Patrocinadores

Entidades Organizadoras

- AdaSpain.
- Asociación de Enseñantes Universitarios de la Informática (AENUI).
- Asociación de Técnicos Informáticos (ATI).
- Asociación Española para la Inteligencia Artificial (AEPIA).
- Asociación para la Interacción Persona-Ordenador (AIPO).
- Asociación para el Desarrollo de la Informática Educativa (ADIE).
- Ayuntamiento de Zaragoza.
- Capítulo Español de la IEEE Computational Intelligence Society.
- Comité Español de Automática (CEA).
- Conferencia de Decanos y Directores de Informática (CODDI) de las Universidades Españolas.
- Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza.
- European Society for Fuzzy Logia and Technology (EUSFLAT).
- Federación de Asociaciones de Ingenieros en Informática (AI2).
- W3C España (World Wide Web Consortium).
- Programa Nacional de Tecnologías Informáticas - Dirección General de Investigación. Ministerio de Educación y Ciencia.
- Red Española de Metaheurísticas.
- Red Española de Minería de Datos y Aprendizaje.
- Sección Española de la European Association for Computer Graphics (EUROGRAPHICS).
- Sociedad de Arquitectura y Tecnología de Computadores (SARTECO).
- Sociedad de Ingeniería del Software y Tecnologías de Desarrollo del Software (SISTEDES).
- Universidad de Zaragoza.

ISBN: 978-84-9732-595-0

CEDI 2007 XII Jornadas de Ingeniería del Software y Bases de Datos | JISBD'07 |

CEDI 2007

II CONGRESO ESPAÑOL DE INFORMÁTICA

ZARAGOZA SPAIN

AUDITORIO PALACIO DE CONGRESOS
11 AL 14 DE SEPTIEMBRE DE 2007

XII Jornadas de Ingeniería del Software y Bases de Datos

| JISBD'07 |

EDITOR
Xavier Franch

ACTAS DE LAS XII JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS

EDITOR

Xavier Franch

PATROCINA

InterSystems

COLABORA

THOMSON

ACTAS DE LAS XII JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS (JISBD'07)

Magallanes, 25; 28015 Madrid, ESPAÑA
Teléfono 91 4463350
Fax: 91 4456218
clientes@paraninfo.es

ISBN: 978-84-9732-595-0
Depósito legal: M-

Maquetación: Los Editores
Coordinación del proyecto: @LIBROTEX
Portada: Estudio Dixi
Impresión y encuademación: FER Fotocomposición, S. A.

IMPRESO EN ESPAÑA-PRINTED IN SPAIN

Comité Ejecutivo

Presidente del Comité de Programa

Xavier Franch (Universitat Politècnica de Catalunya)

Secretario de la Comisión Permanente

Mario Piattini (Universidad de Castilla-La Mancha)

Coordinadora de Tutoriales

Ana M. Moreno (Universidad Politécnica de Madrid)

Coordinador de Talleres

Vicente Pelechano (Universidad Politécnica de Valencia)

Coordinador de Demostraciones

Antonio Vallecillo (Universidad de Málaga)

Coordinador de la Sesión de Divulgación de Trabajos Relevantes ya Publicados

Oscar Díaz (Universidad del País Vasco)

Composición y Maquetación de Actas

Jordi Marco (Universitat Politècnica de Catalunya)

Organización y Relaciones con CEDI 2007

Fran J. Ruiz (Universidad de Zaragoza)
M. Elena Gómez (Universidad de Zaragoza)
Javier Tuya (Universidad de Oviedo)

Comité Organizador

Presidente del CEDI

Alberto Prieto (Universidad de Granada)

Presidente del Comité Ciéntifico

Juan J. Moreno (Universidad Politécnica de Madrid)

Presidente del Comité Organizador CEDI 2007

Victor Viñals (Universidad de Zaragoza)

Coordinador de Actividades Plenarias CEDI 2007

José Duato (Universidad Politécnica de Valencia)

Secretario del CEDI 2007

José A. Castellanos (Universidad de Zaragoza)
José A. Bañares (Universidad de Zaragoza)

Comité de Programa

Alberto Abelló, Univ. Polit. Catalunya
Silvia Abrahão, Univ. Polit. Valencia
Jesus Aguilar, Univ. Sevilla
José Aldana, Univ. Málaga
Bárbara Álvarez, Univ. Polit. Cartagena
María J. Aramburu, Univ. Jaume I
João Araújo, Univ. Nova de Lisboa
Orlando Belo, Univ. do Minho
Rafael Berlanga, Univ. Jaume I
Pere Botella, Univ. Polit. Catalunya
Nieves Brisaboa, Univ. Coruña
Isabel S. Brito, Inst. Polit. Beja
Coral Calero, Univ. Castilla-La Mancha
Carlos Canal, Univ. Málaga
José M. Cavero, Univ. Rey Juan Carlos
Matilde Celma, Univ. Polit. Valencia
Rafael Corchuelo, Univ. Sevilla
Dolors Costal, Univ. Polit. Catalunya
Yania Crespo, Univ. Valladolid
Oscar Dieste, Univ. Polit. Madrid
Javier Dolado, Univ. País Vasco
João Falcão e Cunha, Univ. Porto
Pablo de la Fuente, Univ. Valladolid
Lidia Fuentes, Univ. Málaga
Mario Gaspar da Silva, Univ. Lisboa
Marcela Genero, Univ. Castilla-La Mancha
Cristina Gómez, Univ. Polit. Catalunya
Jaime Gómez, Univ. Alicante
Alfredo Goñi, Univ. País Vasco
Juan Hernández, Univ. Extremadura
Jon Iturrioz, Univ. País Vasco
Natalia Juristo, Univ. Polit. Madrid
Patricio Letelier, Univ. Polit. Valencia
Antonia Lopes, Univ. Lisboa
Adolfo Lozano, Univ. Extremadura
Esperanza Marcos, Univ. Rey Juan Carlos
Eduardo Mena, Univ. Zaragoza
Ana Moreira, Univ. Nova de Lisboa
Juan J. Moreno, Univ. Polit. Madrid
Juan M. Murillo, Univ. Extremadura
Oscar Pastor, Univ. Polit. Valencia
Antonio Polo, Univ. Extremadura
Carme Quer, Univ. Polit. Catalunya
Celia Ramos, Univ. Algarve
Isidro Ramos, Univ. Polit. Valencia
José Riquelme, Univ. Sevilla
Antonio Rito, Univ. Técnica de Lisboa
Antonio Ruíz, Univ. Sevilla
Francisco Ruíz, Univ. Castilla-La Mancha
José Samos, Univ. Granada
Fernando Sánchez, Univ. Extremadura
Juan Sánchez, Univ. Polit. Valencia
Ernest Teniente, Univ. Polit. Catalunya
Miguel Toro, Univ. Sevilla
Ambrosio Toval, Univ. Murcia
Juan C. Trujillo, Univ. Alicante
Javier Tuya, Univ. Oviedo
Belén Vela, Univ. Rey Juan Carlos
Cristina Vicente, Univ. Polit. Cartagena

Comité Asesor para la Selección de Trabajos de Prestigio

Oscar Díaz (Presidente), Univ. País Vasco
Alan Davis, Univ. of Colorado
Neil A.M. Maiden, City Univ. London
Timos Sellis, Nat. Technical Univ. Athens

