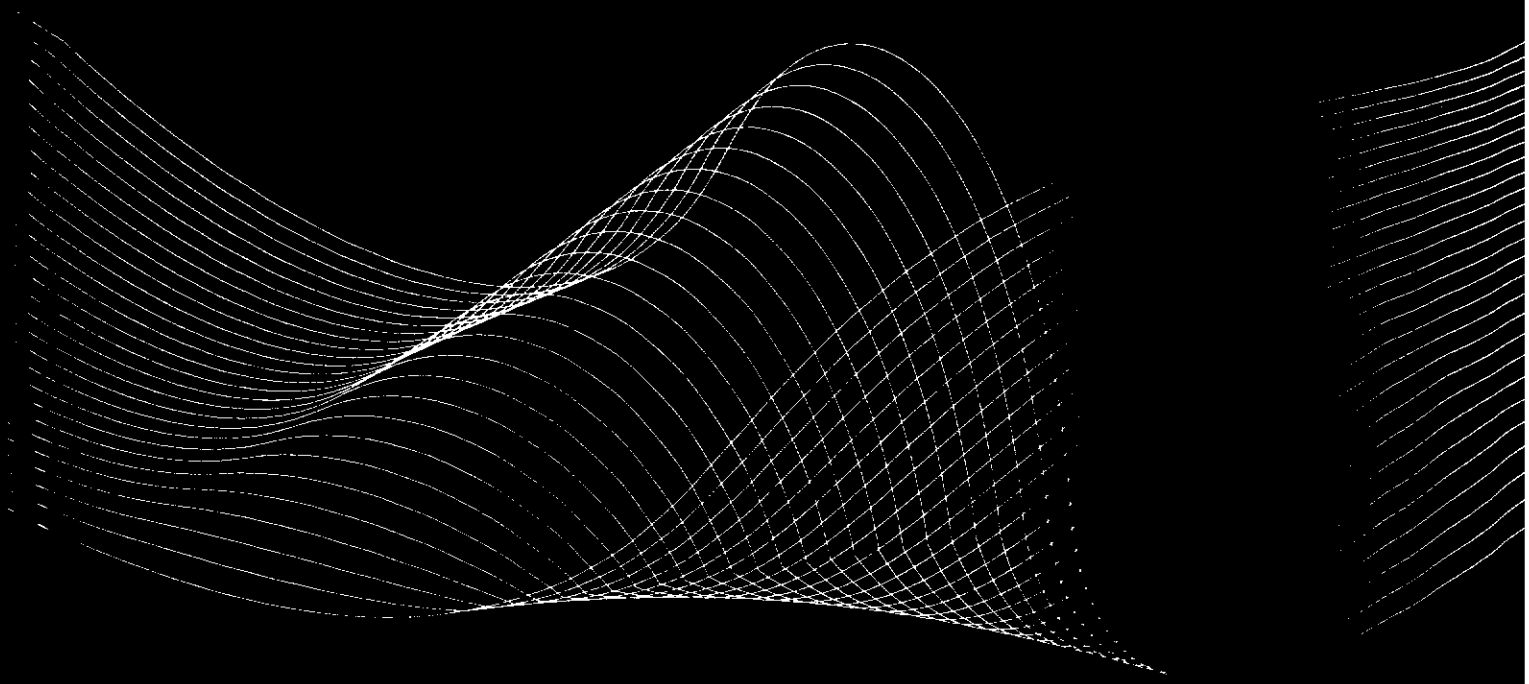
▶**Del 24 al 28 de abril de 2006**
La Plata | Buenos Aires | Argentina

ideas.06

9° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software

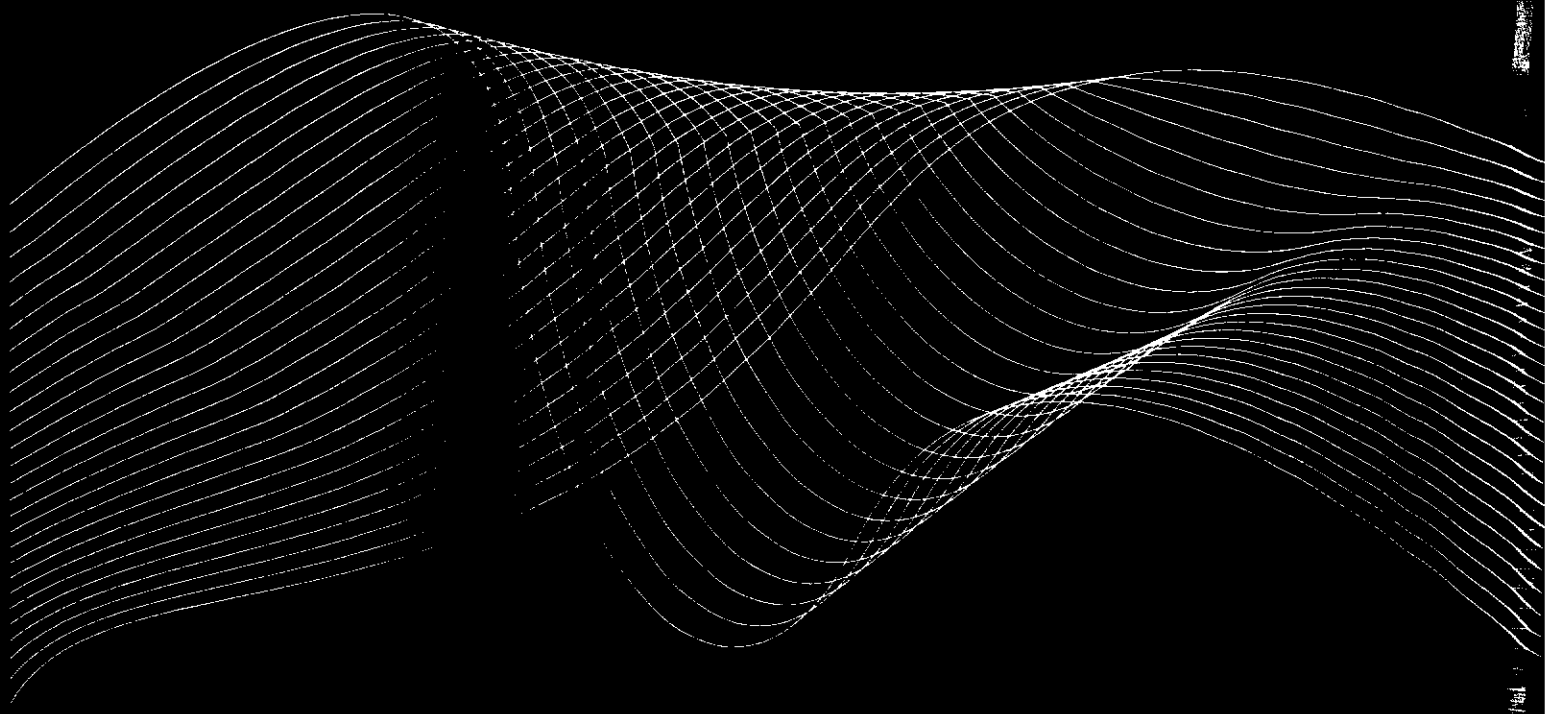


Auspician:

Centro Latinoamericano de Estudios en Informática

Microsoft

SADIO



ISBN-10:950-34-0360-X ISBN-13 :978-950-34-0360-0

Actas

IDEAS'06

9º Workshop Iberoamericano de
Ingeniería de Requisitos y
Ambientes de Software

24 al 28 de Abril de 2006
La Plata, Argentina

Editores
Jaelson Castro
Luca Cernuzzi
Silvia Gordillo

Actas del 9º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software IDEAS'06

ISBN-10: 950-34-0360-X
ISBN-13: 978-950-34-0360-0

PRÓLOGO

El presente volumen contiene los trabajos aceptados y presentados en el IX Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software – IDEAS 06 celebrado en la ciudad de La Plata, Argentina, del 24 al 28 de Abril de 2006.

