

Completitud de Glosarios: un Estudio Experimental

Jorge H. Doorn ⁽¹⁾⁽²⁾, Marcela Ridaó ⁽¹⁾

⁽¹⁾ INTIA, Facultad de Ciencias Exactas - Univ. Nacional del Centro de la Provincia de Buenos Aires

⁽²⁾ Universidad Tecnológica Nacional, FRBA
email: {jdoorn, mridao}@exa.unicen.edu.ar

Abstract. La introducción de técnicas de predicción basadas en modelos estadísticos en el campo de la Ingeniería de Software lleva ya varios años, especialmente en la estimación de errores residuales luego de una actividad de prueba de un sistema o de defectos totales, detectados y no detectados, durante una actividad de inspección. Estas técnicas son muy valiosas en aquellas áreas en las que la estimación de la completitud resulta un problema prácticamente inabordable de otra manera. En el presente artículo se estudia experimentalmente el uso de los datos de captura y recaptura en el proceso de la Ingeniería de Requisitos, en particular en la construcción del Léxico Extendido del Lenguaje. Los resultados de esta experiencia inicial son muy promisorios e indican que ésta es un área en la que se debieran realizar nuevos estudios a los efectos de corroborar los datos disponibles y consolidar el conocimiento acerca de la capacidad predictiva de estas técnicas.

1 Introducción

Son de amplio conocimiento las dificultades que deben enfrentar los desarrolladores de software para entender y elicitar las necesidades de los clientes-usuarios. Cuanto más complejo es el contexto del problema, más difícil se hace la elicitación de los requisitos de software. A menudo, los ingenieros de requisitos deben convertirse casi en expertos en el dominio del problema durante la adquisición de conocimiento del contexto de la aplicación. El objetivo de la Ingeniería de Requisitos (IR) es sistematizar el proceso de definición de requisitos [1] [2] [3] [4], así como ganar un compromiso entre los clientes-usuarios y los ingenieros de requisitos puesto que ambos deberían participar y colaborar en ese proceso. El proceso de IR consiste en tres actividades principales: elicitación, modelado y análisis del dominio de la aplicación [5] [6], y una actividad adicional: la administración de requisitos que se encarga de los cambios en los requisitos y la aparición de nuevos requisitos durante y después del proceso de desarrollo de software.

La IR provee métodos, técnicas y herramientas que ayudan a los ingenieros de requisitos a elicitar y especificar requisitos, asegurando el máximo de calidad y completitud. Sin embargo, el problema de la completitud es una amenaza constante a la calidad de los requisitos y cubre todo el proceso de IR con un gran manto de duda. La completitud es una meta inalcanzable y estimar el grado de completitud logrado en cualquier momento del proyecto es muy difícil. El ingeniero de requisitos se enfrenta

con un Universo de Discurso (UdeD) que nunca parece terminar de conocer. Esta situación no es única en el proceso global de desarrollo de software ya que por ejemplo durante las pruebas o durante las inspecciones de software ocurre algo muy similar.

El problema de la completitud en la Ingeniería de Software y en la Ingeniería de Requisitos en particular es similar a muchos otros problemas en otras áreas del conocimiento. Otis [7] introdujo un método para estimar el tamaño de una población cerrada de animales basándose en los datos de captura y recaptura de especímenes. Este método ha sido utilizado exitosamente en el área de inspecciones de software por varios autores [8] [9] [10].

La validación de los requisitos se ha convertido en una tarea compleja, principalmente debido al tipo de modelos de representación que requieren en muchos casos clientes-usuarios con habilidades especiales para entenderlos. Como señalan [6] y [11], la validación de requisitos rara vez descubre todos los problemas, de modo que éstos llegan a etapas posteriores del desarrollo. El uso de representaciones basadas en lenguaje natural ayuda a la validación de los requisitos, que mejora notablemente cuando las representaciones son expresadas usando el vocabulario de los clientes-usuarios. Para ello, los ingenieros de requisitos tienen que adquirir conocimiento del vocabulario de la aplicación y aplicar ese conocimiento. De ese modo, se establece una adecuada comunicación entre los involucrados. La dificultad de las representaciones basadas en lenguaje natural es la ambigüedad [12] [6] [13], un inconveniente que podría reducirse y aún evitarse con la elaboración de un glosario de la aplicación. La construcción de un glosario no sólo sirve a los propósitos de la comunicación, sino que también brinda información al desarrollador y lo hace participar de una realidad diferente: el contexto de la aplicación. A lo largo del proceso de desarrollo de software, este glosario sirve como un medio de entendimiento mutuo entre los involucrados.