Revisores Adicionales

César J. Acuña
Amaia Aguirregoitia
Diego Alonso
David Benavides
Jordi Cabot
Paloma Cáceres
Javier Cámara
Dante Carrizo
Pedro J. Clemente
Jose M. Conejero
Javier Cubo
Norberto Díaz
Amador Durán
Sergio España
Mauricio Espinoza
Ismael Etxeberria
Antonio Fariña
Raul Fernandez
L. Fredlund
Antonielly Garcia
Antonio Cesar Gómez
Ángel Herranz
Sergio Ilarri
Miguel Ángel Laguna
Maria Lencastre
Marta López
Francisco Javier Lucas
María Esperanza Manso
Julio Mariño
José Manuel Marqués
Francisco Martínez
Jorge Martínez
Miguel Ángel Martínez
Fernando Molina
Ana M. Moreno
Elena Navarro
Ismael Navas
Isabel Nepomuceno
Juan A. Nepomuceno
Joaquín Nicolás
Guadalupe Ortiz
Juan Angel Pastor
Joaquin Peña
Jenifer Pérez
Juan Manuel Pérez
Beatriz Pontes
Álvaro Prieto
Antonia M. Reina
Domingo Savio Rodríguez
Roberto Rodríguez
Oscar Romero
Fran J. Ruiz
Angeles Saavedra
Gwen Salaün
Pedro Sánchez
André L. Santos
Diego Seco
Jesús Serrano
Encarna Sosa
Toufik Taibi
Raquel Trillo
José Antonio Troyano
Juan Manuel Vara

Sistema Automático de Revisión

Quercus Software Engineering Group

Jose Javier Berrocal Universidad de Extremadura

Conferencia auspiciada por

Prólogo

Respondiendo a su cita anual, las XII Jornadas de Ingeniería del Software y Bases de Datos (JISBD) se han celebrado en Zaragoza, entre el 11 y el 14 de septiembre de 2007. Las Jornadas representan un punto de encuentro de la comunidad investigadora en ingeniería del software y en bases de datos. En sus inicios se celebraron dos eventos diferenciados, las Jornadas de Ingeniería del Software y las Jornadas sobre Investigación y Docencia en Bases de Datos. Posteriormente, en 1999, ambos eventos se unificaron en uno solo, reflejando la interrelación existente entre estas disciplinas. En esta duodécima edición, las Jornadas han constituido, una vez más, un punto de encuentro en el que profesionales y académicos de España, Portugal y Latinoamérica, de ambos campos, han podido compartir experiencias y resultados entre distintos grupos de investigación, desarrollo e innovación tecnológica.

Actualmente, JISBD es un evento auspiciado por Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES, http://www.sistedes.org). Entre los fines de dicha organización destacan el de promover la investigación, la innovación y la transferencia de tecnología entre los distintos agentes involucrados en el avance las tecnologías del Software y el de fomentar actividades con otras asociaciones nacionales e internacionales con fines similares, consiguiendo así proporcionar una mayor visibilidad a la investigación de sus asociados.

Al igual que en 2005, las XII Jornadas de Ingeniería del Software y Bases de Datos se han realizado en el marco del II Congreso Español de Informática (CEDI 2007). Esto ha permitido a los participantes de las Jornadas participar en las diversas actividades de CEDI de interés para toda la comunidad de investigación en Informática, tales como conferencias invitadas y mesas redondas. La celebración cada dos años de JISBD en el marco de CEDI encaja con los objetivos citados de dicha organización.

Este volumen recoge los trabajos seleccionados por el Comité de Programa de JISBD'07. Se recibieron un total de 87 contribuciones de 9 países: España, Portugal, Argentina, Brasil, Chile, Colombia, Cuba, México y Venezuela. Cada contribución fue revisada por tres miembros del Comité de Programa. Posteriormente, se abrió una fase de discusión en la que se debatieron en mayor profundidad algunos trabajos y eventualmente se pidieron revisiones adicionales para ellos; asimismo, algunos trabajos se aceptaron condicionalmente, pendientes de verificar que la versión definitiva trataba adecuadamente los comentarios de los revisores; gracias al esfuerzo de los autores, todos estos trabajos fueron finalmente aceptados. Como resultado de todo el proceso, se configuró un programa compuesto por 30 artículos. Adicionalmente, se seleccionaron 5 trabajos más para su presentación como artículos cortos. Además, en esta edición de JISBD se recogió la posibilidad de presentar trabajos ya publicados en foros de prestigio reconocido. Se seleccionaron 4 artículos de esta modalidad. Finalmente, destacamos la celebración de una sesión para la presentación de herramientas, cuya convocatoria tuvo una acogida excelente por parte de la comunidad de JISBD, de manera que en dicha sesión se programaron un total de 19 demostraciones de herramientas.

El día previo a la conferencia, se organizaron un total de 7 talleres y un tutorial. Estos eventos están ganando importancia a cada nueva edición de JISBD y en el caso de los talleres, están creando sus propias comunidades con intereses más específicos. Algunos talleres ya están plenamente consolidados y llegan a acumular hasta un total de 8 ediciones. Cabe destacar que a partir de este año, las actas de los talleres se recogen en una publicación única en formato electrónico, con el soporte de SISTEDES, para potenciar la difusión de los trabajos presentados.

En referencia al programa, mencionar la participación de dos conferenciantes invitados de reconocido prestigio, siguiendo la pauta de ediciones anteriores. La primera conferencia impartida por Stephen Mellor, miembro del Object Management Group, y con un largo historial en la formulación de métodos para el análisis orientado a objetos. La segunda conferencia a cargo del profesor

John Mylopoulos, que posee igualmente una dilatada experiencia en diversos ámbitos de la ingeniería del software. La presencia de estos dos investigadores representó un elemento importante en el programa de las Jornadas.

Quisiera destacar un hecho que no por obvio, deja de ser merecedor de mención. La celebración de un evento de las características de JISBD, con una participación cada vez más numerosa y consolidada, y con unas exigencias de calidad que se van incrementando en cada edición, no podría realizarse sin la dedicación totalmente desinteresada de un gran número de personas. Desde el punto de vista científico, el trabajo en equipo desarrollado por los miembros del Comité Ejecutivo, en cuyo seno se han debatido los temas más candentes en la configuración de la oferta científica del congreso; y por supuesto la ardua y puntual labor de revisión efectuada por los miembros del Comité de Programa y los revisores adicionales. Desde el punto de vista organizativo, destacar la gran dedicación de los miembros del Comité Ejecutivo responsables de las tareas de enlace con CEDI, y la labor del Grupo Quercus de Ingeniería del Software de la Universidad de Extremadura, quienes han estado a cargo de todo el sistema de recepción y revisión de artículos. También deseo agradecer el soporte recibido por las entidades patrocinadoras y colaboradoras, y en especial la labor de respaldo de SISTEDES, tanto por lo que se refiere a apoyo logístico como a tareas de difusión, como ya se ha comentado. Y por último, especialmente, a los autores de los trabajos enviados a JISBD'07, en definitiva son ellos los que hacen posible la celebración del evento.