Muy brevemente quisiéramos mencionar los números y recordar los pasos que dimos juntos para el proceso de evaluación.
Originalmente se han postulado 107 resúmenes que finalmente han resultado en la postulación de 100 artículos. Cada articulo ha sido asignado a 3 revisores y se han discutido las eventuales discrepancias con respecto a un mismo articulo para alcanzar, donde fuera posible, un consenso. Finalmente, en la difícil decisión de aceptación/rechazo hemos adoptado el criterio de aceptar como artículos todos y solo aquellos que obtuvieron un promedio y un promedio ponderado (considerando el nivel de experiencia del evaluador) igual o superior a 4.5 sobre 7; para una sesión especial se han aceptado adicionalmente, en calidad de poster, aquellos artículos que obtuvieron un promedio y un promedio ponderado igual o superior a 4 sobre 7.
Así, el resultado final ha sido que de los 100 artículos originalmente postulados, 33 han sido aceptados para su presentación integral y otros 8 para su presentación como poster.

Consideramos que el nivel de postulación es síntoma de un creciente interés por el workshop y nos hace pensar que él mismo se está afianzando cada vez más como un foro importante de Ingeniería de Software (en términos amplios) para Ibero América. Al mismo tiempo, el porcentaje de aceptación indica un nivel de exigencia interesante para la realidad Iberoamericana que puede ayudar a posicionar mayormente el workshop para obtener un mayor reconocimiento en las calificaciones de los autores en sus respectivos países.

Evidentemente, todo esto no hubiera sido posible sin la valiosa colaboración de distintos actores a quien van nuestros más sinceros agradecimientos. Entre ellos, cabe mencionar a los autores por su esfuerzo en investigación, los revisores, ya sean miembros del comité de programa o adicionales, que han tenido que cumplir un esfuerzo particularmente intenso por la cantidad de postulaciones y el proceso adoptado, los miembros del comité organizador, los disertantes de los tutoriales, el comité de conducción de

IDEAS y todas las demás personas que de distintas formas han colaborado a que este evento pueda ser un ocasión importante para la comunidad científica en nuestra región.

Esperamos puedan disfrutar de IDEAS 06 así como de la calida acogida de La Plata.

Silvia Gordillo	**Jaelson Castro**	**LucaCernuzzi**
Presidente Conferencia	Co-Pte. Comité de Programa	Co-Pte. Comité de Programa

Comité de Programa

Presidencia de la Conferencia

Silvia Gordillo, Universidad Nacional de La Plata, Argentina

Co-Presidentes del Comité de Programa

Jaelson Castro , Universidad Federal do Pernabuco, Brasil
Luca Cernuzzi, Universidad Católica "Nuestra Señora de la Asunción", Paraguay

Miembros del Comité de Programa

Amador Duran, U. Servilla, España
Alexandre Vasconcelos, U.F. Pernambuco,Brasil
Antonio Brogi, U. de Pisa, Italia
Antonio Vallecillo, U. de Málaga, España
Carme Quer, U. P. de Catalunya, España
Cecilia Bastarrica, U. de Chile, Chile
Daniel Riesco, U. de San Luis, Argentina
Ernesto Cuadros, UNAS, Perú
Ernesto Pimentel, U. de Málaga, España
Francisco Ruiz, U. Castilla de la Mancha, España
Guilherme H. Travassos, UFRJ, Brasil
Hernán Astudillo, UTFSM, Chile
Joao Falcão e Cunha, U.do Porto, Portugal
João Araújo, U. Nova de Lisboa, Portugal
José Carlos Maldonado, U. de São Paulo, Brasil
Júlio Leite, PUC-Rio, Brasil
Luis Olsina, U. Nacional de La Pampa, Argentina
Marcelo Campo, UNICEN, Argentina
Marcelo Frias, UBA, Argentina
Mario Piattini, U. Castilla-La Mancha, España
Miguel Katrib, U. de La Habana, Cuba
Óscar Pastor, U. Politécnica de Valencia, España
Pere Botella, U. Politécnica de Catalunya, España
Raúl Monje, UTFSM, Chile
Ricardo de A. Falbo, UFES, Brasil
Tereza Kirker, UNIMEP, Brazil
Xavier Franch, U. P. de Catalunya, España

Revisores adicionales

Gabriel Infante-López - UBA, Argentina
Diego Garbervetsky – UBA, Argentina
Carlos Lopez Pombo - UBA, Argentina
Nazareno Aguirre – UBA, Argentina
Juan Pablo Galeotti – UBA, Argentina
Alejandro Vaisman – UBA, Argentina
Silvia Gordillo – U. Nacional de La Plata, Argentina
Gustavo Rossi - U. Nacional de La Plata, Argentina
Claudia Pons - U. Nacional de La Plata, Argentina
Maria de los Angeles Martín - U. Nacional de La Pampa, Argentina
Hernán Molina - U. Nacional de La Pampa, Argentina
Guillermo Covella - U. Nacional de La Pampa, Argentina
Rodrigo de Oliveira Spinola - UFRJ, Brazil
Marco Pereira Araujo - UFRJ, Brazil
Wladmir Araujo Chapetta – UFRJ, Brazil
Tayana Uchoa Conte – UFRJ, Brazil
Ana Candida Cruz Natali - UFRJ, Brazil
Jobson Massolar da Silva - UFRJ, Brazil
Paula Gomes Mian - UFRJ, Brazil
Carla Silva - UFPE, Brazil
Alex Sandro Gomes – UFPE, Brazil
Sandro Ronaldo Bezerra – UFPE, Brazil
Fernanda Ma. R. Alencar – UFPE, Brazil
Regiane Andrade Brito – UFPE, Brazil
Carlos Menezes de Albuquerque – UFPE, Brazil
Carina Alves – UFPE, Brazil
Thaizel Fuentes - U. de La Habana, Cuba
Abel Marrero - U. de La Habana, Cuba
Marcello Visconti - Universidad Técnico Federico Santa María, Chile
Horst H. von Brand - UTFSM, Chile
Benjamin Piwowarski - U. de Chile, Chile
Sara Corfini – U. de Pisa, Italy
Razvan Popescu - U. de Pisa, Italy
Massimo Cossentino - ICAR – CNR, Italy
Enrique Vargas – U. Católica "Nuestra Señora de la Asunción", Paraguay
Vicente González – U. Católica "Nuestra Señora de la Asunción", Paraguay
Magalí González – U. Católica "Nuestra Señora de la Asunción", Paraguay
Alfredo Paz-Valderrama - UNAS, Perú
Percy Pari Salas – UNAS, Perú

