The Third International Conference on Software Engineering Advances

ICSEA 2008

ENTISY 2008: International Workshop on Enterprise Information Systems

26-31 October 2008

Sliema, Malta

Editors

Herwig Mannaert
Tadashi Ohta

Cosmin Dini
Robert Pellerin

Sponsored by

IARIA

Published by

IEEE computer society

CPS
Conference Publishing Services

IEEE Computer Society Order Number E3372
BMS Part Number CFP0891B-CDR
ISBN 978-0-7695-3372-8
Library of Congress Number 2008930370

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Patrick Kellenberger
Cover art production by Patrick Kellenberger

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

Proceedings

The Third International Conference on Software Engineering Advances

ICSEA 2008

Includes
ENTISY 2008: International Workshop on Enterprise Information Systems

26-31 October 2008
Sliema, Malta

Editors
Herwig Mannaert
Tadashi Ohta
Cosmin Dini
Robert Pellerin

CPS Conference Publishing Services

Los Alamitos, California

Washington • Tokyo

IEEE computer society

Preface
ICSEA 2008

The Third International Conference on Software Engineering Advances (ICSEA 2008), held between October 26 and October 31, 2008 in Sliema, Malta, is a multi-track event covering related topics on designing, implementing, and testing software.

The conference covers fundamentals on designing, implementing, testing, validating and maintaining various kinds of software. The tracks treat the topics from theory to practice, in terms of methodologies, design, implementation, testing, use cases, tools, and lessons learnt. The conference topics cover classical and advanced methodologies, open source, agile software, as well as software deployment and software economics and education.

The conference had the following tracks:

- Advances in fundamentals for software development
- Advanced mechanisms for software development
- Advanced design tools for developing software
- Advanced facilities for accessing software
- Software performance
- Software security, privacy, safeness
- Advances in software testing
- Specialized software advanced applications
- Open source software
- Agile software techniques
- Software deployment and maintenance
- Software economics, adoption, and education
- Improving research productivity

ICSEA 2008 also included:

- ENTISY 2008: International Workshop on Enterprise Information Systems

Similar to the previous edition, this event continued to be very competitive in its selection process and very well perceived by the international software engineering community. As such, it is attracting excellent contributions and active participation from all over the world. We were very pleased to receive a large amount of top quality contributions.

We take here the opportunity to warmly thank all the members of the ICSEA 2008 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the ICSEA 2008. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ICSEA 2008 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success.

We hope the ICSEA 2008 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in networking research.

We hope Malta provided a pleasant environment during the conference and everyone saved some time for exploring this historic island.

ICSEA 2008 Chairs

Herwig Mannaert, Universiteit Antwerp, Begium
Tadashi Ohta, Soka University, Tokyo, Japan
Cosmin Dini, Université de Franche-Comté, France
Robert Pellerin, Ecole Polytechnique Montreal, Canada

The Third International Conference on Software Engineering Advances

ICSEA 2008

Table of Contents

ICSEA 3: Mechanisms I

ICSEA 4: Mechanisms II

ICSEA 5: Open Source

ICSEA 6: Deployment and Maintenance I

ICSEA 7: Deployment and Maintenance II

ICSEA 8: Software Testing

ICSEA 9: Agile Software Techniques

ICSEA 10: Software Economics, Adoption, & Education

ICSEA 11: Advances in Fundamentals I

ICSEA 12: Advances in Fundamentals II

ICSEA 13: Advances in Fundamentals III

ICSEA 14: Advances in Fundamentals IV

ICSEA 15: Advances in Fundamentals V

ICSEA 16: ENTISY

Security Requirements Engineering Process for Software Product Lines: A Case Study

Daniel Mellado
National Competition Commission, IT Department; Madrid, Spain
Daniel.Mellado@alu.uclm.es

Eduardo Fernández-Medina
University of Castilla La-Mancha, Alarcos Research Group, Information Systems and Technologies Department; Spain.
Eduardo.FdezMedina@uclm.es

Mario Piattini
University of Castilla La-Mancha, Alarcos Research Group, Information Systems and Technologies Department; Spain.
Mario.Piattini@uclm.es

Abstract

The majority of the current product line practices in requirements engineering do not adequately address security requirements engineering despite the fact that security requirements engineering is both a central task and a critical success factor in product line development due to the complexity and extensive nature of product lines. Therefore, our contribution is to present and to demonstrate the applicability of our proposed security quality requirements engineering process (SREPPLine), which is based on a security requirements decision model driven by security standards along with a security variability model. We shall demonstrate our proposal by describing part of a real case study as a preliminary validation of these models. The final aim of this approach is to deal with security requirements variability from the early stages of the product line development in a systematic way, in order to facilitate conformance of the products with the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408.