Finalmente, desear que el volumen que ahora tienes en tus manos, y que refleja el estado del arte en la investigación en Ingeniería del Software y Bases de Datos en la comunidad de habla hispana y portuguesa, sea de utilidad para tu trabajo.

Zaragoza, Septiembre 2007
Xavier Franch (editor)

Índice

CONFERENCIAS INVITADAS

TUTORIAL

TRABAJOS RELEVANTES YA PUBLICADOS

DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS

ALMACENES Y MINERÍA DE DATOS

PRUEBAS DEL SOFTWARE

TECNOLOGÍAS DE BASES DE DATOS

LÍNEAS DE PRODUCTO. ORIENTACIÓN A ASPECTOS

REQUISITOS. METAMODELADO EN MEDICIÓN

MODELIZACIÓN CONCEPTUAL DE DATOS

ARQUITECTURAS SOFTWARE

MODELOS DE CALIDAD

PROCESOS

ARTÍCULOS CORTOS

DEMOSTRACIONES

Gestión Integral de Requisitos de Seguridad en Líneas de Producto Software

Daniel Mellado
Centro Informático del Instituto Nacional de la Seguridad Social, Gerencia de Informática de la Seguridad Social, Ministerio de Trabajo y Asuntos Sociales. Madrid, España
Daniel.Mellado@alu.uclm.es

Eduardo Fernández-Medina
Grupo ALARCOS
Departamento de Tecnologías y Sistemas de Información.
Universidad de Castilla-La Mancha
Paseo de la Universidad 4, 13071
Ciudad Real, España
Eduardo.FdezMedina@uclm.es

Mario Piattini
Grupo ALARCOS
Departamento de Tecnologías y Sistemas de Información.
Universidad de Castilla-La Mancha
Paseo de la Universidad 4, 13071
Ciudad Real, España
Mario.Piattini@uclm.es

Resumen

La gestión de los requisitos de seguridad es especialmente importante en las líneas de producto software, debido a que una brecha o vulnerabilidad de seguridad puede provocar problemas a todos los productos de la línea y afectar a todo el ciclo de vida. La principal contribución de este trabajo es proporcionar un proceso que facilite la ingeniería de requisitos de seguridad, especialmente adaptado para el desarrollo basado en líneas de producto software, así como un prototipo de una herramienta (add-in del IBM/Rational RequisitePro) a medida que facilite soporte automatizado para la aplicación de dicho proceso. Este proceso trata los requisitos de seguridad y el modelo de variabilidad de la seguridad desde las primeras fases del proceso de desarrollo de una forma intuitiva y sistemática, mediante las últimas técnicas de requisitos de seguridad y junto con la integración de los Criterios Comunes (ISO/IEC 15408) en el desarrollo de la línea. Asimismo, se facilita que los productos de la línea sean conformes con los estándares de seguridad más relevantes (ISO/IEC 27001, ISO/IEC 17799, ISO/IEC 15408 e ISO/IEC 21827) en lo relativo a la gestión de requisitos de seguridad, al igual que se ayuda en la reutilización de los artefactos de seguridad mediante un repositorio de recursos de seguridad, implementado por la herramienta.

Palabras clave: líneas de producto, requisitos de seguridad, ingeniería de requisitos, seguridad, Criterios Comunes.

1. Introducción

En los últimos años se está observando un incremento en la demanda de software y en su complejidad requerida. Por ello, hoy, multitud de sistemas se están desarrollando basándose en el paradigma de ingeniería de Líneas de Producto Software (LPS) para poder alcanzar los niveles deseados de calidad y mejorar la productividad, ya que las LPS ayudan a reducir significativamente el tiempo de puesta en producción y los costes de desarrollo, mediante la reutilización de todo tipo de artefactos [4, 5].

Debido a la complejidad y a la naturaleza extensiva de las LPS, la seguridad y la ingeniería de requisitos son mucho más importantes para la puesta en práctica del desarrollo basado en LPS, de lo que ya son para el desarrollo de un Sistema de Información (SI), ya que una brecha de seguridad o vulnerabilidad en la línea puede provocar importantes problemas a largo plazo a todos los productos de la misma [12].

En varios estudios recientes, como en [15, 18, 23], se defiende el principio que establece que la seguridad debería considerarse desde las primeras fases del desarrollo y que los requisitos de seguridad deberían definirse junto con los demás requisitos del SI, ya que esto permite soluciones más eficientes y robustas así como ayuda a reducir los conflictos entre los requisitos de seguridad y los demás requisitos. Sin embargo, sin una herramienta CARE (Computer-Aided Requirements Engineering), la aplicación de cualquier metodología o proceso de ingeniería de requisitos está avocado al fracoso si se tiene que realizar de forma manual [6]. Por lo tanto, la

Actas de las XII Jornadas de Ingeniería del Software y Bases de Datos, JISBD'2007, pp.177-187 ISBN: 978-84-9732-595-0

disciplina conocida como Ingeniería de Requisitos de Seguridad [16] junto con las herramientas CARE que dan soporte, son una parte muy importante en el proceso de desarrollo software y especialmente dada su complejidad para conseguir LPS seguras, ya que facilitan técnicas, métodos y normas para abordar esta tarea desde las primeras fases del desarrollo e implica 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.

En [20, 21], realizamos un análisis comparativo de las propuestas más recientes y relevantes relativas a los requisitos de seguridad en SI y las herramientas que los soportan, analizándose entre otros estudios los siguientes: [8, 9, 19, 25, 27, 28], etc. Después de haberlas analizado junto con las propuestas más importantes sobre gestión de requisitos en LPS y herramientas de gestión de requisitos para LPS, como [5, 12, 14, 24, 26], así como las arquitecturas de seguridad de referencia para LPS, como [1, 7, 10], llegamos a la conclusión de que las propuestas existentes no eran lo suficientemente específicas para facilitar la gestión de requisitos de seguridad en LPS de una forma sistemática e intuitiva. Al no facilitar una integración concreta de las actividades relativas a la gestión de requisitos de seguridad en el desarrollo basado en LPS ni en proporcionar un soporte intuitivo, sistemático y metodológico automatizado específico para la gestión de requisitos de seguridad y del modelo de variabilidad de seguridad de la línea. Asimismo, no se ayuda a que las LPS y sus productos sean conformes respecto a la gestión de requisitos de seguridad con los estándares de seguridad internacionales más importantes (como ISO/IEC 15408, ISO/IEC 17799, ISO/IEC 27001 o ISO/IEC 21827).

Por último, en este artículo con el objetivo de tratar de resolver los aspectos mejorables en la actual gestión de requisitos de seguridad en LPS, se presentan ambos sub-procesos de los que se compone SREPPLine (Security Requirements Engineering Process for software Product Lines) y sus actividades correspondientes, así como una breve descripción del prototipo de herramienta de gestión de requisitos de seguridad para LPS, denominada SREPPLineTool, que ha sido rediseñada y desarrollada a partir de SREPTOOL [22], con el fin de dar soporte automatizado a la aplicación de los dos sub-procesos de SREPPLine.

El resto del artículo está organizado de la siguiente forma: en la sección 2, se resumen los principales conceptos de la ingeniería de requisitos en las LPS. A continuación, en la sección 3, se presenta SREPPLine. Seguidamente en la sección 4, se describirá brevemente la herramienta de soporte al proceso, SREPPLineTool. Y por último, en la sección 5, presentamos nuestras conclusiones y trabajos futuros.