Alfredo Paz-Valderrama - UNAS, Perú
Raul Romero - U. de Málaga ,Spain
Nathalie Moreno - U. de Málaga ,Spain
Manuel F. Bertoa - U. de Málaga ,Spain
José Luis Pastrana - U. de Málaga, Spain
Juan Pablo Carvallo - U. Politécnica de Catalunya, Spain
Xavier Burgués - U. Politécnica de Catalunya, Spain
Enric Mayol - U. Politécnica de Catalunya, Spain
Nuria Rodríguez - U. Politécnica de Catalunya, Spain
Tomás Aluja - U. Politécnica de Catalunya, Spain
Emilio Insfrán – U. Politécnica de Valencia, Spain
Vicente Pelechano – U. Politécnica de Valencia, Spain
Hugo Estrada Esquivel - U. Politécnica de Valencia, Spain
Pedro Valderas - U. Politécnica de Valencia, Spain
Javier Muñoz - U. Politécnica de Valencia, Spain
Marta Ruiz - U. Politécnica de Valencia, Spain
Juan Sánchez Díaz - U. Politécnica de Valencia, Spain
Nelly Condori-Fernandez – U. Politécnica de Valencia, Spain
Alicia Martínez Rebollar - U. Politécnica de Valencia, Spain
Silvia Abrahão - U. Politécnica de Valencia, Spain
Victoria Torres - U. Politécnica de Valencia, Spain
Brian Matthews - CCLRC Rutherford Appleton Laboratory
Simon Lambert - CCLRC Rutherford Appleton Laboratory

Comité Organizador

Co-Presidentes del Comité Organizador

Claudia Pons, Universidad Nacional de La Plata, Argentina
Roxana Giandini, Universidad Nacional de La Plata, Argentina

Miembros

Claudia Banchoff, Universidad Nacional de La Plata, Argentina
Gabriela Pérez, Universidad Nacional de La Plata, Argentina
Ileana Carrizo, Universidad Nacional de La Plata, Argentina

Auspicios

Universidad Nacional de La Plata (UNLP)
Sociedad Argentina de Informática (SADIO)
Centro Latinoamericano de Estudios en Informática (CLEI)
Microsoft Argentina

Diseño Gráfico y Comunicación Visual

Axel Hochegger

INDICE

Análisis Comparativo de Propuestas de Establecimiento de Requisitos de Seguridad para el Desarrollo de Sistemas de Información Seguros

Daniel Mellado[1], Eduardo Fernández-Medina[2] y Mario Piattini[2]

[1] Ministerio de Trabajo y Asuntos Sociales; Gerencia de Informática de la Seguridad Social; Madrid, España
Daniel.Mellado@alu.uclm.es
[2] Grupo ALARCOS, Dpto. de Tecnologías y Sistemas de Información, Centro Mixto de Investigación y Desarrollo de Software UCLM-Soluziona; Universidad de Castilla-La Mancha. Ciudad Real, España, Paseo de la Universidad 4, 13071.
(Eduardo.Fdez-Medina, Mario.Piattini)@uclm.es

Abstract. Las soluciones de seguridad actuales están principalmente centradas en proporcionar defensas de seguridad, en vez de resolver una de las principales causas de los problemas de seguridad, un diseño apropiado de los Sistemas de Información (SI). En este artículo se hace un análisis comparativo de nueve propuestas técnicas relevantes que dan importancia al establecimiento de requisitos de seguridad en el desarrollo de SI. Las propuestas analizadas aportan aspectos muy importantes relativos a la seguridad que pueden ser usados como base para nuevas metodologías o extensiones de las existentes. Sin embargo, satisfacen parcialmente los criterios necesarios para el establecimiento de requisitos de seguridad, con garantías e integración en el desarrollo de SI. Por tanto concluimos que no son lo suficientemente específicas para el tratamiento sistemático e intuitivo de requisitos de seguridad en las primeras fases de desarrollo software, aunque partes de éstas propuestas empleadas complementariamente pueden ser usadas para dicho propósito.

1 Introducción

Hoy en día los sistemas de información son vulnerables a multitud de amenazas. Además, cuanto más se incrementa la complejidad de las aplicaciones y servicios, más aumenta la potencialidad de sufrir brechas de seguridad [28]. Y en la actual Sociedad de la Información, que depende de multitud de sistemas software cuya misión es crítica. Resulta crucial que los SI sean asegurados apropiadamente desde el principio [3, 21], debido a las potenciales pérdidas a las que se enfrentan las organizaciones que confían en todos estos SI.

Como sabemos, es ampliamente aceptado el principio que establece que la construcción de la seguridad en las etapas tempranas del proceso de desarrollo es más eficaz respecto a los costes y tiene como resultado diseños más robustos [18]. Sin embargo, el gran problema es que en la mayoría de los proyectos software la seguridad se trata una vez el sistema ha sido diseñado e implementado. Debido en muchos

casos, a una gestión inapropiada de la especificación de los requisitos de seguridad del nuevo sistema, ya que la denominada fase de especificación de requisitos suele realizarse con unas cuantas descripciones o la especificación de objetivos plasmados en unos pocos folios [11]. Además, habitualmente los requisitos de seguridad no son bien entendidos en sí. De forma que, incluso cuando se intenta especificar los requisitos de seguridad, muchos desarrolladores tienden a describir soluciones de diseño en términos de mecanismos de protección en lugar de realizar proposiciones declarativas sobre el grado de protección requerido [11]. Otro motivo por el que se infravaloran los requisitos de seguridad puede tener su origen en que muchas veces son percibidos como un obstáculo para alcanzar los hitos de los proyectos y suponen un gasto extra de recursos para el proyecto, como así lo demuestra la controversia que rodea al asunto del Retorno de la Inversión en la Seguridad o ROSI [9].

Una parte muy importante en el proceso de desarrollo software para conseguir sistemas software seguros es la denominada Ingeniería de Requisitos de Seguridad. La cual proporciona técnicas, métodos y normas para abordar esta tarea en el ciclo de desarrollo de los SI. Y debería implicar, como afirman Kotonya y Sommerville en [19], el uso de procedimientos repetibles y sistemáticos para asegurar que el conjunto de requisitos obtenidos es completo, consistente y fácilmente comprensible y analizable por parte de los diferentes actores implicados en el desarrollo del sistema. Un buen documento de requisitos debe incluir tanto requisitos funcionales (relativos a los servicios que el software o sistema debe proporcionar) y los no-funcionales (relativos a las denominadas características de calidad, como rendimiento, portabilidad, seguridad etc.) [11]. Por su parte, los requisitos de seguridad deben ser descritos de forma concreta antes de que se diseñe la arquitectura del sistema [2], en consonancia con [22], en el sentido de que la seguridad debe ser considerada durante todo el proceso de desarrollo y debería ser definida conjuntamente con la especificación de requisitos.

