UNA EVALUACIÓN A LOS MÉTODOS PARA ELICITAR REQUISITOS DE SEGURIDAD

Marina Torrente A., Estella Periutella W.
Universidad de Río Negro, Argentina
matorrente@urn.edu.ar, eperiutella@urn.edu.ar

(Tipo de artículo: **INVESTIGACIÓN**. Recibido el 07/08/2011. Aprobado el 28/11/2011)

RESUMEN
Utilizar un método de elicitación puede ayudar para la especificación de un conjunto coherente y completo de requisitos de seguridad. Sin embargo, usualmente, los métodos comunes utilizados para elicitar requisitos funcionales no se orientan a requisitos de seguridad, por lo cual, el conjunto resultante de requisitos no los incluye. En este artículo se analizan algunos métodos de elicitación de requisitos de seguridad y se presenta una propuesta para seleccionar el más adecuado; posteriormente, se seleccionan algunos métodos y se aplican a varios estudios de caso.

Palabras clave
Elicitación de requisitos, requisitos de seguridad, método.

AN EVALUATION TO THE METHODS FOR ELICITING SAFETY REQUIREMENTS

ABSTRACT
Using an elicitation method can help specifying a coherent and comprehensive set of security requirements. However, usually, the common methods used to elicit functional requirements are not oriented to security requirements, therefore, the resulting set of requirements does not includes them. In this article are analyzed some methods of security requirements elicitation and is presents a proposal to select the most suitable, subsequently, some methods are selected and applied to several case studies.

Keywords
Requirements elicitation, security requirements, method.

UNE EVALUATION AUX MÉTHODES POUR ÉLUCIDER DES EXIGENCES DE SÉCURITÉ

RÉSUMÉ
Utiliser une méthode d'élicitation peut aider pour la spécification d'un ensemble cohérent et complet des exigences de sécurité. Cependant, les méthodes communes utilisées pour éliciter des exigences fonctionnelles usuellement ne sont pas prévus pour exigences de sécurité, ainsi, l'ensemble résultant des exigences ne les inclut pas. Dans cet article on analyse quelques méthodes d'élicitation des exigences de sécurité et on présente une proposition pour sélectionner le plus approprié ; après, nous avons choisi quelques méthodes pour les appliquer sur cas d'étude divers.

Mots-clés
Élicitation des exigences, exigences de sécurité, méthode.

1. INTRODUCCIÓN

Utilizar un método de elicitación puede ayudar en la especificación de un conjunto coherente y completo de requisitos de seguridad. Sin embargo, los métodos comunes de lluvia de ideas y de elicitación utilizados para elicitar los requisitos funcionales, usualmente no están orientados para requisitos de seguridad, por lo cual, el conjunto resultante no los tiene incluidos. Cuando los requisitos de seguridad se elicitan de forma sistemática es probable que el producto final tenga menos riesgos de seguridad.

En este artículo se analizan algunos métodos de elicitación y se presenta una propuesta para seleccionar el más adecuado y luego, los seleccionados, se aplican a estudios de caso. Los resultados que muchos estudios presentan pueden variar de una organización a otra, por lo que la propuesta que en este trabajo debe considerarse como de propósito general. La elicitación de requisitos es un área de investigación activa y se esperan importantes avances en el futuro cercano, igualmente estudios de medición acerca de cuáles son los métodos más eficaces para elicitar requisitos de seguridad. Actualmente, sin embargo, existen pocos trabajos en los que se compara la eficacia de los diferentes métodos para elicitar requisitos de seguridad.

El resto del documento se estructura de la siguiente manera: la sección 2 describe algunos métodos de elicitación de requisitos, en la 3 se detallan los criterios de evaluación aplicados en esta investigación, en la sección 4 se muestran los resultados obtenidos y las discusiones respectivas y en la 5 se encuentran las conclusiones de este proceso investigativo.

2. DESCRIPCIÓN DE ALGUNOS MÉTODOS DE ELICITACIÓN DE REQUISITOS

La siguiente lista es un ejemplo de los métodos que podrían considerarse para elicitar requisitos de seguridad. Algunos se han desarrollado específicamente pensando en la seguridad –por ejemplo, *misuse cases*–, mientras que otros se han utilizado para la ingeniería de requisitos tradicional y potencialmente podrían utilizarse para requisitos de seguridad. En el futuro se podrá tener una mejor comprensión de cómo los aspectos únicos de la elicitación de requisitos de seguridad direccionan la selección de un método. Se ha presentado algunos trabajos [1-3] en elicitación de requisitos en general que podrían considerarse para elaborar una lista como la que aquí se propone, lo mismo que para llevar a cabo el proceso de selección [2].