2. Un Resumen de la Ingeniería de Requisitos en Líneas de Producto Software

Una Línea de Producto Software (LPS) es "un conjunto intensivo de sistemas software que comparten un conjunto común y gestionado de características (features, entendidas como una característica visible para el usuario final del sistema [13]), dónde estas características están pensadas para satisfacer las necesidades especificas de una misión o de un segmento de mercado. Asimismo, los productos son desarrollados de una forma pre-establecida a partir de un conjunto común de componentes" [5].

Por lo tanto, los requisitos de una línea de productos definen los productos de dicha línea y sus características comunes y variables [5]. La gestión de requisitos para líneas de productos debe gestionar los requisitos de la línea de productos y los requisitos de los productos concretos de la línea. Se tiene que hablar por tanto de gestión de requisitos del dominio y gestión de requisitos de la aplicación, siguiendo a [24] se hablará de los procesos de gestión de requisitos del dominio y de la aplicación.

3. SREPPLine: Proceso de Ingeniería de Requisitos de Seguridad para Líneas de Producto Software

El Proceso de Ingeniería de Requisitos de Seguridad para Líneas de Producto Software (SREPPLine) es un add-in de actividades (que se descomponen en tareas, donde se generan

artefactos de entrada y salida, y con la participación de distintos roles) que se integran sobre el proceso de desarrollo de LPS existente en cualquier organización, proporcionándole un enfoque en ingeniería de requisitos de seguridad específico para LPS. Los sub-procesos y actividades descritos en este artículo se pueden combinar con los procesos de desarrollo como el Proceso Unificado u otros. En este artículo describiremos la integración de SREPPLine en el marco de trabajo de ingeniería de LPS propuesto por Pohl et al. en [24].

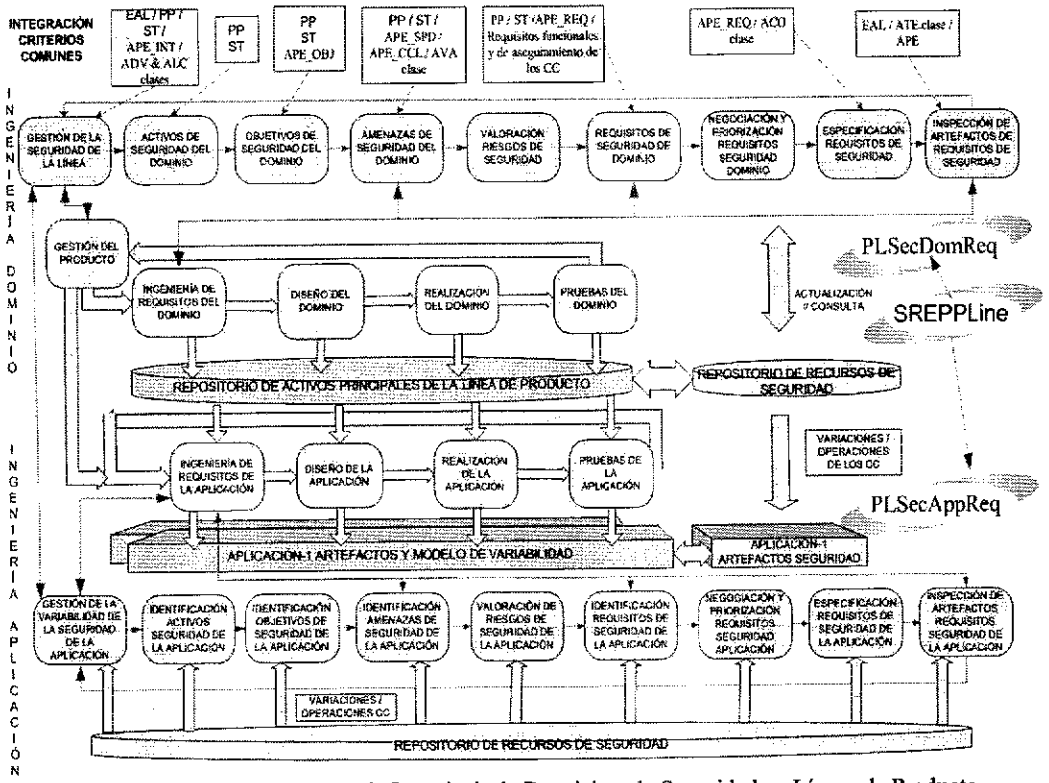

Fig. 1 Marco de Trabajo para la Ingeniería de Requisitos de Seguridad en Líneas de Producto

SREPPLine es un proceso basado en activos y dirigido por el riesgo para el establecimiento de requisitos de seguridad en el desarrollo de LPS seguras. Básicamente este proceso describe cómo integrar los Criterios Comunes (CC) en el desarrollo de LPS junto con el uso de un repositorio de recursos de seguridad para facilitar la variabilidad y reutilización de requisitos, activos, amenazas, test y contramedidas en la LPS. Asimismo, facilita la gestión del modelo de variabilidad de la seguridad y los distintos tipos de trazabilidad implicados entre los artefactos de seguridad (del producto y de la línea). Y que la LPS y los sistemas de información de la misma sean conformes a los estándares de seguridad actualmente más relevantes relativos a la gestión de requisitos de seguridad (como ISO/IEC 15408, ISO/IEC 27001, ISO/IEC 17799 o ISO/IEC 21827).

Como se puede observar en la Fig. 1, SREPPLine se compone de dos sub-procesos con sus respectivas actividades: PLSecDomReq (Product Line Security Domain Requirements Engineering sub-process) y PLSecAppReq (Product Line Security Application Requirements Engineering sub-process). Estos sub-procesos cubren las cuatro fases básicas de la ingeniería de requisitos [16]: elicitación de requisitos; análisis y negociación de requisitos; documentación de requisitos; y validación y verificación de requisitos. Estos sub-procesos se ejecutarán al menos, por cada iteración del proceso de

ingeniería del dominio y/o de la aplicación de la LPS, respectivamente. Sin embargo, dadas las restricciones de espacio, se describirán de forma general las actividades de cada uno de los dos sub-procesos: PLSecDomReq y PLSecAppReq.

Además, como se observa en la Fig. 1, el Repositorio de Recursos de Seguridad se debe de integrar en el repositorio de activos comunes de la LPS, para posibilitar las relaciones de trazabilidad entre el modelo de variabilidad de la LPS y los diferentes tipos de artefactos de seguridad y otros artefactos de desarrollo, así como la trazabalidad entre los artefactos de la línea y los productos. El modelo de variabilidad de seguridad implementado por SREPPLine se apoya en el concepto de modelo de variabilidad ortogonal [24], lo cual nos permite flexibilidad para aplicarlo, ya que permite que el proceso se integre con otros modelos de desarrollo software (como modelos de características o 'features', modelos de casos de uso, modelos de diseño, modelos de componentes o de pruebas).