Actualmente, las soluciones de seguridad están principalmente centradas en proporcionar defensas de seguridad en vez de resolver una de las principales razones de los problemas de seguridad, que se refieren a un diseño apropiado de los sistemas de información. En este artículo se estudian 9 propuestas técnicas relevantes que dan importancia a la elicitación de requisitos de seguridad en de desarrollo de Sistemas de Información (Brescianiet et al. 2004; Breu et al. 2004 y Breu y Innerhofer–Oberperfler 2005; Firesmith 2003 y 2004; Jennex 2005; Myanmar, Lee, y Yurcik, 2005; Toval et al. 2001 y Gutierrez et al. 2005; Peeters 2005; Poopp et al. 2003; Yu 1997). Estas propuestas serán expuestas brevemente y comparadas en este artículo, sirviendo una primera aproximación al estado del arte actual de los requisitos de seguridad en el desarrollo de Sistemas de Información.

El resto del artículo esta organizado de la siguiente forma: en la sección 2, describiremos cada una de las propuestas que incorporan la elicitación de requisitos de seguridad en las primeras fases del desarrollo del SI. A continuación, en la sección 3, mostraremos el marco de comparación que hemos usado para realizar el análisis comparativo de cada una de estas propuestas, y ofrecemos una comparación de las mismas. Finalmente en la sección 4 expondremos nuestras conclusiones.

2 Propuestas Técnicas que Incorporan Requisitos de Seguridad

Las propuestas que serán analizadas en nuestra comparación se mencionan a continuación:

- Bresciani, et al. 2004: "Requisitos de seguridad en Tropos" [6].
- Breu, et al. 2004 y Breu & Innerhofer–Oberperfler, 2005: "Hacia un desarrollo sistemático de sistemas seguros" [7] y "Análisis de seguridad de SI dirigido por el modelo de negocio" [8].
- Firesmith, 2003 y 2004: "Casos de uso de seguridad" [12] y "Requisitos de seguridad en Open Process Framework" [13].
- Jennex 2005: "Modelando requisitos de seguridad para el desarrollo de SI" [16].
- Myagmar, Lee, y Yurcik, 2005: "Modelado de amenazas como base para los requisitos de seguridad" [23].
- Toval et al. 2001: "Requisitos de seguridad en SIREN" [27] y Gutiérrez, et al. 2005: "Requisitos de seguridad para servicios web basados en SIREN" [14].
- Peeters 2005: "Ingeniería ágil de requisitos de seguridad" [24].
- Popp et al. 2003: "Desarrollo de sistemas con seguridad crítica mediante extensión de los casos de uso" [25].
- Yu 1997: "Requisitos de seguridad basado en el marco de trabajo i*" [29].

Hemos elegido estas propuestas porque la mayoría de ellas tratan de resolver el problema de la seguridad en las primeras fases del desarrollo de SI y hacen énfasis en los requisitos de seguridad para el desarrollo de sistemas de información seguros. A continuación se describen brevemente cada una de las propuestas.

2.1 Requisitos de Seguridad en Tropos (propuesta de Bresciani et al. 2004) [6]

Tropos es una metodología dirigida por los requisitos, orientada a agentes y metas que emplea las nociones básicas tomadas de este tipo de metodologías (meta, tarea, dependencia social, etc.) como elemento central a través de todo el ciclo de desarrollo. Y se considera un método de desarrollo conducido por los requisitos ya que son los artefactos desarrollados durante la etapa de análisis de los requisitos los que marcan el resto de etapas (arquitectura y diseño a bajo nivel) del marco de trabajo. Por tanto, Tropos es un método de desarrollo de sistemas software que se compone de las siguientes etapas fundamentales:

1. Análisis de Requisitos.
   - Análisis de los Requisitos en las Etapas Iniciales: análisis organizacional que define los actores y sus metas.
   - Análisis de los Requisitos en las Etapas Finales: requisitos funcionales y no funcionales (la seguridad entre ellos) refinados.
2. Diseño a Alto Nivel o Diseño Arquitectónico, en el que se define el sistema a partir de sus subsistemas principales y del conjunto de relaciones de control y de información presentes entre ellos.
3. Diseño a Bajo Nivel, en el que se realiza un refinamiento de los subsistemas hasta obtener sus componentes atómicos y sus relaciones..

2.2 Hacia un Desarrollo Sistemático de Sistemas Seguros y Análisis de Seguridad de SI Dirigido por el Modelo de Negocio (propuestas de Breu et al. 2004 [7] y Breu & Innerhofer–Oberperfler, 2005 [8])

Los autores proponen un nuevo modelo de procesos para la ingeniería de seguridad. Este modelo de procesos extiende el desarrollo de software dirigido por casos de uso y orientado a objetos a través de un tratamiento sistemático de los temas relacionados con la seguridad. También introducen la noción de aspectos de seguridad describiendo los más relevantes requisitos de seguridad y las contramedidas en su nivel adecuado de abstracción. Además definen un micro-proceso para el análisis de seguridad que apoya el desarrollo sistemático de componentes seguros dentro del desarrollo iterativo de sistemas.

El proceso central está basado en una aproximación general que puede ser fácilmente mapeado a cualquiera de los modelos de procesos establecidos. En cuanto a los artefactos, éstos esenciales son el Modelo de Negocio (que describe los procesos de trabajo), el Sistema de Requisitos (que describe los casos de uso del sistema), la Arquitectura de Aplicación (que describe los componentes lógicos del sistema y los flujos principales de mensajes), y la Arquitectura Software (que describe la estructura técnica). Basándose en el concepto de construcción iterativa de software, introducen un micro-proceso para el análisis de seguridad, que consta de 5 pasos: elicitación de requisitos de seguridad, análisis de amenazas y riesgos, toma de medidas y comprobar la corrección de las medidas relacionadas y los requisitos. De forma que estos 5 pasos son repetidamente realizados en cada nivel de abstracción durante el desarrollo incremental.

Por último, los autores concluyen que la seguridad de la información es una cuestión del negocio, y por esta razón su gestión debe ser dirigida por el negocio. Además, piensan que una vinculación con los estándares más relevantes y un soporte en la valoración de la conformidad con los mismos, debería ser parte integral de su metodología de gestión de la información de seguridad. Lo cual iría junto con proporcionar una función de la documentación que registre la toma de decisiones.

2.3 Casos de uso de seguridad y requisitos de seguridad en Open Process Framework (propuesta de Firesmith, 2003 [12] y 2004[13])

Donald Firesmith en [13] ofrece unos pasos que permiten definir requisitos de seguridad a partir de plantillas reutilizables. Su análisis de los requisitos de seguridad se basa en dos principios básicos obtenidos de OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [1] basado en los recursos y dirigido por el riesgo. Los pasos en su proceso para la identificación y análisis de los requisitos de seguridad son: identificación de los activos; identificar los tipos de atacantes más probables; identificar las posibles amenazas sobre los activos; determinar los impactos negativos, para cada recurso vulnerable; estimar y priorizar los riesgos de la seguridad en relación con los recursos vulnerables y en base a las amenazas más relevantes y a su impacto potencialmente negativo; seleccionar los subfactores de seguridad, para limitar el riesgo a un nivel aceptable; seleccionar las plantillas relevantes, para cada subfactor y riesgo de la seguridad; identificar los requisitos funcionales relevan-

tes; determinar el criterio de seguridad; determinar la métrica de seguridad y determinar el nivel mínimo aceptable; especificar el requisito.