En una buena parte de las metodologías de la IR se enfatiza la importancia de contar con un glosario, para satisfacer diferentes necesidades, tales como establecer una terminología común entre los desarrolladores, entender el vocabulario del cliente-usuario, proveer una fuente de información de fácil acceso, etc. [14] [15] [16] [17] [18]

En particular, una de las posibles estrategias para conocer el UdeD se basa en la construcción de un Léxico Extendido del Lenguaje (LEL) y luego los Escenarios que modelan situaciones en dicho universo [19]. El proceso de construcción del Léxico Extendido del Lenguaje y luego de los Escenarios es sólo el comienzo de una secuencia de actividades cuyo propósito consiste en elicitar primero conocimiento del UdeD, y luego el conjunto de los requisitos del sistema de software a ser desarrollado.

En este trabajo se presenta un método cuantitativo para estimar la cantidad máxima de términos que podría contener el LEL de un UdeD determinado utilizando una estrategia de captura/recaptura ya aplicada en la estimación del número total de defectos en inspecciones. [9] [8] [20]

El presente artículo está organizado de la siguiente manera: La sección 2 presenta un muy breve resumen del proceso en el que se realizó el experimento, la sección 3 describe las técnicas de predicción basadas en la captura y recaptura, especialmente el método basado en el perfil de detecciones introducido por Wohlin [20], la sección 4 detalla el caso de estudio utilizado, la sección 5 analiza los resultados obtenidos y

finalmente se elaboran algunas conclusiones y se detallan las cuestiones abiertas que se planifica investigar en un futuro cercano.

2 Léxico en el Proceso de Construcción de Escenarios

La idea general de la estrategia presentada en [19] es “anclar” la descripción de Escenarios en el vocabulario del UdeD. El conocimiento adquirido por medio de observaciones, lectura de documentos, entrevistas, y otras técnicas, es modelado usando el LEL, en primer término, y los escenarios, más tarde. Una primera versión de los escenarios es derivada a partir de la información del LEL. Estos escenarios son completados y mejorados utilizando otras fuentes de información y organizados con el fin de obtener un conjunto de escenarios que representen el dominio de la aplicación. Durante o después de estas actividades, los escenarios son verificados y validados con los clientes/usuarios.

2.1 Léxico Extendido del Lenguaje

El propósito del léxico [21] es registrar símbolos (palabras o frases) peculiares del dominio de la aplicación y su semántica, dejando para un siguiente paso la comprensión del problema. El LEL está compuesto por un conjunto de símbolos que conforman el lenguaje de la aplicación y que representan, en general, palabras o frases que el cliente/usuario repite o enfatiza o son relevantes para el dominio, más allá de su frecuencia de repetición.

La representación de los símbolos se realiza identificando cada símbolo con uno o más nombres y registrando para cada símbolo la *noción* (oraciones que lo definen) y el *impacto* (oraciones que indican cómo repercute en la aplicación).

En la figura 1 se presenta el modelo utilizado para representar los símbolos del LEL.

La construcción del léxico parte de la identificación de las fuentes de información, luego se identifican los símbolos del dominio, se los clasifica, se los describe y, finalmente, el léxico sufre procesos de verificación y validación que retroalimentan el proceso de construcción en sí mismo.

En la descripción de los símbolos deben cumplirse simultáneamente dos reglas básicas:

Principio de circularidad: en la descripción de la noción o impacto de los símbolos se debe maximizar el uso de otros símbolos del léxico. De esta manera, el conjunto de símbolos determina una red, que permite representar al LEL mediante un hipertexto que puede ser navegado para conocer todo el vocabulario del problema.