3.1. PLSecDomReq: Sub-proceso de Ingeniería de Requisitos del Dominio de la Línea de Producto Software

Las principales metas de este sub-proceso (SP1) son: identificar y desarrollar los requisitos de seguridad junto con sus artefactos de seguridad asociados comunes y variables del dominio; y garantizar que sean conformes al estándar IEEE 830:1998 y facilitar su conformidad con los Criterios Comunes, generando la documentación precisa para generar un Perfil de Protección ("Protection Profile" de los CC) junto con los artefactos de seguridad asociados a los requisitos siguiendo los CC. En la Tabla 1 se resumen las actividades de PLSecDomReq.

Tabla 1 Actividades de PLSecDomReq

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.1: Gestión de la Seguridad de la Línea**
Tareas	
T1.1.1 – Mejora del repositorio de recursos de seguridad (actualización de artefactos y relaciones) T1.1.2 – Identificación de las partes interesadas específicas T1.1.3 – Acuerdo en las definiciones de seguridad T1.1.4 – Identificación de las 'features' de seguridad (variabilidad y elementos comunes): identificación del entorno de seguridad (política de seguridad, estándares de seguridad, legislación, restricciones, necesidades de seguridad en el dominio, criterio de aceptación de la seguridad y nivel de evaluación del aseguramiento (EAL: Evaluation Assurance Level) de los CC); identificación de los tipos de activos relevantes y de los objetivos de seguridad. T1.1.5 – Impacto del coste de seguridad y estimación del riesgo a alto nivel	
Artefactos de Entrada	
• Peticiones para modificar o agregar artefactos de seguridad (de A1.9 o de SP2-A2.1 "Gestión de la Variabilidad de la Seguridad de la Aplicación") • Informe de validación de los artefactos de seguridad (de A1.9) • Informes de defectos y errores (de fase Gestión del Producto) • Dominio del negocio (de fase Gestión del Producto) • Hoja de ruta y "Features" de la línea (de fase Gestión del Producto)	
Artefactos de Salida	
• Hoja de ruta de seguridad: 'features' de seguridad comunes y variables y su calendario, junto con la introducción del Perfil de Protección (APE_INT CC class) • Lista de Perfiles de Protección relacionados existentes, Declaraciones de Seguridad, Paquetes de Requisitos de Seguridad susceptibles de ser reutilizados para la LPS, así como los activos y objetivos de seguridad relacionados	
Técnicas, prácticas y guías de referencia	
• Entrevistas y reuniones. • UML (casos de uso del negocio, diagramas de actividad o gráficos del proceso o flujogramas, y flujos de trabajo de los procesos de negocio como BPMN, etc...) • Análisis de coste/beneficio y planificación de proyecto. • Repositorio de recursos de seguridad • Criterios Comunes: APE_INT, ADV, ALC.	
Roles	
• Gerente de línea • Experto en el dominio del negocio • Analista de Seguridad • Cliente y Usuarios expertos	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.2: Activos de Seguridad del Dominio**
Tareas	
T1.2.1 – Identificación de los activos o grupos de activos de la LPS y del entorno T1.2.2- Determinación las asunciones de seguridad	

T1.2.3 – Determinación del alcance de los activos de seguridad, identificando los componentes particulares a desarrollar para ser reutilizados, así como los activos comunes y los variables (puntos de variación). T1.2.4 – Identificación de las dependencias entre los activos de seguridad (restricciones de la variabilidad de la seguridad de la LPS) y y trazabilidad entre activos seguridad - activos del negocio T1.2.5 – Valoración de los activos de seguridad
Artefactos de Entrada
• Artefactos de salida de A1.1 • Procesos de negocio • Inventarios de la organización y planes de contingencia existentes, manuales y actas de reuniones/entrevistas (ADV_FSP.3.1D de los CC)
Artefactos de Salida
• Lista de activos valorados y sus dependencias • Modelo de variabilidad de los elementos de seguridad
Técnicas, prácticas y guías de referencia
• Entrevistas y reuniones. • UML (casos de uso del negocio, diagramas de actividad o gráficos del proceso o flujogramas, y flujos de trabajo de los procesos de negocio como BPMN, etc...) • Valoración Delphi • Repositorios de recursos de seguridad • Criterios Comunes: ADV_FSP.3.1D, Protection Profile, Security Target.
Roles
• Experto en el dominio del negocio • Analista de seguridad • Arquitecto de la línea • Ingeniero de requisitos

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.3: Objetivos de Seguridad del Dominio**
Tareas	
T1.3.1 – Identificación de los objetivos de seguridad (análisis de variabilidad y similitudes) y trazabilidad entre activos de seguridad - objetivos de seguridad – objetivos del negocio T1.3.2 – Modelado de los objetivos de seguridad y especificación de los mismos. T1.3.3- Valoración de los objetivos de seguridad.	
Artefactos de Entrada	
• Artefactos de salida de A1.2 y A1.1 • Metas del negocio • Hoja de ruta de la seguridad	
Artefactos de Salida	
• Árbol de objetivos de seguridad y valorados cada uno de ellos y relacionados con los activos • Modelo de variabilidad de los objetivos de seguridad • Fundamentación de los objetivos de seguridad	
Técnicas, prácticas y guías de referencia	
• Reuniones y entrevistas • Valoración Delphi • Repositorios de recursos de seguridad • Criterios Comunes: APE_OBJ, Protection Profile.	
Roles	
• Experto en el dominio del negocio • Analista de seguridad • Gerente de la línea • Ingeniero de requisitos	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.4: Amenazas de Seguridad del Dominio**
Tareas	
T1.4.1 – Identificación y análisis de las vulnerabilidades potenciales de la línea buscando en fuentes públicas o propias de la organización, para las tecnologías y sistemas en las que se basa o de las encontradas en anteriores iteraciones. T1.4.2 – Identificación del árbol de amenazas asociado al patrón de negocio de la LPS, e identificación de tipos de atacantes y tipos de ataques. T1.4.3 – Identificación de los casos de mal uso y amenazas para cada objetivo de seguridad y activo (análisis de variabilidad y similitudes de las amenazas de la línea). T1.4.4 – Modelado de las amenazas y especificación de éstas y relaciones de trazabilidad activos – amenazas - objetivos T1.4.5 – Validación de los objetivos de seguridad contra las amenazas y los activos en el modelo de variabilidad de la seguridad existente.	
Artefactos de Entrada	
• Artefactos de salida de A1.3 y A1.1 • Casos de uso del negocio y procesos del negocio (de la fase Ingeniería de Requisitos del Dominio) • Lista de vulnerabilidades comunes de los sistemas que conforman la LPS y lista pública de vulnerabilidades de la tecnología base utilizada • Informes previos de vulnerabilidades, defectos y errores (en caso de no ser la primera iteración)	

• Informes de ataques y brechas de seguridad en la organización
Artefactos de Salida
• Lista de amenazas modeladas y el modelo de variabilidad de las amenazas de seguridad y las relaciones de trazabilidad con los objetivos y activos. • Definición del problema de seguridad. • Suposiciones y afirmaciones de conformidad.
Técnicas, prácticas y guías de referencia
• UML (casos de uso del negocio, diagramas de actividad o gráficos del proceso o flujogramas, y flujos de trabajo de los procesos de negocio como BPMN, etc…) • Patrones de ataque y árboles de amenazas • Casos de mal uso y plantillas asociadas • Reuniones y valoración Delphi • Repositorios de recursos de seguridad • Criterios Comunes: AVA-VAN.5.2E, APE_CCL, APE_SPD, Protection Profile
Roles
• Experto en el dominio del negocio • Analista de seguridad • Arquitecto de la línea • Ingeniero de requisitos