Adicionalmente, en [12], el autor propone que estos requisitos de seguridad deberían basarse en el análisis de activos y servicios que tienen que ser protegidos y en las amenazas de seguridad de las que estos activos y servicios tienen que ser protegidos. Para ello propone los casos de uso de seguridad, que deben de ser usados para especificar los requisitos de seguridad que la aplicación tiene que cumplir para protegerse satisfactoriamente de las amenazas de seguridad más relevantes [12]. A diferencia de los casos de mal uso, que son un tipo especializado de casos de uso, usados para analizar y especificar amenazas de seguridad

Por último, dado que los sistemas suelen poseer requisitos de seguridad similares el autor sugiere el uso de plantillas para especificar los requisitos de seguridad de forma que sean fácilmente reutilizables entre sistemas.

2.4 Modelando requisitos de seguridad para el desarrollo de SI (propuesta de Jennex, 2005) [16]

Jennex plantea, usando análisis de barreras y el concepto de defensa en profundidad, modificar el paradigma de diseño integrado de Siponen y Baskerville [26] en una metodología más gráfica y fácil de entender.

El análisis de barreras, se trata de un método para identificar las amenazas o peligros y determinar la efectividad de los factores de mitigación actualmente existentes. Es también usado en el momento en que una barrera falla para determinar la causa y corregir el problema determinando la futura acción preventiva. La ventaja de este método es que ayuda a identificar los factores fortuitos/ocasionales y las acciones necesarias para corregir los problemas. Sin embargo, su desventaja es que no asegura que sean reconocidas todas las barreras susceptibles de fallar, ni que los efectos de los riesgos/amenazas sean adecuadamente identificados. Jennex se apoya en los diagramas de barreras, los cuales muestran visualmente los ingredientes necesarios para un accidente. Posteriormente el análisis de barreras es usado para valorar la efectividad total del sistema de barreras y cada una de las barreras individualmente en prevenir el evento que provoca el accidente.

En cuanto a la defensa en profundidad, se trata de un concepto que utiliza múltiples mecanismos de compensación de control para prevenir o reducir una amenaza, y que usa una aproximación basada en capas para aumentar el nivel de esfuerzo requerido para dañar el sistema.

La metodología propuesta por el autor plantea por tanto, usar análisis de diagramas de barreras como un método gráfico de identificación y documentación de requisitos de seguridad. Esta aproximación además usa meta-notación para añadir detalles de seguridad a los existentes diagramas de desarrollo. También se entiende que el proceso sigue la aproximación de diseño integrado de seguridad en el ciclo de vida de desarrollo del software. Con lo que el objetivo del uso de diagramas de barreras en la fase de requisitos es que se identifiquen adecuadamente los requisitos de seguridad.

2.5 Modelado de amenazas como base para los requisitos de seguridad (propuesta de Myagmar, Lee, y Yurcik, 2005) [23]

En esta propuesta los autores parten de la siguiente pregunta, como cuestión importante que hay que hacerse en todo SI, es si ¿son necesarias las características de seguridad del sistema y satisfacen las necesidades de seguridad del mismo?

Ofrecen una visión del proceso de ingeniería de requisitos resumido a través de la (Figura 1. Proceso de ingeniería de requisitos de seguridad):

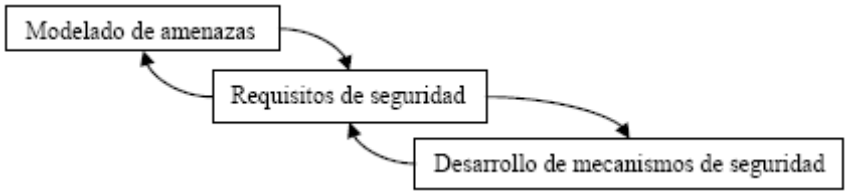


Figura 1. Proceso de ingeniería de requisitos de seguridad

De esta forma las amenazas son analizadas, y o bien se mitiga la amenaza o bien se acepta el riesgo asociada a ella. De tal modo que con una apropiada identificación de amenazas y una apropiada selección de medidas de salvaguarda se reduce la capacidad de los atacantes de hacer un mal uso / abusar del sistema.

El proceso de modelado de amenazas se suele requerir para sistemas de software complejos, soluciones personalizadas de aplicaciones y todos los otros casos en los que no se puede aplicar las listas estándar predefinidas de amenazas. Este proceso de modelado de amenazas planteado por los autores consta de tres pasos de alto nivel:

- Caracterización del sistema. Lo que supone la comprensión de todos los componentes y sus relaciones, definir escenarios de uso y identificar suposiciones y dependencias.
- Identificación de activos y puntos de acceso. Los analistas deben pensar como si fueran atacantes, determinando potenciales enemigos, sus motivaciones, sus objetivos y la información que disponen.
- Identificación de amenazas. Encontrar una lista de amenazas conocidas y vulnerabilidades encontradas en sistemas similares. Y revisar una lista de objetivos de ataque para cada activo. La salida de esta fase es un perfil de amenazas del sistema, describiendo todos los potenciales ataques y cada uno de los cuales necesita ser mitigado o aceptado, para lo cual una posible forma de acometerlo es usando de árboles de ataque.

En cuanto a la especificación de los requisitos de seguridad, éstos son frecuentemente especificados en términos de qué no debe ser permitido que pase. Un modelo de amenazas proporciona la mayor parte de la información necesaria para la elicitación de requisitos. Mediante la conversión de una declaración de amenaza en un requisito incluyendo “no debería” en la declaración, permite componer un conjunto inicial de requisitos de seguridad.

Por último, ya que se considera imposible garantizar el 100% de seguridad, los autores trabajan en conseguir el 100% de aceptación de riesgo. Para ello la gestión de riesgos que plantean consiste en: una valoración del riesgo, reducción del riesgo, y aceptación del riesgo. Para valorar el riesgo de las amenazas identificadas, éstas deben ser priorizadas según el daño que provoquen y su probabilidad de ocurrencia.

2.6 Requisitos de seguridad en SIREN y requisitos de seguridad para servicios web basados en SIREN (propuestas de Toval, et al. 2001 [27] y Gutierrez, et al. 2005 [14])

En [27] Toval et al. definen un proceso de Ingeniería de Requisitos basado en la reutilización de los requisitos de seguridad y que es compatible con MAGERIT (Metodología de Análisis y GEstión de RIesgos del MinisTerio de Administraciones Públicas), el cual a su vez es compatible con el CCF (Common Criteria Framework) definido por el ISO 15408 (ISO/IEC, 1999). La reutilización de los requisitos de seguridad se lleva a cabo mediante dos niveles: A nivel de documentación mediante la definición de una estructura jerárquica de especificaciones de requisitos de seguridad; y a nivel de requisito de seguridad mediante su almacenamiento en un repositorio de requisitos reutilizables. Luego se aplica el método SIREN (SImple REuse of software requiremeNts) en la especificación de los requisitos de seguridad. SIREN define un modelo de proceso, unas directrices básicas, técnicas y herramientas. Las directrices de SIREN consisten de una jerarquía de documentos de especificación de requisitos conjuntamente con una plantilla para cada documento. En la siguiente figura (Figura 2. Jerarquía de documentación de Ingeniería de Requisitos definida en SIREN) se muestra dicha jerarquía.