Principio del vocabulario mínimo: se debe minimizar el uso de símbolos externos al lenguaje de la aplicación. De este modo, se acota el lenguaje al menor conjunto de símbolos posible. Si se

utilizan símbolos externos, éstos deben pertenecer al vocabulario básico del lenguaje natural que se está utilizando [22].

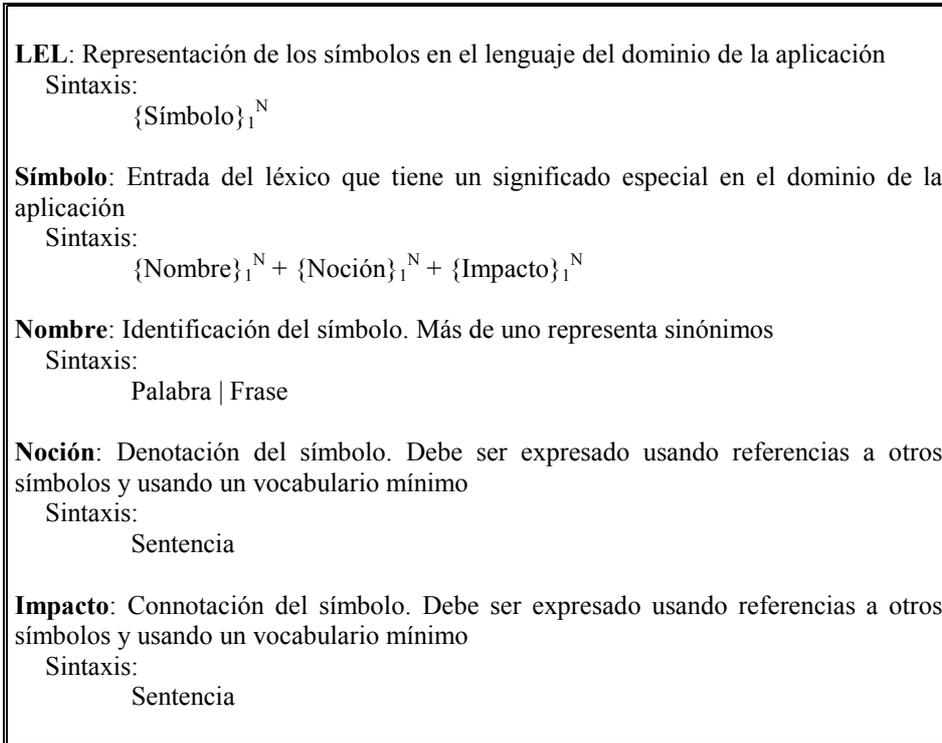


Fig. 1. Modelo del Léxico Extendido del Lenguaje

donde:

Sentencia está compuesta por Símbolos y No-Símbolos pertenecientes al vocabulario mínimo,
+ significa composición,
{x} significa cero o más ocurrencias de x,
() es usado para agrupamiento,
| significa or, y
[x] significa que x es opcional

3 Estimaciones de Tamaño en Universos Cerrados

Los métodos de estimación de poblaciones cerradas de cualquier naturaleza, animales, errores en el software y en este caso términos de LEL se basan en la idea de la captura y recaptura. Inicialmente, cuando se aplicó a poblaciones de animales, se

capturó una cierta cantidad de animales en una primera cacería [7]. Obviamente se utilizaron cacerías incruentas con el objeto de minimizar la perturbación a la población animal. A cada animal capturado se lo marcó y liberó sin ningún otro tipo de acción sobre el mismo. Transcurrido un cierto período de tiempo se realizó otra cacería, registrando la cantidad de animales que ya habían sido capturados anteriormente. En los casos en que se planeaba efectuar una o varias cacerías más, se marcó de manera diferente los animales capturados una sola vez de aquellos capturados dos veces. La hipótesis básica de esta estrategia consiste en que si el nivel de recaptura es alto, es decir si la mayoría de los animales es capturada en ambas oportunidades, entonces la población es reducida, mientras que si el nivel de recaptura es bajo, entonces la población es amplia. En estos usos iniciales se hipotetizó adicionalmente que todos los animales tenían la misma probabilidad de ser capturados y que todos los capturadores eran igualmente eficaces. Sin embargo, aun en las aplicaciones iniciales, ambas hipótesis fueron puestas bajo la lupa, dudándose de su veracidad.

