000), 5 - 19.

[Wieringa 1995] Wieringa, R.: “An Introduction to Requirements Traceability”, Reporte Técnico, Faculty of Mathematics and Computer Science, Vrije Universiteit, Esprit Project 2RARE, Noviembre 1, (1995).

Anexo I – Resumen de fuentes y sus características

Autor	Fuente	Observaciones sobre la fuente	Se obtiene de la fuente	Técnica aplicada a la fuente
Jitnah	Usuarios	Inadecuada, se obtiene información en forma inapropiada		
Valenti	Expertos del dominio		Información oral	
Laguna	Expertos del dominio		Información oral no estructurada	
Bostrom	Expertos del dominio			Entrevistas estructurada
Finkelstein	Usuarios, clientes u otro beneficiario	Es difícil elicitar de las personas por su falta de consenso		Trabajo colaborativo para fomentar consenso
Land	Personas como individuos o grupos.			Técnicas individuales: entrevistas estructuras y no estructuradas, análisis de protocolo, cuestionarios y encuestas. Técnicas colectivas: brainstorming, reuniones de inspección de software, aprendiendo historias, sistemas de soporte de grupo, elicitación de conocimiento basado en eventos.
France	Expertos del dominio		Anotaciones de los expertos utilizando su propia notación	Entrevistas grupales de 1 hora, con 4 o 5 personas de diferente perspectiva, por ejemplo expertos, personal de producción y marketing.
Leite	Clientes del macrosistema		LEL y escenarios	Construye el client oriented requirements baseline a partir de entrevistas
Miura	Entrevistas con clientes y usuarios.		Produjo reportes de 8, 7 y 15 hojas	3, 4 y 5 entrevistas de casi 5, 12, y 4 horas de duración,
Goguen	Personas	No pueden expresar todo su conocimiento oralmente	Lenguaje tácito	
Instille	Usuarios	Los usuarios saben más de lo que pueden decir en una o varias entrevistas	Información tácita	Entrevistas o focus group. Observación directa u observación de fotografías y videos
Macaulay	usuarios			Entrevistas, cuestionarios u observación

Ransley	Personas: Dueños del problema, stakeholders. Documentos: regulaciones o business rules	Distintos niveles de fuentes, los documentos pueden ser: referencia a un memo, a una reunión o una sesión de casos de usos		
Rook	Expertos del dominio, doctrinas u otra fuente de procedimientos, metodologías o lineamientos			
Drake	Clientes, staff (junior y senior), material crudo y documentación de referencia (instrucciones, libros y artículos)			Se analizan los documentos y se rastrea las fuentes utilizadas para construir los documentos
Lubars	Documento con diagramas y tablas. Curso de entrenamiento. Interacción directa con el instructor y otros miembros.			
Blyth	Dueño del problema, sistema de software		Requerimientos en lenguaje natural	Se elicitan de los dueños de los problemas mejoras del sistema actual
Van Lamsweerde	Sistema de software, Stakeholders			Analizar los sistemas de software existente y luego entrevistar a los stakeholders para abstraer el conocimiento.
Holibaugh	Software y producto del desarrollo, material de referencia y de entrenamiento.			Comienza analizando el software, luego entrevista a los expertos del dominio para obtener información de alto nivel y verificar la información de los sistemas.
Iscoe	Expertos y sistemas de software.	Las entrevistas son un trabajo laborioso, costoso y propenso a errores.		Se entrevistan a los expertos para luego realizar ingeniería inversa y luego recurrir a los expertos para resolver problemas.

Arango	Expertos y aplicaciones			Los expertos proveen información general y se recurren a las aplicaciones para obtener información detallada
Lam	Documentación del sistema y expertos del dominio			
Chung	Descripción del sistema de información. Documentos de requerimientos, diseño e implementación. Registros de workload		Requerimientos no funcionales	
Hoffmann	Cualquiera sea la fuente: expertos, repositorios o de aplicaciones de software.			
Loucopoulus	Expertos, literatura, software del dominio, software de otros dominios estándares y otros stakeholders.	Las personas presentan dificultades. Pueden no tener una idea clara o no saber describir lo que necesitan. Pueden utilizar una terminología distinta a la de los analistas. Les puede disgustar la idea del nuevo sistema de software.		