2.1 Misuse cases [4, 5]

Un caso de uso generalmente describe el comportamiento del sistema que el cliente desea [5]. Los modelos de caso de uso y sus diagramas asociados han demostrado ser muy útiles para la especificación de requisitos [6, 7]. Sin embargo, una colección de casos de uso no debe utilizarse como sustituto de un documento de especificación de requisitos, ya que este enfoque puede generar sólo simples vistas de requisitos importantes [8]. Como resultado, es controversial utilizar modelos de casos de uso para elicitar requisitos del sistema y de calidad.

Misuse cases aplica el concepto de escenario negativo en el contexto de casos de uso, es decir, una situación que el propietario del sistema no quiere que se produzca. Por ejemplo, los líderes empresariales, los planificadores militares, y los jugadores analizan los mejores movimientos de sus oponentes como una amenaza identificable. El método *Misuse cases* también es conocido como “casos de abuso”. Una discusión más profunda de los casos de abuso como un enfoque para identificar los requisitos de seguridad se puede encontrar en McGraw [5].

Una característica importante de *Misuse cases* es que parece producir los requisitos de calidad como los de seguridad y protección, mientras que otros métodos de elicitación se centran en los requisitos del usuario final, por lo que se desconoce su eficiencia para identificar requisitos de seguridad. Los casos de uso describen el comportamiento del sistema en términos de requisitos funcionales. La interacción entre *Misuse cases* y casos de uso podría mejorar la eficiencia de la elicitación de todos los requisitos en el ciclo de vida de la ingeniería de software. *Misuse cases* y casos de uso se pueden desarrollar desde el sistema a los niveles del subsistema –y menos si es necesario. Los casos de nivel más bajo pueden llamar la atención acerca de problemas de fondo que no se consideran en los niveles superiores y pueden obligar a los ingenieros a volver a analizar el diseño del sistema. *Misuse cases* no es un método arriba-abajo, sino que proporciona oportunidades para investigar y validar los requisitos de seguridad necesarios para llevar a cabo la misión del sistema.

2.2 Metodología Soft Systems [9]

SSM trata con situaciones problemáticas en las que existe un alto componente social, político, y de actividades humanas, y puede enfrentar "problemas blandos" que son difíciles de definir, en lugar de "problemas duros" que están más orientadas a la tecnología. Ejemplos de problemas blandos son: cómo solucionar la falta de vivienda, cómo gestionar la planificación de desastres, y cómo mejorar el sistema de salud. Eventualmente, los problemas orientados a la tecnología pueden surgir de estos problemas suaves, pero se necesita mucho más análisis para llegar a ese punto. SSM está compuesta por siete fases:

1. Encontrar la situación problema.
2. Expresar la situación problema a través de buenas imágenes; es decir, representaciones de la estructura organizacional y los procesos pertinentes a la situación.
3. Seleccionar cómo ver la situación y producir definiciones básicas.
4. Construir modelos conceptuales de lo que debe hacer el sistema para cada definición básica.
5. Comparar los modelos con el mundo real.

6. Identificar los cambios posibles y deseables.
7. Hacer recomendaciones para mejorar la situación problema.

2.3 Quality Function Deployment [10]

QFD es un concepto global que proporciona un medio para traducir los requisitos del cliente en requisitos técnicos apropiados para cada fase del desarrollo y producción del producto. El atributo distintivo de QFD es el énfasis en las necesidades del cliente en todas las actividades de desarrollo del producto. Mediante el uso de QFD, las organizaciones pueden promover el trabajo en equipo, priorizar los detalles de acción, definir objetivos claros, y reducir el tiempo de desarrollo. A pesar de QFD cubre una amplia porción del ciclo de vida del desarrollo del producto, las primeras fases del proceso son aplicables para capturar requisitos para la Ingeniería de Software. Estas fases incluyen:

1. Identificar al cliente.
2. Capturar los requisitos de alto nivel.
3. Construir un conjunto de características del sistema que puedan satisfacer las necesidades del cliente.
4. Crear una matriz para evaluar las características del sistema contra la satisfacción de las necesidades del cliente.