Al aplicar este tipo de estrategias a las inspecciones de software [8] [9] [10], el rol de los animales salvajes fue tomado por los defectos del objeto inspeccionado y el rol de los capturadores le correspondió a los inspectores. En este caso, las dudas acerca de la veracidad de la afirmación que todos los defectos tienen una dificultad similar de ser descubiertos y que los inspectores son igualmente idóneos es también cuestionada, aunque con más énfasis aún. Actualmente, se utilizan 4 modelos basados en las hipótesis de detección:

- M0: Todos los defectos tienen la misma probabilidad de ser detectados y los inspectores tienen la misma probabilidad de detectar defectos.
- Mt: Todos los defectos tienen la misma probabilidad de ser detectados pero la probabilidad con que los inspectores detectan errores es diferente.
- Mh: Los defectos tienen diferente probabilidad de ser detectados pero los inspectores tienen la misma probabilidad de detectar defectos.
- Mth: Tanto los defectos como los inspectores tienen distintas probabilidades.

Se ha comprobado que los modelos Mh y Mth son los más adecuados para el caso de defectos detectados en inspecciones [23].

Wohlin y Runeson [20] propusieron posteriormente métodos alternativos que no se basan en las hipótesis de los modelos M0, Mh, Mt y Mth, sino que simplemente se basan en ajustar la curva que representa los datos de la cantidad de inspectores que encontró cada defecto. El fundamento de este método reside en que un ajuste de los datos experimentales con una curva teórica simple como puede ser una exponencial decreciente empotra dentro de los parámetros de la curva las variaciones de probabilidades del modelo Mth sin necesidad de suponer ninguna distribución de probabilidades en particular.

Considerando:

$$m_x = A \times e^{-bx}, \quad (1)$$

donde m_x indica el número total de inspectores que encuentran un defecto x , b describe el decrecimiento de la función exponencial, y A es una constante. Mediante técnicas de regresión es posible estimar los parámetros A y b . Una vez estimados dichos parámetros, el número total de defectos es determinado por el mayor valor de x para el cual la ecuación (1) produce un resultado mayor o igual que 0.5.

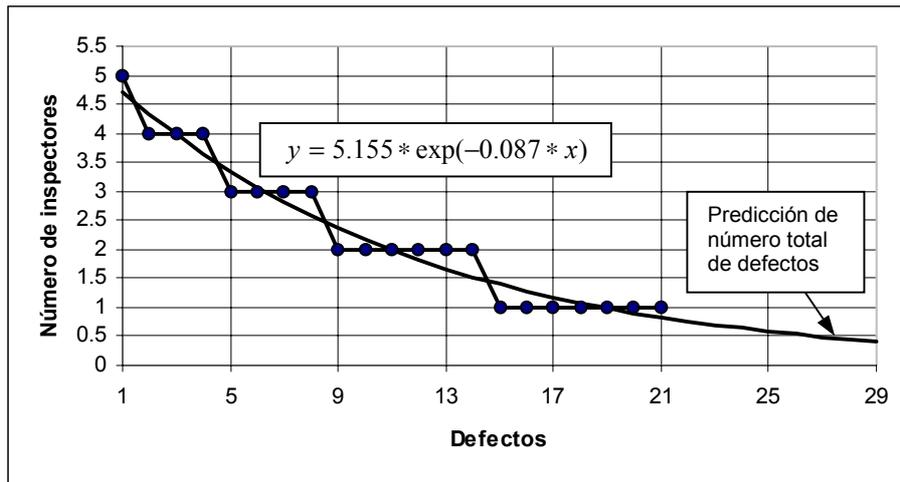


Fig. 2. Ejemplo de aplicación del método Detection Profile Method (Extraída de [20])

En la figura 2 se presenta un ejemplo de la aplicación del método Detection Profile Method (DPM) para estimar el número total de defectos en un artefacto de software. En este ejemplo, cinco inspectores han detectado 21 defectos diferentes. Para cada defecto, se ha calculado cuántos inspectores lo detectaron. En el gráfico, los defectos han sido ordenados en forma decreciente de acuerdo al número de inspectores que detectó cada uno de ellos.