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.5: Valoración de Riesgos de Seguridad**
Tareas	
T1.5.1 – Valoración de las amenazas y determinación de su relevancia frente al nivel de seguridad necesario o acordado en A1.1 y que permita el cumplimiento de los objetivos de seguridad identificados. T1.5.2 – Valoración de las contramedidas, salvaguardas y elementos de seguridad existentes. T1.5.3 – Estimación del riesgo de las amenazas relevantes según su potencialidad de ocurrencia e impacto negativo	
Artefactos de Entrada	
• Artefactos de salida de A1.4 y A1.1 • Inventario de salvaguardas y contramedidas existentes en la organización y arquitecturas y elementos de seguridad existentes en los componentes de la LPS. • Planes de contingencia	
Artefactos de Salida	
• Mapa de riesgos (riesgos potenciales y residuales)	
Técnicas, prácticas y guías de referencia	
• Análisis algorítmico o análisis mediante tablas (según MAGERIT v.2) • Análisis de costes de las salvaguardas y de los impactos de las amenazas. • Entrevistas, reuniones y valoración Delphi. • ISO/IEC 13335	
Roles	
• Analista de seguridad	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.6: Requisitos de Seguridad del Dominio**
Tareas	
T1.6.1 – Elicitación de los requisitos de seguridad del dominio, identificando los requisitos de seguridad adecuados que mitiguen las amenazas según el nivel de riesgo. T1.6.2 – Identificación de los requisitos comunes según los requisitos elicitados y el análisis de riesgos anterior T1.6.3 – Definición de los requisitos variables y análisis de las dependencias de variabilidad y determinación de las operaciones de los CC (iteración, asignación, selección o refinamiento) T1.6.4 – Establecimiento de las relaciones de trazabilidad de los requisitos con los activos, amenazas, objetivos	
Artefactos de Entrada	
• Artefactos de salida de A1.1, A1.2, A1.3, A1.4 y A1.5 • Defectos de los requisitos de seguridad encontrados en iteraciones anteriores (de A1.9) • Solicitudes para detallar o revisar requisitos de seguridad (de la fase de Diseño del Dominio) • Requisitos funcionales y no- funcionales identificados hasta el momento de la iteración (de la fase Ingeniería de Requisitos del Dominio)	
Artefactos de Salida	
• Listado de requisitos funcionales y de aseguramiento (conformes a IEEE 830:1998) • Modelo de variabilidad de los requisitos de seguridad y las relaciones de trazabilidad con los objetivos, amenazas y activos y los riesgos asociados a cada requisito.	
Técnicas, prácticas y guías de referencia	
• UML (casos de uso del negocio, diagramas de actividad o gráficos del proceso o flujogramas, y flujos de trabajo de los procesos de negocio como BPMN, etc…) • Repositorios de recursos de seguridad • Casos de uso de seguridad • Criterios Comunes: APE_REQ, requisitos funcionales de seguridad y aseguramiento de los CC, Protection Profile y Security Target	
Roles	
• Ingeniero de requisitos • Experto en el dominio del negocio • Analista de seguridad	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.7: Priorización y Negociación de Requisitos de Seguridad**
Tareas	
T1.7.1 – Identificación y especificación de las dependencias entre los requisitos de seguridad y los requisitos funcionales y no-funcionales de la LPS T1.7.2 – Priorización de los requisitos de seguridad en función del riesgo T1.7.3 – Valoración del impacto económico de las medidas de salvaguarda a implementar y en el resto de requisitos frente al riesgo T1.7.4 – Negociación y acuerdos para la priorización de los requisitos de seguridad y el resto de requisitos y por tanto en el modelo de variabilidad de la seguridad y en el modelo de variabilidad de la LPS	
Artefactos de Entrada	
• Artefactos de salida de A1.6 y A1.5 • Requisitos funcionales y no- funcionales identificados hasta el momento de la iteración (de la fase Ingeniería de Requisitos del Dominio)	
Artefactos de Salida	
• Lista de requisitos de seguridad priorizados y valorados, junto con sus relaciones de dependencia y variabilidad	
Técnicas, prácticas y guías de referencia	
• Asignación numérica ("Grouping") [2]. • Win-Win • Entrevistas, Reuniones y Valoración Delphi • Análisis coste-beneficio	
Roles	
• Ingeniero de requisitos • Analista de Seguridad • Experto en el dominio del negocio • Arquitecto de la Línea • Cliente	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.8: Especificación de Requisitos de Seguridad**
Tareas	
T1.8.1 – Modelado de los requisitos de seguridad T1.8.2 – Especificación de los requisitos de seguridad y sus artefactos T1.8.3 - Descripción de las contramedidas / medidas de salvaguarda y de las pruebas de seguridad y establecimiento de las métricas de seguridad para cuantificar el cumplimiento de los requisitos T1.8.4 - Establecimiento de las relaciones de trazabilidad de los requisitos con las pruebas y contramedidas	
Artefactos de Entrada	
• Artefactos de salida de A1.7, A1.3 y A1.4	
Artefactos de Salida	
• Especificación de los requisitos de seguridad y sus artefactos • Descripción de las contramedidas / medidas de salvaguarda • Descripción de las pruebas de seguridad y establecimiento de las métricas de seguridad para cuantificar el cumplimiento de los requisitos • Fundamentación de los requisitos de seguridad • Perfil de Protección de la LPS (recoge la información fundamental de la iteración) y su modelo de variabilidad de la seguridad	
Técnicas, prácticas y guías de referencia	
• Plantillas de Casos de uso de seguridad • UMLSec • Repositorios de recursos de seguridad • Entrevistas y reuniones • Criterios Comunes: APE_REQ y clase ACO, Protection Profile	
Roles	
• Ingeniero de requisitos • Analista de Seguridad	

Sub-proceso	Actividad
SP1: PLSecDomReq	**A1.9: Inspección de Requisitos de Seguridad**
Tareas	
T1.9.1 – Verificación de la conformidad del Perfil de Protección y de los artefactos generados con los CC T1.9.2 – Verificación de la conformidad de los requisitos de seguridad con IEEE 830:1998 T1.9.3 – Verificación del grado de cumplimiento de los requisitos de seguridad aplicando las pruebas de seguridad definidas y según las métricas establecidas. T1.9.4 – Comprobación del nivel de madurez de la seguridad del proceso según ISO/IEC 21827 (SSE-CMM) aplicando CC_SSE-CMM [17]. T1.9.5 – Verificación en la fase de Pruebas del Dominio la conformidad con el EAL y los CC de aseguramiento	
Artefactos de Entrada	

• Artefactos de salida de las actividades anteriores
Artefactos de Salida
• Informe de validación con los requisitos de seguridad y sus artefactos validados o rechazados • Informe de vulnerabilidades, defectos y errores • Solicitudes de cambio o incorporación de artefactos de seguridad
Técnicas, prácticas y guías de referencia
• Lista de verificación y registro de revisión [19] • Repositorios de recursos de seguridad • Reuniones y walkthroughs • Criterios Comunes: EAL, APE y clase ATE, Protection Profile
Roles
• Analista de Seguridad • Asegurador de la Calidad de la Línea • Equipo de Inspección y Auditoría • Cliente y Usuarios expertos

3.2. PLSecAppReq: Sub-proceso de Ingeniería de Requisitos de la Aplicación de la Línea de Producto Software

Las principales metas del sub-proceso (SP2) PLSecAppReq (Product Line Security Application Requirements Engineering sub-process) son: elicitar y documentar los requisitos de seguridad del producto y sus artefactos relacionados; garantizar la conformidad de los requisitos con el estándar IEEE 830:1998 así como con los CC junto el resto de artefactos; generar el documento de Declaración de Seguridad; y reutilizar lo máximo posible los requisitos y demás artefactos de seguridad comunes de la línea.