Tenga en cuenta que la evaluación de las características y necesidades podría además ser utilizada para la priorización de los requisitos, en el contexto de una actividad QFD de elicitación de requisitos.

2.4 Controlled Requirements Expression [11, 12]

CORE es un método de análisis y especificación de requisitos que clarifica los puntos de vista del usuario acerca de los servicios que debe suministrar el sistema propuesto y las limitaciones impuestas por el entorno operativo de ese sistema, junto con cierto grado de investigación de rendimiento y fiabilidad [13]. Además, proporciona métodos y notaciones para cada fase de la elicitación, especificación y análisis de requisitos, y los resultados en forma de un flujo de datos estructurados de la especificación [14]; es un método de madurez con un conjunto de directrices acerca de cómo aplicar el método a un problema [11]. El método es un enfoque flexible para elicitar requisitos, que facilita la posibilidad de aplicarlo a un amplio conjunto de problemas; estimula las contribuciones provenientes desde muchas comunidades para desarrollar requisitos; determina las tareas de los miembros de esta comunidad y estructura la comunicación entre estos grupos [12]. Con CORE, se puede implementar una revisión incremental de flujos de información y de actividades de procesamiento, debido a que con cada fase previa proporciona las bases para la actual fase de la especificación. CORE ayuda a descubrir las limitaciones de diseño.

2.5 Issue-Based Information Systems [15]

El método de IBIS se basa en el principio de que el proceso de diseño para problemas complejos esencialmente es un intercambio entre las partes interesadas, debido a que aportan su experiencia y perspectiva personal para la solución de los temas de diseño. Cualquier problema, inquietud o pregunta puede ser una cuestión que puede requerir discusión y resolución antes de proceder al diseño. El modelo IBIS se centra en este dar y recibir que constituye el proceso de diseño. El modelo se ha implementado eficientemente en situaciones de diseño variado, desde el diseño arquitectónico hasta la planificación en la Organización Mundial de la Salud.

El modelo IBIS se centra en la articulación de las cuestiones clave en el problema de diseño. Cada problema puede tener muchas posiciones. Una posición es una declaración o afirmación que resuelve el problema. A menudo, las posiciones pueden ser mutuamente excluyentes, pero el método no requiere esto. Cada una de las posiciones del problema, a su vez, puede tener uno o más argumentos que apoyan o se oponen a ella. Hay varios tipos de enlaces entre los conceptos in IBIS. Por ejemplo, una posición responde a un problema con un enlace "responde a". Los argumentos deben estar relacionados con sus posiciones, con sus “soportes” o con los enlaces "se opone a". Los problemas pueden generalizarse o, más estrechamente, focalizarse en otros problemas y pueden preguntar o ser propuestos por otros problemas, posiciones y argumentos.

2.6 Joint Application Development [16]

JAD se ha diseñado específicamente para el desarrollo de grandes sistemas informáticos. El objetivo del JAD es involucrar a todas las partes interesadas en la fase de diseño del producto a través de reuniones altamente estructuradas y focalizadas. Los participantes típicos en una sesión incluyen a un facilitador, a usuarios finales del producto, a los desarrolladores principales, y a los observadores. En las fases preliminares del JAD, el equipo de ingeniería de requisitos se encarga de investigar los hechos y de recopilar la información. Por lo general, los resultados de esta fase, tal como se aplica a la elicitación de requisitos de seguridad, son los objetivos y artefactos de seguridad. La sesión del JAD se utiliza para validar esta información mediante el establecimiento de un conjunto acordado de requisitos de seguridad para el producto.

2.7 Feature-Oriented Domain Analysis [17]

FODA es un método de ingeniería y de análisis de dominio que se centra en desarrollar activos. Al examinar los sistemas software relacionados y la teoría subyacente de la clase de sistemas que representan, el análisis de dominio puede proporcionar una descripción genérica de los requisitos de esa clase de sistemas en la forma de un modelo de dominio y un conjunto de criterios para su implementación. El método FODA se basa en dos conceptos de modelado: la abstracción y el refinamiento [18]. La abstracción se utiliza para crear modelos de dominio, como se describió anteriormente, desde las aplicaciones específicas en el dominio. Estos

productos de dominio genéricos abstraen la funcionalidad y el diseño de las aplicaciones en un dominio. La naturaleza genérica de los productos de dominio se crea mediante la abstracción de los factores que hacen una aplicación diferente de otras aplicaciones relacionadas. El método FODA aboga por que las aplicaciones en el dominio deben ser extraídas para el nivel en que no existen diferencias entre las aplicaciones. Las aplicaciones específicas en el dominio se desarrollan como refinamientos de él.