Ajustando los datos representados con una curva exponencial decreciente, es posible predecir el número total de defectos [20]. En este caso, los parámetros A y b de la expresión (1) son, respectivamente, 5.155 y -0.087. Calculando el mayor valor de x , para el cual se obtiene un valor mayor o igual que 0.5, se obtiene una estimación de 27 defectos.

La estimación del número total de defectos mediante la propuesta de Wohlin y Runeson tiene dos grandes atractivos:

1. La calidad de su estimación es igual o superior a la lograda con el método Mth
2. La aplicación es mucho más sencilla.

4 Caso de Estudio

El caso de estudio que se utilizó en este trabajo corresponde a un Sistema de Planes de Ahorro Previo para la Adquisición de Vehículos 0Km [24] [25]. Este sistema se dedica a la gestión y la administración de planes de ahorro para adquirir automotores 0Km. Funciona a través de grupos de una cierta cantidad de personas físicas o legales, los cuales participan mensualmente de los sorteos que la Administradora organiza, con el objeto de entregar una unidad por grupo. Los integrantes o adherentes de cada grupo pueden tratar de anticipar la entrega de la unidad que les corresponde, presentando ofertas de licitación. En cada acto de adjudicación, además de sortear el número del adherente favorecido de cada grupo, se abren las ofertas de licitación, si las hubiera, para determinar la mejor oferta y establecer así el beneficiario. Además, la Administradora debe arbitrar en caso de renuncia o fallecimiento de participantes, o falta de pago de las cuotas mensuales, entiende en cuestiones legales tales como trámites sucesorios, seguros de vida y del automotor, intimaciones de pago, contratos con los fabricantes, etc.

Para este caso de estudio, desarrollado en la Universidad Nacional del Centro de la Provincia de Buenos Aires, se utilizó como primera técnica de recolección de datos la lectura de documentos.

Nueve grupos, utilizando la misma técnica de elicitación, procedieron a la confección del Léxico Extendido del Lenguaje.

La construcción de los glosarios fue realizada sin ninguna indicación o regla de detención [26], dejándose librada la finalización de la búsqueda de nuevos términos, al buen criterio de los miembros de cada grupo.

Por otra parte, el caso fue realizado con otros fines, y sólo tiempo después se decidió revisarlo con el propósito de estudiar la completitud. Esta secuencia de acontecimientos, garantiza absolutamente la ausencia de factores significativos que pudieran entorpecer la validez de los datos analizados.

Los resultados obtenidos por los diferentes grupos se presentan en la tabla 1:

Tabla 1. Cantidad de símbolos elicitados por cada grupo

Grupo	Número de símbolos
1	52
2	29
3	30
4	33
5	65
6	13
7	44
8	24
9	54

El número total de símbolos diferentes detectados por los 9 grupos fue de 118. En forma intuitiva se considera que este número es probablemente cercano al número máximo de símbolos relevantes en el UdeD.

5 Análisis de los Datos

El problema de completitud en el LEL tiene dos dimensiones: la completitud interna y la completitud externa. Por completitud interna o intra símbolo se entiende haber descrito completamente cada símbolo, es decir haber colocado todos los posibles significados y todos los posibles impactos del mismo. Por completitud externa se entiende haber identificado todos los posibles términos del LEL. En la presente sección y en todo el artículo se estudia la completitud del LEL desde el punto de vista externo.

En el desarrollo del caso de estudio la habilidad para detectar símbolos del LEL de los diferentes grupos era extremadamente diferente, por lo que es esperable que este dato introduzca una degradación en la calidad de la estimación del número total de símbolos que se realice. Por este motivo, se realizaron dos estimaciones diferentes. En primer lugar, se procedió a estimar el número total de símbolos en base a los resultados obtenidos por los tres grupos que detectaron mayor cantidad de símbolos, demostrando su mayor habilidad para realizar la tarea. En la figura 3 se presenta la predicción realizada a partir de los datos obtenidos por los grupos 1, 5 y 9 que fueron los que detectaron mayor cantidad de símbolos. El número total de símbolos detectados por estos tres grupos fue 96, y el total estimado fue de 122, como puede observarse en la figura.