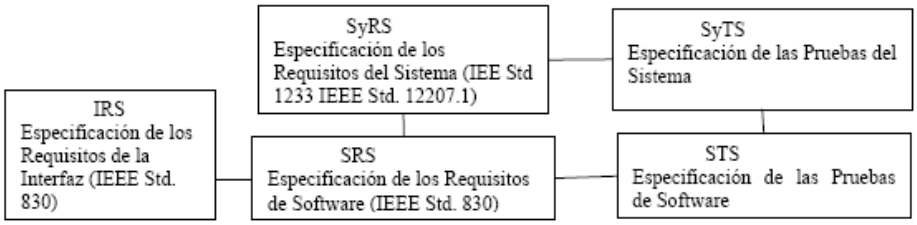


Figura 2. Jerarquía de documentación de Ingeniería de Requisitos definida en SIREN

El modelo del proceso definido es en espiral, y abarca las fases de elicitación de requisitos, análisis y negociación de requisitos, especificación de requisitos y validación. Además, se define un repositorio de requisitos clasificados por dominios y perfiles.

Adicionalmente, en [14] Gutiérrez et al. presentan un catálogo de plantillas de requisitos de seguridad para servicios web (WS) basados en el método de ingeniería de requisitos SIREN. Y centran sus esfuerzos en las plantillas de requisitos de seguridad para los siguientes subfactores: autenticación, autorización, confidencialidad, integridad y privacidad.

2.7 Ingeniería ágil de requisitos de seguridad (propuesta de Peeters, 2005)[24]

Para seguir siendo seguros los sistemas deben cambiar a la vez que lo hace su entorno, preferiblemente anticipándose a los cambios. Partiendo de esta premisas Peeters observa que las demandas de la ingeniería de seguridad encajan con la perspectiva de la agilidad. Por ello, Peeters propone extender prácticas ágiles para tratar con la seguridad con un espíritu informal, comunicativo y dirigido por la seguridad.

Las prácticas actuales de ingeniería ágil de requisitos se basan en las “historias de usuarios”, que son breves e informales descripciones de requisitos escritas por los clientes del sistema. Al igual que se apoyan en la planificación, de modo que el desarrollo ágil es iterativo. Al final de cada iteración, los clientes del sistema verifican si el sistema satisface los requisitos capturados en las “historias de usuarios”, mediante la realización por cada uno de ellos de unas pruebas de aceptación.

Para aumentar la agilidad de la ingeniería de requisitos Peeters plantea el uso de las “historias de abuso”. Éstas identifican como los atacantes podrían abusar del sistema y hacer peligrar los activos de los interesados. De esta manera se facilita el establecimiento de los requisitos de seguridad. Y de la misma forma que las “historias de usuarios”, son breves e informales, y son puntuadas y ordenadas según la percepción de amenaza sobre los activos de los clientes. Una correcta planificación por tanto, supondría considerar las “historias de usuarios” y las “historias de abuso” juntas, lo cual aseguraría un explícito y racional intercambio entre funcionalidad y seguridad.

2.8 Desarrollo de sistemas con seguridad crítica mediante extensión de los casos de uso (propuesta de Popp, et al. 2003) [25]

Esta propuesta proporciona una extensión al proceso convencional de los procesos desarrollados orientados a casos de uso [10, 15]. Este proceso consiste normalmente en 3 actividades en cuanto a la ingeniería de requisitos se refiere:

1. Tratan con los conceptos estáticos del dominio de una aplicación en un modelo de clases denominado Núcleo de la Aplicación. En este punto lo extienden modelando políticas de acceso y propiedades de seguridad basadas en UMLSec [17].
2. Identificación de los casos de uso y su manifestación en un Modelo de Casos de Uso, que son completados con la descripción textual mediante características que analizan las amenazas y las vulnerabilidades de la entrada y la salida. Además esbozan las políticas de seguridad que respondan a las anteriores amenazas. Al modelo resultante lo denominan Modelo de Casos de Uso de Seguridad
3. Integración de las dos vistas anteriores en un modelo orientado a objetos único, mediante, principalmente, la descripción de casos de uso en términos de flujos de mensajes entre los objetos. La extensión consiste en la integración del Modelo de Casos de Uso de Seguridad y el Núcleo de Aplicación refina la política de seguridad en términos de flujos de mensajes entre objetos

2.9 Requisitos de seguridad basado en el marco de trabajo i* (propuesta de Yu, 1997) [29]

El marco de trabajo *i**, es un marco de trabajo que soporta el modelado estratégico orientado a metas y a agentes así como la actividad de análisis de los requisitos, en resumen, ofrece una representación estructural de relaciones intencionales entre actores así como define una serie de conceptos estructurales como agentes o roles intencionales. Estos conceptos proporcionan un marco de trabajo que permite integrar fácilmente otras técnicas y conceptos para tratar la seguridad de los sistemas. La

representación estructural definida en *i** muestra las relaciones de dependencia entre los actores lo cual hace que surjan los aspectos de la seguridad

En la (Figura 3. Proceso de requisitos i*) se muestra el proceso de elicitación y análisis de requisitos, funcionales definidos por *i** y cómo se integra éste con el proceso de elicitación y análisis de los requisitos de seguridad de [20].

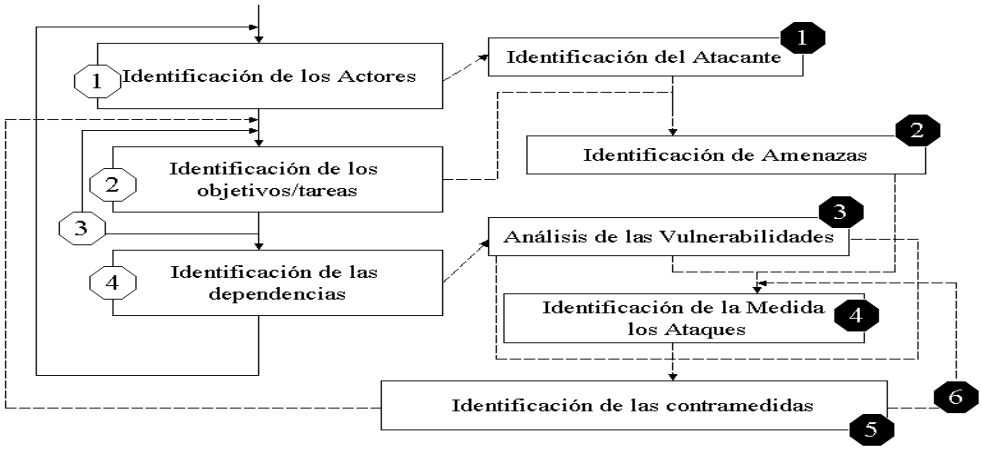


Figura 3. Proceso de requisitos i*

3. Comparación

En esta sección, con el objetivo de obtener una visión general de las diferentes propuestas expuestas anteriormente, se realizará una comparación de las mismas; para lo cual, proponemos un marco de análisis basado en los siguientes criterios:

- Grado de agilidad. Se refiere al grado agilidad de la metodología de desarrollo frente a las metodologías tradicionales planificadas. Es decir, si se trata de una propuesta de metodología más o menos pesada, o más o menos ágil. Para ello nos basaremos en las observaciones realizadas por Boehm y Turner [4, 5]. Quienes proponen un método basado en riesgos a través del cual tratan de mantener en equilibrio ambos métodos, aprovechando las ventajas de ambos (metodologías ágiles y las dirigidas por la planificación o tradicionales) y compensando sus desventajas.
- Soporte. Se refiere a aspectos como herramientas, procedimientos, guías, estándares y casos de estudio, que faciliten y ayuden a usar la propuesta.
- Grado de integración con los otros requisitos software. Se trata de cómo encaja el establecimiento de los requisitos de seguridad con el establecimiento de otros requisitos software (con los demás requisitos no funcionales y también con los funcionales) durante el desarrollo de un Sistema de Información. Para ello se tendrá en cuenta aspectos como el uso de técnicas similares ya existentes para el establecimiento de otros requisitos, como los diagramas UML, grado de paralelización y coordinación con la elicitación de otros requisitos, etc.

- Facilidad de uso. Se refiere a la facilidad con la que la técnica podría ser utilizada sin mucho conocimiento previo de la misma, y sin una formación especial. Para lo cual se tendrá en cuenta aspectos tales como el soporte de ayuda, el uso de técnicas similares ya existentes para otros requisitos, el uso de estándares ampliamente extendidos, etc.
- Contribuciones de la propuesta en cuanto a la seguridad. Aportación/es nuevas que dicha propuesta hace para mejorar el proceso de establecimiento de requisitos de seguridad.

Se calificará cada propuesta de forma cualitativa en la siguiente escala: alta, media-alta, media, media-baja, baja. De manera que para cada una de las propuestas hemos aplicado para cada criterio de comparación un método de ponderación lineal, mediante el cual hemos fijado un peso de manera directa para cada atributo y establecido unos intervalos aproximados para calificar cada criterio de la propuesta como alta, media-alta, media, media-baja, baja, en función de la puntuación aproximada obtenida. Por ejemplo, para el criterio de soporte de ayuda, el hecho de que la propuesta disponga de herramientas CARE (Computer-Aided Requirements Engineering) será el atributo de más peso a la hora de calificar el soporte ofrecido por la propuesta.

En la siguiente tabla (Tabla 1. Comparativa de propuestas) se muestra la comparación de las propuestas de los diferentes autores dentro del marco de análisis que proponemos.

Tabla 1. Comparativa de propuestas

Criterios / Propuestas	Grado de agilidad	Soporte de ayuda	Grado de integración con otros requisitos	Facilidad de uso	Contribuciones respecto a la seguridad
Bresciani, et al. 2004 [6]	Bajo	Medio (ST-Tool)	Media-Alta	Media	▪Desarrollo de sistemas software dirigido por los requisitos, orientada a agentes y metas
Breu, et al. 2004 [7] y Breu & Innerhofer–Oberperfler, 2005 [8]	Bajo	Medio (Herramienta que soporte UML)	Media	Media-Alta	▪"Aspectos de Seguridad". ▪Micro-proceso de análisis de seguridad

Firesmith, 2003 y 2004 [12, 13]	Medio	Medio-Alto (RequisitePro)	Media	Media-Alta	▪ Casos de uso de seguridad. ▪Y definición de requisitos de seguridad a partir de plantillas reutilizables
Jennex, 2005 [16]	Alto	Medio-Bajo	Media-Alta	Alta	▪Diagramas de barreras
Myagmar, Lee, y Yurcik 2005 [23]	Medio-Bajo	Medio	Media	Media-Alta	▪Modelado de amenazas como base para los requisitos de seguridad
Peeters, 2005 [24]	Alto	Medio-Bajo	Alta	Media-Alta	▪Historias de abuso
Popp, et al. 2003 [25]	Medio-Bajo	Alto (Poseidon + MDRLibrary)	Media-Alta	Media-Alta	▪UMLSec
Toval,, et al. 2001 [27] y Gutierrez, et al.2005 [14]	Bajo	Medio-Alto (RequisitePro + plugin)	Media	Media-Alta	▪Reutilización de requisitos de seguridad compatible con MAGERIT ▪Catálogo de requisitos de seguridad reutilizables para WS
Yu, 1997 [29]	Bajo	Medio-Bajo	Media-Alta	Media-Alta	▪Integración de requisitos funcionales y de seguridad

Como se observa en la tabla, y tras el análisis que realizamos, llegamos a la conclusión de que las anteriores propuestas presentan algunas debilidades:

- Como la dificultad de integrarlos con el desarrollo del SI, que provoca un creciente distanciamiento entre el desarrollo de SI y la implementación de la seguridad necesaria.
- La falta de soporte integral/completo del modelado de seguridad a nivel organizacional, conceptual y técnico.
- Estas propuestas nos son lo suficientemente específicas para el tratamiento sistemático e intuitivo de requisitos de seguridad de SI en las primeras fases de desa-

rrollo de software, ya que por ejemplo pocas propuestas apuestan por la reutilización de requisitos de seguridad, cuya utilización garantiza ciclos de desarrollo rápidos y basado en soluciones ya probadas, y ayudan a mejorar la calidad de dichos requisitos para los proyectos sucesivos.

- Y en su mayoría no tratan de integrar los Criterios Comunes (ISO/IEC 15408) en su proceso de establecimiento de requisitos de seguridad, siendo éstos el catálogo de requisitos estándar para la evaluación de sistemas donde la seguridad es crítica. Ni hacen mención de su conformidad o no con los estándares más relevantes de gestión de la seguridad como la norma ISO/IEC 17799.

En definitiva, las propuestas analizadas, satisfacen parcialmente los criterios necesarios para el establecimiento de requisitos de seguridad con garantías y no alcanzan el grado deseado de integración en el desarrollo de SI. Aunque, al mismo tiempo, cada una de estas metodologías aportan aspectos muy importantes relativos a la seguridad que pueden ser usados como base para nuevas metodologías o extensiones de las existentes. Destacándose de cada propuesta sus aportaciones más importantes respecto a la seguridad en la última columna de la Tabla 1.