2.8 Critical Discourse Analysis [19]

CDA usa métodos sociolingüísticos para analizar el discurso verbal y escrito. La sociolingüística asigna especial importancia a la estructura del discurso y los textos, y proporciona métodos para especificar, en grandes unidades de significado, las características lingüísticas de los diferentes tipos de unidades del discurso y la forma en que están unidos [20]. Por otra parte, CDA se ocupa de examinar el contexto social a lo largo de las líneas de la ideología, el poder y la desigualdad. A través del examen del discurso se exponen los temas de las desigualdades de poder usualmente a lo largo de las líneas de raza, clase, género, sexualidad y la ocupación. En particular, CDA se puede utilizar para analizar las entrevistas de elicitación de requisitos y para comprender las narrativas e "historias" que surgen durante ellas.

2.9 Accelerated Requirements Method [21]

El proceso ARM es una actividad que facilita la descripción y elicitación de requisitos. El proceso tiene tres fases: 1) Fase de preparación, 2) Fase de sesión y 3) Fase de cierre.

Durante la fase de preparación, se completa la planificación y la preparación para garantizar una sesión eficaz. Durante esta actividad, se definen los objetivos y metas generales y el alcance preliminar del esfuerzo; se definen las medidas clave para éxito; se identifican los principales participantes y se desarrolla el calendario preliminar. Generalmente, esta fase tiene una duración de uno a cuatro días.

Durante la fase de sesión, un facilitador capacitado, y neutral, conduce a los participantes seleccionados a través de un proceso estructurado para colectar los requisitos funcionales del proyecto. El proceso facilitado emplea técnicas de alcance definido, lluvia de ideas y explicativas y de priorización. Esta fase, generalmente tiene una duración de tres días.

Durante la fase de cierre, se pulen, difunden y publican los principales resultados como una colección de requisitos, y se planifican las siguientes diferentes actividades. El proceso de ARM es similar al JAD, pero tiene algunas diferencias significativas que contribuyen a su singularidad. Por ejemplo, en este proceso, los facilitadores son neutrales, las técnicas de dinámica de grupo utilizadas son diferentes de las utilizados en JAD, las técnicas de lluvia de ideas utilizadas son diferentes, y los requisitos son grabados y organizados utilizando diferentes modelos conceptuales.

1. CRITERIOS DE EVALUACIÓN

Los siguientes son los criterios de evaluación que se utilizaron en la realización de esta investigación y que puede ser útil en la selección de un método de elicitación, pero sin duda que existen otros criterios que se podrían utilizar. El punto principal es utilizar un criterio y tener una comprensión común de lo que significan.

- *Adaptabilidad.* El método puede ser usado para generar requisitos en múltiples entornos. Por ejemplo, el método de elicitación funciona igual de bien con un producto software a punto de finalizar que con un proyecto en la fase de planificación.

- *Herramientas de IS asistidas por computador – CASE*. El método incluye una herramienta CASE. El Software Engineering Institute define una herramienta CASE como un producto basado en computadora destinado a apoyar a una o más actividades de Ingeniería de Software en un proceso de desarrollo de software [22].

- *Aceptación de las partes interesadas*. Es probable que las partes interesadas estén de acuerdo con el método de elicitación al analizar sus requisitos. Por ejemplo, el método no es demasiado invasivo en un ambiente de negocios.

- *Fácil implementación*. El método de elicitación no es muy complejo y puede ejecutarse fácil y adecuadamente.

- *Salida gráfica*. El método produce artefactos visuales fácilmente comprensibles.

- *Rápida implementación*. Los ingenieros de requisitos y las partes interesadas pueden ejecutar totalmente el método de elicitación de un plazo de tiempo razonable.

- *Curva de aprendizaje ligera*: Los ingenieros de requisitos y las partes interesadas pueden comprender completamente el método de obtención en un plazo de tiempo razonable.

- *Madurez alta*. El método de elicitación ha experimentado una considerable exposición y análisis en la comunidad de ingeniería de requisitos.

- *Escalabilidad*. El método puede ser utilizado para elicitar los requisitos de proyectos de diferentes tamaños, desde sistemas a nivel empresarial hasta aplicaciones de pequeña escala.