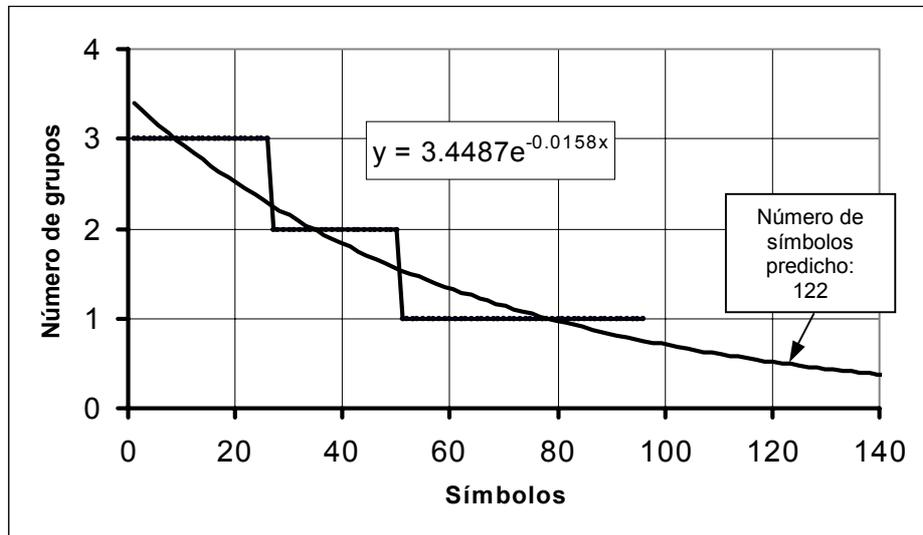


Fig. 3. Predicción a partir de los datos obtenidos por los grupos 1, 5 y 9

Luego, se hizo la estimación para todos los grupos participantes, obteniendo el gráfico que se presenta en la figura 4. El número total de símbolos diferentes detectados por los nueve grupos fue de 118, y la estimación correspondiente fue de 127.

La observación de las figuras permite afirmar que en el caso de todos los grupos la calidad del ajuste de la ecuación de regresión exponencial no es la mejor posible, por lo que sería esperable que otra fórmula de regresión proveyera una mejor estimación del número de símbolos totales del léxico que está siendo capturado. Sin embargo, la coherencia de ambas predicciones permite suponer que la mejora no sería importante.

Es posible que la distorsión presente en la figura 4 se deba a la gran diferencia en la calidad del trabajo entre los grupos, la cual no sería adecuadamente modelizada por una exponencial decreciente.

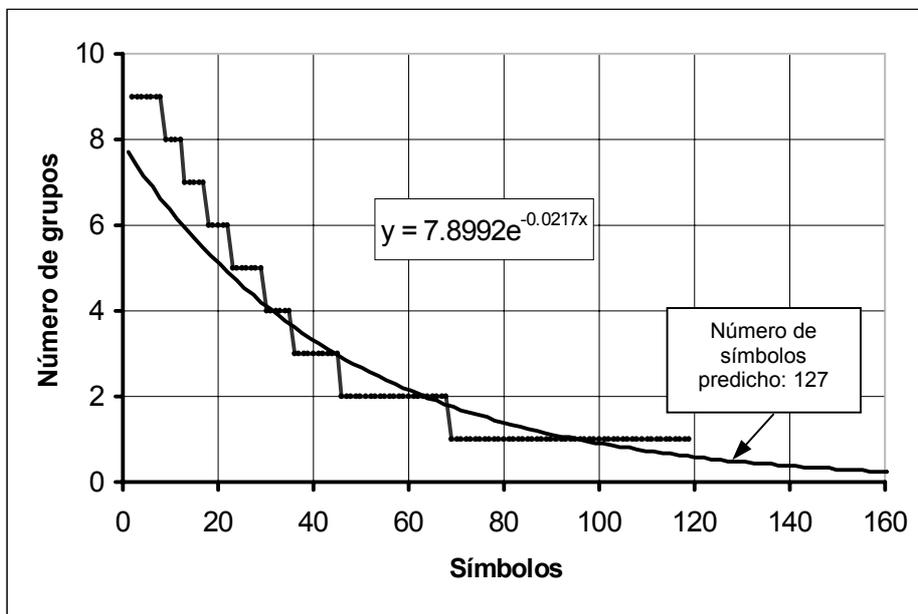


Fig. 4. Predicción a partir de los datos obtenidos por todos los grupos

En la Tabla 2 se comparan las predicciones de las figuras 3 y 4 con la cantidad de símbolos conocidos.