1. Introduction

In the search for improved software quality and high productivity, software product line (SPL) engineering has proven to be one of the most successful paradigms for developing a diversity of similar software applications and software-intensive systems at low costs, in a short time, and with high quality, by exploiting commonalities and variabilities among products to achieve high levels of reuse [2, 3].

In software intensive systems, such as SPL, security is a cross-cutting concern and should consequently be subject to careful requirements analysis and decision making. Moreover, in SPL engineering, security is one of the most important attributes with regard to quality, given that a weakness in security may cause problems in all the products in a product line. In addition, many requirements engineering practices must be appropriately tailored to the specific demands of product lines [1]. Hence, specifying requirements for a SPL is a challenging task [12] and specifying security quality requirements for an SPL is even more challenging due to the varying security properties required in different products.

Therefore, the discipline known as Security Requirements Engineering is essential for secure SPL and products development, because it provides techniques, methods, standards and systematic and repeatable procedures for tackling SPL security requirement issues throughout the SPL development lifecycle both to ensure the definition of security quality requirements and to manage the variability of security properties. Nevertheless, software engineering methodologies and standard proposals of SPL engineering have traditionally ignored security requirements and security variability issues. Although some of them include a few security requirements activities, most of them focus only on the design of implementation aspects of SPL development.

978-0-7695-3372-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSEA.2008.14

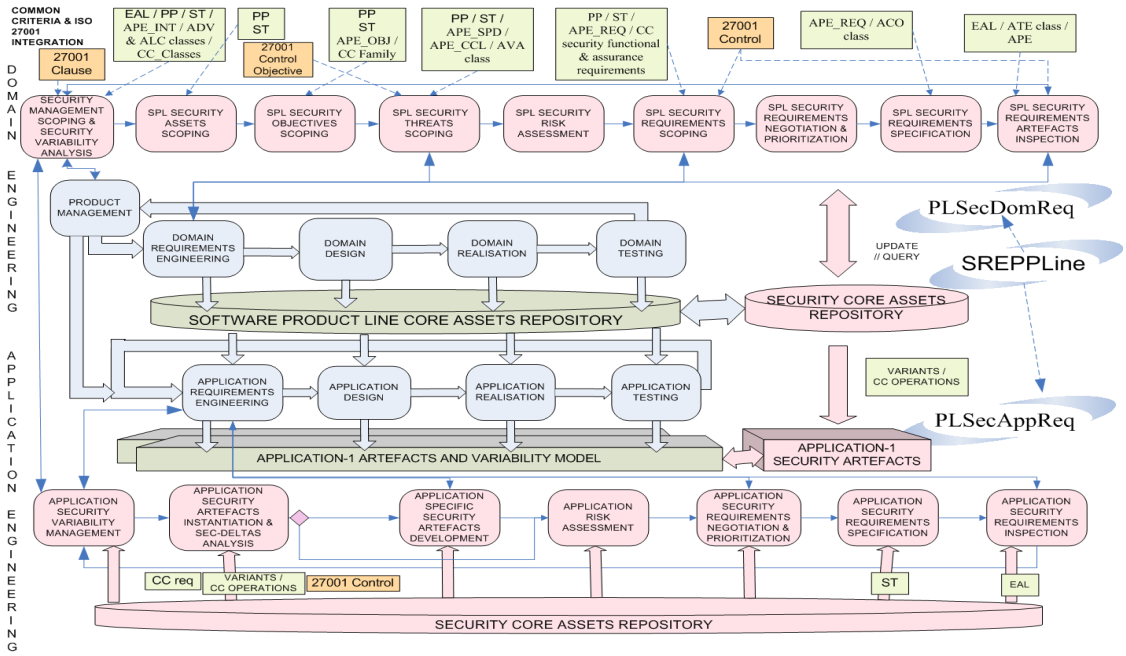

Fig. 1 Software Product Line Security Requirements Engineering Framework

As an evolution of our previous "generic" security requirements engineering process (SREP) [10], in [11] we presented the Security quality Requirements Engineering Process for Software Product Lines (SREPPLine) in [11], in which we described the most important tasks of the activities its subprocesses of it (shown in Fig. 1), along with its workflows. In this paper, we shall describe part of a real case study focusing on security requirements artefacts variability for a Public Registry Online Product Line performed at a Spanish Public Institution IT Department as a preliminary validation of the application of SREPPLine. The aim of our approach is to deal with the security requirements artefacts and their variability from the early stages of the SPL development and its products in a systematic way, in order to facilitate the conformance of SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 [7] and ISO/IEC 15408 (Common Criteria) [6]. To this end, we will propose a systematic and iterative process based on a security requirements decision model driven by security standards in order to assist in SPL products security certification along with a security variability model to manage the variability and traceability of the security requirements artefacts of the SPL and its products.