2. RESULTADOS Y DISCUSIÓN

Es preciso tener en cuenta que este enfoque presupone que todos los criterios son igualmente importantes. Si algunos criterios son más importantes que otros, se puede utilizar una media ponderada. Por ejemplo, la disponibilidad de una herramienta CASE puede ser más importante que la salida gráfica. En esta investigación se aplicó un esquema de

ponderación típica teniendo en cuenta criterios como "Muy bueno" con un peso de 3, "Bueno", con un peso de 2, y "Regular" con un peso de 1. Esto no pretende ser una recomendación para utilizar un método específico, cada usuario puede desarrollar sus propios criterios de comparación y clasificación.

En esta investigación, se analizaron los métodos de elicitación antes descritos a través de la experiencia de los equipos de Ingeniería de Requisitos de varias compañías de desarrollo de software. Se hizo una consulta telefónica con los líderes de proyecto y los resultados obtenidos, que constituyen la base para proponer un método de selección, se muestran en la Tabla 1.

Para los estudios de caso de este trabajo, se utilizaron los métodos ARM, IBIS y JAD en tres proyectos diferentes. Estos tres métodos se clasificaron subjetivamente como los candidatos más adecuados para los estudios de caso, dadas las limitaciones de tiempo y esfuerzo que se aplicaron en esta investigación. No sólo se consideró la puntuación total, también la curva de aprendizaje fue un factor importante, y el equipo de trabajo trató de seleccionar los métodos que no fueran demasiado similares entre sí para tener algo de variedad.

Tabla 1
Resultados de la comparación de los métodos de elicitación

	Misuse Cases	SSM	QFD	CORE	IBIS	JAD	FODA	CDA	ARM
Adaptabilidad	3	1	3	2	2	3	2	1	2
Herramienta CASE	1	2	1	1	3	2	1	1	1
Aceptación	2	2	2	2	3	2	1	3	3
Fácil implementación	2	2	1	2	3	2	1	1	2
Salida gráfica	2	2	1	1	2	1	2	2	3
Rápida implementación	2	2	1	1	2	1	2	2	3
Curva de aprendizaje	3	1	2	1	3	2	1	1	1
Alta madurez	2	3	3	3	2	3	2	2	1
Escalabilidad	1	3	3	3	2	3	2	1	2
Totales	**18**	**18**	**17**	**16**	**22**	**19**	**14**	**14**	**18**

La eficacia de IBIS en la elicitación de requisitos de seguridad depende de la calidad de las preguntas de la entrevista. En la mayor medida posible, el alcance de las preguntas deben cubrir toda la gama de requisitos de seguridad que podrían afectar el sistema. Como suele suceder en la elicitación de requisitos, se encontró que el entrevistador debe ser persistente en alentar a las partes interesadas a que sean racionales a la hora de proponer una solución a un problema. Para explicar por qué han elegido esa posición, las partes interesadas pueden naturalmente discutir las ventajas y desventajas entre ellos. La entrevista IBIS, en este caso, sólo tiene que grabar sus declaraciones durante el debate.

Se encontró que las preguntas directas no funcionan muy bien. Por ejemplo, se recomienda hacer preguntas como: en su caso, ¿qué medidas toma para proteger contra errores de configuración? en lugar de ¿cómo responde el sistema a errores de configuración? La última pregunta implica un requisito acordado, a pesar de que puede haber debate sobre si el tema de la pregunta es un requisito en todos. Al igual que el interrogatorio directo en los procesos judiciales, las preguntas directas le permiten a las partes interesadas proveer una respuesta honesta, imparcial y completa a cada pregunta.

El ingeniero de requisitos también debe considerar los artefactos previos del sistema. Tras el examen del primer borrador de preguntas en IBIS para las partes interesadas, se descubrió que muchas de ellas fueron contestadas en la fase de colección de artefactos del proceso. Debido a limitaciones de tiempo y del conocimiento general del proyecto de parte de las partes interesadas, equivocadamente se incluyeron algunas preguntas relacionadas con artefactos tales como: ¿cuáles son los requisitos computacionales mínimos para ejecutar el software? La sugerencia es que las preguntas en IBIS se formulen cuidadosamente para excluir aquellas que han sido contestadas previamente.