Tabla 2. Comparación entre las predicciones y la cantidad de símbolos conocidos

Cantidad total de símbolos conocidos.	118
Cantidad estimada en base a los grupos 1, 5 y 9.	122
Cantidad estimada en base a todos los grupos.	127

El análisis de estos datos confirma casi totalmente lo predicho por [20] en el caso de inspecciones, donde se supone que al aumentar el número de inspectores, la

cantidad de defectos estimada será igual a la cantidad real existente. En este sentido, se considera más apropiado enunciar que cuando el número de observaciones independiente crece, es razonable suponer que la diferencia entre la cantidad estimada y la cantidad efectivamente obtenida se reduce. En este caso, con tres captosres, a partir de 96 términos reales, se predice la existencia de 122. Es decir, se estima que quedan por descubrir 26. Con 9 grupos, en cambio, la cantidad estimada es de 127 y la cantidad real es de 118, reduciéndose la estimación de faltantes a 9.

6. CONCLUSIONES

La correspondencia entre el modelo teórico y los datos experimentales resultan un buen augurio en esta aplicación inicial del método del perfil de detección a la estimación de los símbolos totales que debiera tener el LEL del caso estudiado.

Es evidente que deben realizarse muchos más estudios para mejorar la estimación del número total de símbolos del LEL. Sin embargo, surgen inmediatamente algunas otras cuestiones que en principio parecen ser mucho más relevantes. Por ejemplo, resulta necesario conocer si el metaobjetivo de un LEL lo más completo posible es de alguna manera necesario. Claramente, la Ingeniería de Requisitos tiene implícito el imperativo de realizar su tarea de la manera más completa posible. Pero la completitud aspirada se refiere a los requisitos, no a los modelos intermedios que se construyen para arribar a los mismos. Antes de aspirar a disponer de un LEL extremadamente completo es necesario conocer cómo impacta el grado de completitud del LEL en la calidad de los requisitos obtenidos. Desde un punto de vista más instrumental importa conocer cómo impacta la completitud del LEL sobre la completitud de los escenarios. Este mero enunciado permite ver claramente que no sólo se necesita perfeccionar el uso de las técnicas de estimación para el LEL sino que también es necesario extender su aplicación a los otros modelos utilizados en el proceso, como recurso ineludible para comenzar a comprender la compleja relación entre la completitud de una etapa del proceso de requisitos y la siguiente. Fundamentalmente importa, entonces, dónde deben aplicarse los esfuerzos para lograr procesos no sólo eficaces sino también eficientes y compatibles en costo y en tiempo con las restantes etapas del proceso global de desarrollo de software.

En relación con el léxico, en el futuro se planifica estudiar la completitud externa para otros casos de estudio y, posteriormente, abordar el problema de la completitud interna.

Referencias

1. Maculay, L.: Requirements Capture as a Cooperative Activity. En: Proceedings IEEE International Symposium on Requirement Engineering. IEEE Computer Society Press, San Diego, Ca (1993) pp. 174-181.
2. Reubenstein, H. B., Waters, R.C.: The requirements apprentice: Automated assistance for requirements acquisition. IEEE Transaction on Software Engineering, Vol. 17, N° 3 (1991) pp 226-240.