4. Conclusiones

Hoy en día, en la llamada Sociedad de la Información, dada la criticidad creciente de los SI unido a los nuevos requisitos legales y gubernamentales, se hace necesario el desarrollo de enfoques cada vez más sofisticados para asegurar la seguridad de la información. Normalmente, la seguridad de la información se abordaba desde el punto de vista técnico en la fase de implementación, y aunque se trata de un aspecto importante, consideramos fundamental el tratamiento de la seguridad en todas las fases del desarrollo de SI, especialmente en el establecimiento de los requisitos de seguridad, ya que constituyen la base para la consecución de un SI robusto. Existen varias propuestas metodológicas interesantes al respecto, algunas de ellas han sido descritas y comparadas en este trabajo, aunque presentan algunas debilidades expuestas en la sección anterior. Sin embargo se considera también, que aportan aspectos muy importantes relativos a la seguridad que pueden ser usados para nuevas metodologías o procesos, o bien como extensión o mejora de las existentes.

Por tanto, consideramos que sería interesante obtener una forma sistemática e intuitiva para la elicitación y definición de requisitos de seguridad con garantías. Dicha técnica deberá permitir la integración de los requisitos de seguridad lo más posible con el desarrollo del SI, y permitir la reutilización de los requisitos de unos proyectos a otros. Además tendrá que ser válida para los nuevos SI basados en Internet y en especial para aquellos basados en la arquitectura SOA, apoyados en la tecnología de los Servicios Web. Para ello sería bueno que proporcionara herramientas de apoyo; así como que estuviera basada en estándares de normalización de la definición de requisitos, como pueda ser XML o mediante el uso de plantillas; al igual que el uso de estándares de modelado como UML. Y fuera conforme con los estándares de gestión de la seguridad como la ISO/IEC 17799 o COBIT (las cuales te dicen el qué, pero no el cómo en detalle), así como con la ISO/IEC 15408.

Finalmente, como trabajos futuros, trabajaremos hacia la obtención de un proceso de ingeniería de requisitos de seguridad para el desarrollo de SI seguros, y que tenga en cuenta especialmente las singularidades de los requisitos de seguridad de servicios web, los cuales últimamente han alcanzado una gran popularidad, y en los que la seguridad es un aspecto fundamental para su final adopción por toda la industria como tecnología de integración o '*middleware'*.

Agradecimientos

Este artículo ha sido desarrollado en el contexto de los proyectos CALIPO (TIC2003-07804-CO5-03) y RETISTIC (TIC2002-12487-E), de la Dirección General de Investigación del Ministerio de Ciencia y Tecnología y DIMENSIONS (PBC-05-012-2) de la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha y el FEDER.

Referencias Bibliográficas

1. Alberts, C.J., Behrens, S.G., Pethia, R.D., and Wilson, W.R., *OCTAVE Framework, Version 1.0*. 1999: Networked Systems Survivability Program. p. 84.
2. Barbacci, M.R., Ellison, R., Lattanze, A.J., Stafford, J.A., Weinstock, C.B., and Wood, W.G., *Quality Attribute Workshops (QWAs). Third Edition.*, in *Architecture Tradeoff Analysis Initiative*. 2003, Carnegie Mellon. Software Engineering Institute. p. 36.
3. Baskeville, R., *The development duality of information systems security*. Journal of Management Systems, 1992. **4**(1): p. 1-12.
4. Boehm, B. and Turner, R., *Observations on Balancing Discipline and Agility*. 2003: Agile Development Conference (ADC '03). p. 32.
5. Boehm, B. and Turner, R., *Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods*. 2004: ICSE'04. p. 718-719.
6. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A., *Tropos: Agent-Oriented Software Development Methodology*. 2004: Journal of Autonomous Agents and Multi-Agent System. p. 203-236.
7. Breu, R., Burger, K., Hafner, M., and Popp, G., *Towards a Systematic Development of Secure Systems*. 2004: WOSIS 2004.
8. Breu, R. and Innerhofer–Oberperfler, F., *Model based business driven IT security analysis*. 2005: SREIS 2005.
9. Cavusoglu, H., Mishra, B., and Raghunathan, S., *A Modelo for Evaluating IT Security Investments.* Commun. ACM, 2004. **47**(7): p. 87-92.
10. D'Souza, D.F. and Wills, A.C., *Objects, Components & Frameworks with UML: The Catalysis Approach*. 1998: Addison-Wesley Publishing Company.
11. Fernández-Medina, E., Moya, R., and Piattini Velthus, M., *Gestión de Requisitos de Seguridad*, in *Seguridad de las Tecnologías de la Información "La construcción de la confianza para una sociedad conectada"*, AENOR, Editor. 2003. p. pp 593-618.

12. Firesmith, D.G., *Security Use Cases*. 2003: Journal of Object Technology. p. 53-64.
13. Firesmith, D.G., *Specifying Reusable Security Requirements*. 2004: Journal of Object Technology. p. 61-75.
14. Gutiérrez, C., Moros, B., Toval, A., Fernández-Medina, E., and Piattini, M., *Security Requirements for Web Services based on SIREN*. Symposium on Requirements Engineering for Information Security (SREIS-2005), together with the 13th IEEE International Requirements Engineering Conference – RE'05, 2005.
15. Jacobson, I., Booch, G., and Rumbaugh, J., *The Unified Software Development Process*. 1999: Addison-Wesley Longman Inc.
16. Jennex, M.E., *Modeling security requirements for information systems development*. 2005: SREIS 2005.
17. Jürjens, J., *Secure Systems Development with UML*. 2005: Springer. 309.
18. Kim., H.-K., *Automatic Translation Form Requirements Model into Use Cases Modeling on UML*, C. Youn-Ky, Editor. 2005: ICCSA 2005.
19. Kotonya, G. and Sommerville, I., *Requirements Engineering Process and Techniques*. 1998.
20. Liu, L., Yu, E., and Mylopoulus, J., *Security and Privacy Requirements Analysis within Social Setting*. 2003: 11th IEEE International Requirements Engineering Conference.
21. McDermott, J. and Fox, C. *Using Abuse Case Models for Security Requirements Analysis*. in *Annual Computer Security Applications Conference*. 1999. Phoenix, Arizona.
22. Mouratidis, H., Giorgini, P., Manson, G., and Philp, I. *A Natural Extension of Tropos Methodology for Modelling Security*. in *Workshop on Agent-oriented methodologies, at OOPSLA 2002*. 2003. Seattle, WA, USA.
23. Myagmar, S., J. Lee, A., and Yurcik, W., *Threat Modeling as a Basis for Security Requirements*. 2005: SREIS 2005.
24. Peeters, J., *Agile Security Requirements Engineering*. 2005: SREIS 2005.
25. Popp, G., Jürjens, J., Wimmel, G., and Breu, R., *Security-Critical System Development with Extended Use Cases*. 2003: 10th Asia-Pacific Software Engineering Conference. p. 478-487.
26. Siponen, M. and Baskerville, R., *A new paradigm for adding security into IS development methods,*. 2001: 8th Annual Working Conference on Information Security Management andSmall Systems Security.
27. Toval, A., Nicolás, J., Moros, B., and García, F., *Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach*. 2001: Requirements Engineering Journal. p. 205-219.
28. Walton, J.P., *Developing a Enterprise Information Security Policy*. 2002, ACM Press: Proceedings of the 30th annual ACM SIGUCCS conference on User services.
29. Yu, E., *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*. 1997: 3rd IEEE International Symposium on Requirements Engineering (RE'97). p. 226-235.