En general, ARM fue extremadamente eficaz y, fiel a su nombre, un método rápido de colección de requisitos. Se encontró que simplemente eligiendo el enfoque correcto de la pregunta, el proceso fue muy fácil de adaptar para elicitar requisitos de seguridad. En retrospectiva, la gestión del tiempo perdido fue la mayor falla en este proceso. Debido a la gran cantidad de preguntas que debe plantearse para cada requisito, se recomienda la realización de una gestión estricta del tiempo y de una orientación proactiva de las conversaciones entre las partes interesadas. Los resultados obtenidos son ARM pueden ser un poco sesgados ya que los participantes ya eran expertos en seguridad. Como tal, fueron capaces de generar fácilmente un amplio conjunto de requisitos de seguridad. En otros contextos, es poco probable que todos los participantes tengan esa experiencia, por lo que será necesario que el equipo de Ingeniería de Requisitos necesite revisar algunos conceptos de seguridad con los participantes antes de iniciar la sesión.

Dado que el flujo de trabajo, elementos de datos, las pantallas y los pasos de los reportes de JAD no eran adecuados para la discusión de los requisitos de

seguridad y por lo tanto fueron excluidos, el método resultó ser muy similar a un proceso de entrevista no estructurada. Aunque las entrevistas no estructuradas se utilizaron en otro estudio de caso, no se intentó hacer una comparación directa de los resultados de JAD con los resultados de ese estudio de caso. En esencia, el equipo de trabajo acabó solicitando a las partes interesadas algunas preguntas para el proyecto, por lo que no se hizo un adecuado uso de la capacidad total del método JAD, lo que puede haber sesgado los resultados. La fase de sesiones JAD fue diseñada para requisitos funcionales del desarrollo y no hubo una forma específica para discutir requisitos de calidad como la seguridad. Por lo tanto, el equipo pasó mucho tiempo investigando otros métodos para ayudar a obtener mejores requisitos de seguridad durante la sesión de JAD. Se sugiere utilizar JAD como un método adicional para tratar con requisitos de calidad.

La calidad de los requisitos de seguridad generados depende de la calidad de las preguntas durante la entrevista. Esta relación plantea un gran riesgo para el método JAD. En este caso, el equipo elaboró una serie de preguntas diferentes, simplemente porque las partes interesadas eran profesionales de la seguridad y ya habían examinado los tipos de preguntas utilizadas en otros estudios de caso. Si las partes interesadas hubieran sido menos conscientes de la seguridad, los resultados hubieran sido muy diferentes. El equipo puede no haber sido capaz de obtener toda la información y no haber comprendido los problemas de seguridad fundamentales del proyecto. La variedad de los participantes en las sesiones JAD también es muy importante. En este caso, sólo había unos pocos interesados que participaron en las sesiones y no hubo una adecuada discusión entre ellos. El equipo recomienda que el período de sesiones JAD debe involucrar a todos los interesados y que el facilitador debe animarlos a compartir sus opiniones.

3. CONCLUSIONES

ARM parece más adecuado para elicitar requisitos de seguridad que IBIS o JAD. Los resultados de JAD fueron similares a los que se obtendría a través de un proceso de entrevistas no estructuradas y parece más adaptable para requisitos funcionales de usuario final. No hubo manera específica para discutir requisitos de calidad como la seguridad. Se encontró que IBIS fue efectivo para documentar discusiones de toma de decisiones complejas, pero no proporciona una forma estructurada para generar requisitos de seguridad. Con el fin de obtener un buen conjunto de requisitos de seguridad utilizando IBIS, el equipo de Ingeniería de Requisitos tuvo que generar preguntas cuidadosamente seleccionadas para la entrevista. Los tres métodos de elicitación tienen la debilidad de haber sido diseñados originalmente para centrarse en las características y, como consecuencia, tienden a centrarse en funciones de seguridad y no consideran a la seguridad como una propiedad del sistema.

Es posible que una combinación de métodos pueda funcionar mejor. Se debe considerar esto como parte del proceso de evaluación, suponiendo que haya tiempo y recursos suficientes para evaluar cómo combinar los métodos y para qué combinarlos. También se debe considerar el tiempo necesario para implementar un método de elicitación y el tiempo necesario para aprender una nueva herramienta que lo apoye. Seleccionar un método de elicitación que satisfaga las necesidades de un grupo diverso de interesados ayudará para hacer frente a una amplia gama de requisitos de seguridad.