3. Jacobson, Y., Christerson, M., Jonsson, P., Overgaard, G.: *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison Wesley, Reading, MA, ACM Press, New York (1992).
4. Bubenko, J.A., Wrangler, B.: Objective driven capture of business rules and information systems requirements. En: *Proceedings de IEEE Conference on Systems, Man and Cybernetics* (1993).
5. Kotonya, G., Sommerville, I.: *Requirements Engineering. Processes and Techniques*. John Wiley & Sons (1998).
6. Sommerville, I., Sawyer, P.: *Requirements Engineering. A good Practice Guide*. John Wiley & Sons (1997).
7. Otis, D.L., Burnham, K.P. White G.C., Anderson D.R.: *Statistical inference from Capture on Closed Animal Populations*. *Wildlife Monograph*, 62 (1978).
8. Briand, L., El Emam, K., Freimut, B., Laitenberger, O.: A Comprehensive Evaluation of Capture-Recapture Models for Estimating software Defects Contents. *IEEE Transactions on Software Engineering*, Vol. 26, N° 6 (2000) pp 518-540.
9. Biffl, S.: Using Inspection data for Defect Estimation. *IEEE Software special Issue on recent project estimation methods* (2000).
10. Biffl, S.: Evaluating defect estimation models with major defects. *The Journal of Systems and Software*, 65 (2003) pp. 13-29.
11. Cysneiros, L.M., Yu, E.: Non-Functional Requirements Elicitation. En: Leite, J.S.C.P., Doorn, J.H (eds.): *Perspectives on Software Requirements*. Kluwer Academic Press (2003) pp. 115-138. (en prensa)
12. Jackson, M.: *Software Requirements & Specifications. A lexicon of practice, principles and prejudices*. Addison Wesley, ACM Press (1995).
13. Berry, D.M., Kamsties, E.: Ambiguity in Requirements Specification. En: Leite, J.S.C.P., Doorn, J.H (eds.): *Perspectives on Software Requirements*. Kluwer Academic Press (2003) pp. 7-44. (en prensa)
14. Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C.: Guiding Use Case Authoring: Results of an Empirical Study. En: *Proceedings International Symposium On Requirements Engineering (RE'99)*. IEEE Computer Society Press, Limerick-Ireland (1999) pp. 36-43.
15. Rolland, C., Ben Achour, C.: Guiding the construction of textual use case specifications. *Data & Knowledge Engineering* 25 (1998) pp. 125-160.
16. Oberg, R., Probasco, L., Ericsson, M.: *Applying Requirements Management with Use Cases*. Rational Software Corporation (1998).
17. Regnell, B., Kimbler, K., Wesslén, A.: Improving the Use Case Driven Approach to Requirements Engineering. *Requirements Engineering with Use Cases – a Basis for Software Development*. Technical Report 132, Paper I (1995), Department of Communication Systems, Lund University (1999) pp. 43-63.
18. Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P.: *Scenarios in System Development: Current Practice*. *IEEE Software* (1998) pp. 34-45.
19. Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N.: A Scenario Construction Process. *Requirements Engineering Journal*, Vol.5, N° 1 (2000) pp. 38-61.
20. Wohlin, C., Runeson, P.: Defect content estimations from Review Data. En: *Proceedings of the 20th International Conference on Software Engineering* (1998).
21. Leite, J.C.S.P., Franco, A.P.M.: O Uso de Hipertexto na Elicitação de Linguagens da Aplicação, *Anais de IV Simpósio Brasileiro de Engenharia de Software*. SBC, Brazil (1990) pp. 134-149.
22. Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A.: Enhancing a Requirements Baseline with Scenarios. *Requirements Engineering Journal*, Vol.2, N° 4 (1997) pp. 184-198.

23. Briand, L., El Emam, K., Freimut, B., Laitenberger, O.: Quantitative evaluation of Capture Recapture Models to Control Software Inspections. En: Proceedings of the 8th International Symposium on Software Reliability Engineering (1997) pp 234-244.
24. Mauco, V., Ridaio M., del Fresno, M., Rivero, L., Doorn, J.H.: Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro. Reporte técnico, ISISTAN, UNCPBA, Tandil, Argentina (1997).
25. Rivero, L., Doorn, J., del Fresno, M., Mauco, V., Ridaio, M., Leonardi, M.C.: Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real. En: Proceedings of WER'98, Workshop en Engenharia do Requisitos, Maringá, Brasil (1998) pp. 79-90.
26. Pitts, M.: The Use of Stopping Rules in Information Requirements Determination. Americas Conference on Information Systems. Indianapolis, Indiana (1997).