Las actividades de PLSecAppReq que se reflejan en la Fig.1 son similares a las actividades de SREP [21], salvo las tareas específicas y actividades únicas de ingeniería de requisitos de seguridad propias del desarrollo del producto basado en una LPS. Por tanto y dadas las restricciones de espacio, a continuación resumiremos las principales diferencias entre PLSecAppReq y SREP que se aplica para el desarrollo de sistemas individuales:

- La elicitación de requisitos se basa en la comunicación de la parte común y variable de la LPS. De manera que la mayoría de los requisitos no se elicitan de nuevo, sino que se derivan de los requisitos del dominio (de la LPS) y almacenados en el repositorio.
- Durante la elicitación de requisitos se deben detectar las diferencias entre los artefactos de seguridad de la aplicación y del dominio (sec-deltas) y considerar y evaluar el esfuerzo de adaptación requerido en su caso, así como documentarlo adecuadamente. Ya que si el esfuerzo de adaptación requerido se detecta en fases tempranas, es posible tomar decisiones y concesiones respecto a los artefactos de seguridad de la aplicación, no solo para reducir el esfuerzo sino también para incrementar la cantidad de artefactos del dominio y facilitar la reutilización futura.

4. SREPPLineTool

El prototipo que se presenta es una primera aproximación que servirá para obtener experiencia del problema mediante su aplicación en escenarios de uso y casos de estudio, para así refinarlo y obtener una versión definitiva de SREPPLineTool. Asimismo, por restricciones de espacio, únicamente se describirán las características principales de la herramienta.

SREPPLineTool ha sido rediseñada y desarrollada a partir de SREPTOOL [22], con el fin de dar soporte automatizado a la aplicación de SREPPLine. SREPPLineTool proporciona una forma guiada, sistemática e intuitiva para la aplicación específica para LPS del proceso de ingeniería de requisitos de seguridad, aplicando SREPPLine, asimismo posibilita una sencilla integración con los demás requisitos y con el modelo de variabilidad de las LPS y con las distintas fases del ciclo de desarrollo basado en LPS, así como facilita la aplicación sencilla de los CC y el cumplimiento del estándar IEEE 830:1998, ayudándose para ello de las funcionalidades que ofrece '*IBM Rational RequisitePro*' (herramienta CARE que extiende SREPPLineTool). Además, este prototipo ayuda en que las LPS y sus productos desarrollados sean conformes a los estándares de seguridad más importantes en lo relativo a la gestión de requisitos de seguridad, sin la necesidad de dominar dichos estándares y reduciendo la participación de expertos de seguridad para conseguirlo, es decir, mejora la eficiencia de SREPPLine. Y adicionalmente, gracias al Repositorio de Recursos de Seguridad que integra SREPPLineTool, se facilita la el modelo de variabilidad de la seguridad y por tanto la reutilización de artefactos, mejorándose por ende la calidad sucesivamente. En la Fig.2 se muestran las interfaces de SREPPLineTool de dos de las actividades del sub-proceso PLSecDomReq.

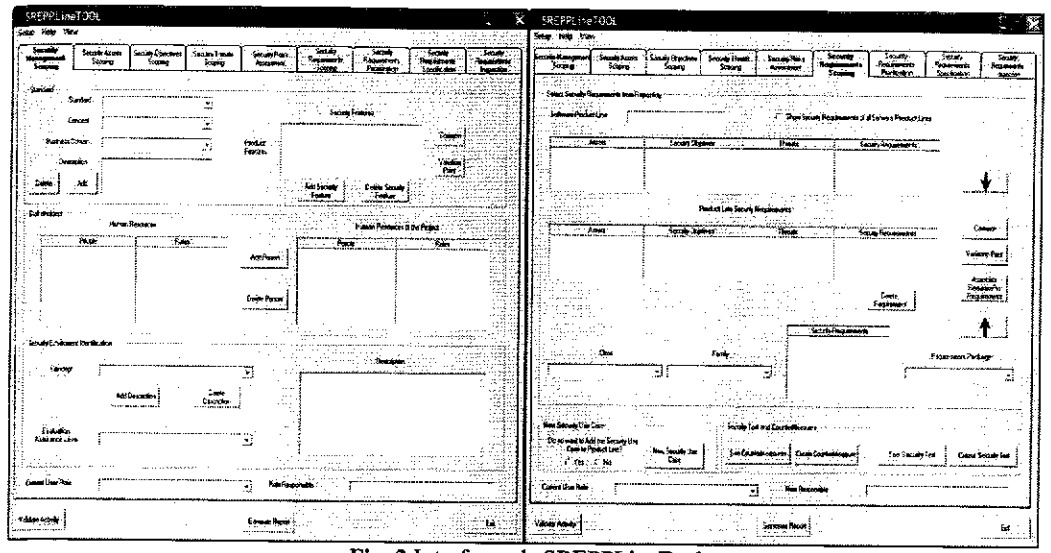

Fig. 2 Interfaces de SREPPLineTool

Para la creación del prototipo se ha utilizado el lenguaje de programación Visual Basic 6, produciendo como artefacto de salida una biblioteca *dll ActiveX*, que será enlazada con IBM/Rational RequisitePro. De esta manera, los objetos de RequisitePro serán visibles desde SREPPLineTool y viceversa, facilitándose las relaciones de trazabilidad necesarias para la aplicación de SREPPLine y su adecuado modelo de variabilidad y reutilización con el repositorio.

Así, la funcionalidad del prototipo estaría accesible desde la ventana principal de RequisitePro, a través del menú *Tools→SREPPLineTool*. Para la integración con RequisitePro, SREPPLineTool se ha desarrollado como un add-in de dicha herramienta CARE, para lo cual se ha utilizado la interfaz de extensibilidad de RequisitePro, en concreto el *RequisitePro Extensibility Server* (RPX) que permite acceder a los datos almacenados en RequisitePro y el *RqGUIApp library* que controla la interfaz de usuario del RequisitePro y permite también controlar los documentos de Microsoft Word. De forma que mediante plantillas Word, SREPPLineTool genera sus informes y los integra en RequisitePro. Además, utilizará una base de datos por producto con los artefactos de seguridad propios o específicos de dicho producto y una base de datos por cada LPS para el repositorio, con los artefactos de la línea (para este prototipo se ha utilizado MS-Access).