Las organizaciones necesitan hacer un mejor trabajo para identificar los requisitos de seguridad. No es suficiente con listar los requisitos obvios para estar seguros, como contraseñas seguras, encriptación y mecanismos de control de acceso. Se necesita un enfoque sistemático para garantizar que se capturen los requisitos de seguridad; de lo contrario, es probable que el sistema resultante contenga muchos huecos de seguridad que son evitables. Se recomienda que las organizaciones tomen el tiempo para seleccionar un método de elicitación utilizando un enfoque equilibrado de análisis sistemático como el que se esbozado en este trabajo.

REFERENCIAS

[1] A. Hickey et al. "Requirements Elicitation Techniques: Analyzing the Gap Between Technology Availability and Technology Use". *Comparative Technology Transfer and Society,* Vol. 1, No. 3, pp. 279-302, 2003.

[2] A. Hickey & A. Davis. "A Unified Model of Requirements Elicitation". *Journal of Management Information Systems,* Vol. 20, No. 4, pp. 65-84, 2004.

[3] D. Zowghi & C. Coulin. "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools". In A. Aurum & W. Claes (Eds.) Engineering and Managing Software Requirements. Heidelberg, Germany: Springer-Verlag, 2005.

[4] G. Sindre & A. L. Opdahl. "Eliciting Security Requirements by Misuse Cases". *Proceedings of the 37th International Conference on Technology of Object-Oriented Languages (Tools 37-Pacific 2000)*. Nov. 20-23, Sydney, Australia, 2000.

[5] G. McGraw. "Software Security: Building Security In". Boston: Addison-Wesley, 2006.

[6] I. Jacobson. "Object-Oriented Software Engineering: A Use Case Driven Approach". Boston: Addison-Wesley, 1992.

[7] J. Rumbaugh. "Getting Started: Using Use Cases to Capture Requirements". *Journal of Object-Oriented Programming*, Vol. 7, No. 5, pp. 8-23, 1994.

[8] A. I. Anton; J. H. Dempster & D. F. Siege. "Deriving Goals from a Use Case Based Requirements Specification for an Electronic Commerce System". *Proceedings of the Sixth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2000)*. Jun 5-6, Stockholm, Sweden, 2000.

[9] P. Checkland. "Soft System Methodology in Action". Toronto: John Wiley & Sons, 1990.

[10] Quality Function Deployment. "Frequently Asked Questions About QFD". Online [May 2011].

[11] Systems Designers. "CORE - The Manual". SD-Scicon, 1986.

[12] M. Christel & K. Kang. "Issues in Requirements Elicitation". *Technical Report CMU/SEI-92-TR-012,*

ADA258932. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1992.

[13] G. P. Mullery. "CORE: A Method for Controlled Requirements Specification". *Proceedings of the 4th International Conference on Software Engineering (ICSE-4)*. Sep. 17-19, Munich, Germany, 1979.

[14] A. Finkelstein. "TARA: Tool Assisted Requirements Analysis". In P. Loucopulos & R. Zicari "*Conceptual Modeling, Databases and CASE: An Integrated View of Information Systems Development"*. John Wiley & Sons, 1992.

[15] W. Kunz & H. Rittel. "Issues as Elements of Information Systems". Berkeley: Institute of Urban & Regional Development, 1970.

[16] J. Wood & D. Silver. "Joint Application Development". New York: Wiley, 1995.

[17] K. Kang et al. "Feature-Oriented Domain Analysis Feasibility Study". Technical Report CMU/SEI-90-TR-021, ADA235785. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1990.

[18] L. Kean, L. "Feature-Oriented Domain Analysis". Technical Report CMU/SEI-90-TR-21 ESD-90-TR-222. Software Engineering Institute. Carnegie Mellon University, 1997.

[19] D. Schiffrin. "Approaches to Discourse". Blackwell Publishers Ltd, 1994.

[20] R. Alvarez. "Discourse Analysis of Requirements and Knowledge Elicitation Interviews". *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35).* Jan. 7-10, Big Island, 2002.

[21] R. Hubbard; N. Mead & C. Schroeder. "An Assessment of the Relative Efficiency of a Facilitator-Driven Requirements Collection Process with Respect to the Conventional Interview Method". *Proceedings of the 4th International Conference on Requirements Engineering (ICRE'00)*. June 19-23, Los Alamitos, California, USA, 2000.

[22] M. Dixon. "A single CASE environment for teaching and learning". *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education"*. Leeds, UK, 28-30 June, 2004.