5. Conclusiones y Trabajos Futuros

Hoy en día, debido a la creciente necesidad de obtener SI de alta calidad y con una productividad alta, el desarrollo basado en LPS se convertido en el enfoque de más éxito para asegurar la calidad, eficiencia económica y mantenibilidad de los SI [3]. Es por ello, y dada la complejidad y a la naturaleza extensiva de las LPS [12], que sea fundamental la incorporación de la seguridad en las líneas de producto software, siendo mucho más importantes para la puesta en práctica del desarrollo basado en LPS, de lo que ya son para el desarrollo de un Sistema de Información (SI).

Debido a que los trabajos existentes que apuntan a especificar seguridad en líneas de producto, en los que se integre la perspectiva de la ingeniería de requisitos de seguridad, son escasos y no proporcionan soporte metodológico y sistemático para la gestión de la seguridad en LPS basada en la ingeniería de requisitos de seguridad y en los estándares de seguridad internacionales más importantes. En este artículo se presenta un

proceso sistemático soportado por una herramienta que juntos ayudan a desarrollar líneas de producto software seguras mediante la gestión integral de los requisitos de seguridad desde las primeras fases del ciclo de desarrollo y apoyándose en los estándares de seguridad internacionales más importantes (como ISO/IEC 15408 e ISO/IEC 21827; ISO/IEC 17799:2005, secciones: 0.3, 0.4, 0.6 y 12.1; ISO/IEC 27001, secciones: 4.2.1, 4.2.3, 4.3, 6.a y A.12.1.1), con el objeto de aportar una perspectiva que permita mejorar la calidad, tanto en las líneas de productos software como en los productos de dicha línea, los cuales serán conformes a dichos estándares.

Por último, hay una serie de aspectos planeados para el futuro del prototipo presentado anteriormente (SREPPLineTool) y que nos permitirá incrementar el nivel de automatización de la aplicación de SREPPLine y mejorar así la eficiencia del proceso de ingeniería de requisitos de seguridad de la LPS. Entre otros, son de destacar los siguientes: refinar la herramienta para automatizar la aplicación del sub-proceso PLSecAppReq; refinar la integración con IBM/Rational RequisitePro, ya que no es una herramienta especializada en gestión de requisitos de LPS; extender SREPPLineTool para poder ser acoplada en otras herramientas CARE o herramientas de LPS; refinar la funcionalidad de la herramienta probándola en escenarios y casos prácticos; mejorar la integración del modelo de variabilidad de la seguridad y su modelo de decisión con los de la LPS; soportar UMLSec [11]. Asimismo, se refinará el modelo teórico junto con la herramienta al ir aplicándolo en casos de estudio.

Agradecimientos

Este artículo es parte del proyecto ESFINGE (TIN2006-15175-C05-05) y RETISTRUST (TIN2006-26885-E) del Ministerio de Educación y Ciencia, y de los proyectos MISTICO (PBC-06-0082) 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

[1] J. L. Arciniegas, J. C. Dueñas, J. L. Ruiz, R. Cerón, J. Bermejo, and M. A. Oltra, "Architecture Reasoning for Supporting Product Line Evolution: An Example on Security," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[2] A. Aurum and C. Wohlin, "Requirements Engineering: Setting the Context," in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds., 2005, pp. 1-15.

[3] A. Birk, G. Heller, I. John, T. v. d. MaBen, K. Müller, and K. Schmid, "Product line engineering industrial nuts and bolts," Fraunhofer IESE, Kaiserslautern November 2003 2003.

[4] J. Bosh, *Design & Use of Software Architectures*: Pearson Education Limited, 2000.

[5] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*: Addison-Wesley, 2002.

[6] A. Davis, "Tracing: A Simple Necessity Neglected," in *IEEE Software*, vol. 12, 1995.

[7] T. E. Faegri and S. Hallsteinsen, "A Software Product Line Reference Architecture for Security," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[8] D. G. Firesmith, "Security Use Cases," *Journal of Object Technology*, pp. 53-64, 2003.

[9] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "ST-Tool: A CASE Tool for Security Requirements Engineering," presented at IEEE International Conference on Requirements Engineering (RE'05), 2005.

[10] A. Immonen, "A Method for Predicting Reliability and Availability at the Architecture Level," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[11] J. Jürjens, "UMLsec: extending UML for secure systems development.," *UML 2002 - The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference.*, vol. LNCS 2460, pp. 412-425, 2002.

[12] T. Käkölä and J. C. Dueñas, *Software Product Lines: Research Issues in Engineering and Management*: Springer, 2006.

[13] K. Kang, S. Cohen, J. A. Hess, W. E. Novak, and S. A. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Software Engineering Institute, Carnegie-Mellon University, 1990.

[14] J. Kim, M. Kim, and S. Park, "Goal and scenario bases domain requirements analysis environment," in *The Journal of Systems and Software*, vol. 79, 2005, pp. 926 - 938.

[15] H.-K. Kim., "Automatic Translation Form Requirements Model into Use Cases Modeling on UML," *ICCSA 2005, LNCS*, pp. 769-777, 2005.

[16] G. Kotonya and I. Sommerville, *Requirements Engineering Process and Techniques*, Hardcover ed. UK: John Willey & Sons, 1998.

[17] J. Lee, J. Lee, S. Lee, and B. Choi, "A CC-based Security Engineering Process Evaluation Model," *27th Annual International Computer Software and Applications Conference (COMPSAC'03)*, pp. 130-, 2003.

[18] J. McDermott and C. Fox, "Using Abuse Case Models for Security Requirements Analysis," presented at Annual Computer Security Applications Conference, Phoenix, Arizona, 1999.

[19] N. R. Mead and T. Stehney, "Security Quality Requirements Engineering (SQUARE) Methodology," presented at Software Engineering for Secure Systems (SESS05), ICSE 2005 International Workshop on Requirements for High Assurance Systems, St. Louis, 2005.

[20] D. Mellado, E. Fernández-Medina, and M. Piattini, "A Comparative Study of Proposals for Establishing Security Requirements for the Development of Secure Information Systems," *The 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Springer LNCS 3982*, vol. 3, pp. 1044-1053, 2006.

[21] D. Mellado, E. Fernández-Medina, and M. Piattini, "A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems," *Computer Standards and Interfaces*, vol. 29, pp. 244 - 253, 2007.

[22] D. Mellado, M. Rodríguez, E. Fernández-Medina, and M. Piattini, "Soporte Automatizado a la Ingeniería de Requisitos de Seguridad," *X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software (IDEAS'07)*, pp. (accepted), 2007.

[23] H. Mouratidis and P. Giorgini, *Integrating Security and Software Engineering: Advances and Future Visions*: Idea Group Publishing, 2007.

[24] K. Pohl, G. Böckle, and F. v. d. Linden, *Software Product Line Engineering. Foundations, Principles and Techniques*. Berlin Heidelberg: Springer, 2005.

[25] G. Popp, J. Jürjens, G. Wimmel, and R. Breu, "Security-Critical System Development with Extended Use Cases." 10th Asia-Pacific Software Engineering Conference, 2003, pp. 478-487.

[26] K. Schmid, K. Krennrich, and M. Eisenbarth, "Requirements Management for Product Lines: A Prototype," Fraunhofer IESE July 2005 2005.

[27] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering 10*, vol. 1, pp. 34-44, 2005.

[28] A. Toval, J. Nicolás, B. Moros, and F. García, "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," in *Requirements Engineering*, vol. 6, 2002, pp. 205-219.