The remainder of this paper is structured as follows. In Section 2, we will outline our Security quality Requirements Engineering Process for software Product Lines (SREPPLine). In Section 3, due to space restrictions we will only describe part of a real case study of SREPPLine as a preliminary validation of it. Finally, in Section 4, we will discuss our contributions and future work.

2. SREPPLine: security quality requirements engineering process for software product lines

A software product line is a set of software-intensive systems sharing a common, managed set of features [8] which satisfy the specific needs of a particular market segment or mission and which are developed from a common set of core assets in a prescribed way [3]. The software product line engineering paradigm differentiates two processes: domain engineering and application engineering [13].

SREPPLine is an add-in of activities, which can be incorporated into an organization's SPL development process model providing it with a security requirements engineering approach.

It is a security features or security goals based process which is driven by risk and security standards (concretely ISO/IEC 27001 and Common Criteria) and deals with security requirements and their related artefacts from the early stages of SPL development in a systematic and intuitive way especially tailored to SPL based development. It is based on the use of the latest and widely validated security requirements techniques, such as security use cases [4] or misuse cases [14], along with the integration of the Common Criteria

(CC) components and ISO/IEC 27001 controls into the SPL lifecycle in order to facilitate SPL products security certification. Moreover, our proposed process suggests using a method to carry out the risk assessment which conforms to ISO/IEC 13335 [5], and concretely it uses Magerit [9] for both SPL risk assessment and SPL products risk assessment. Furthermore, SREPPLine has the aim of minimizing the necessary security standards knowledge as well as security expert participation during SPL products development. To this end, it provides a Security Core Assets Repository to facilitate security artefacts reuse and to implement the Security Variability Model and the Security Requirement Decision Model, which assist in the management of the variability and traceability of the security requirements related artefacts of the SPL and its products. These models are the basis through which the activities of SREPPLine capture, represent and share knowledge about security requirements for SPL and help to certify them against security standards. In essence, it is a knowledge repository with a structure to support security requirements reasoning in SPL.

As is described in Fig. 1 our process, which is integrated into the proposed framework for SPL engineering of Pohl et al. in [13], is composed of two subprocesses (shown in Fig.1): Product Line Security Domain Requirements Engineering (PLSecDomReq) subprocess and Product Line Security Application Requirements Engineering (PLSecAppReq) subprocess.

3. SRPEPPLine in practice

We illustrate the SREPPLine applicability in SPL engineering with the Public Registry Online Product Line of a Spanish Public Administration. This SPL may have several different configurations for different public institutions within Spanish Public Administration. It has a common set of system functionality that forms the deliverable core and a variable set of configurable parameters and non functional requirements. Therefore, this Public Registry Online Product Line is an SPL whose members vary through system configuration and online business services and yet retain the same core functionalities.

This example concentrates on the results from the PLSecAppReq (subprocess of SREPPLine) application to application engineering in order to develop a Public Registry Online in a Spanish Public Institution from the Public Registry Online Product Line and it is focused on the security features of the Public Registry Online platform. This example has had to be simplified and summed up in order to enable points of the model to be easily illustrated in this article.

The Public Registry Online Product Line provides the variability as represented by the variability model in Fig. 2. It offers different variants (V) for the different 'online requests' which are the business services offered by the Public Registry Online Product Line, which could be selected by the application stakeholder. During PLSecAppReq activity 1 ("**Application Security Variability Management**"), the Security Requirements Decision Model together with the Security Variability Model enabled the security requirements engineer to communicate the relevant security related variations points (VP), security related variants and their dependences (security artefacts, security standards and other functional and non-functional requirements) to stakeholders. Once the stakeholders informed the security requirements engineer of their security goals and of the features necessary for the application (or product), the result of this activity was a set of domain security goals and features of the SPL, which did not completely fulfill the stakeholders security goals for the application.

In this example, we selected the security features: user authenticity and secure submissions. As is shown in Fig. 2, for the variation point 'user authenticity' different authenticity methods are selectable from the Public Registry Online Product Line. It offers the security variants: 'password' and 'electronic certificate'. For the variation point 'secure submissions' three security variants are selectable: 'http', 'SSL' and 'https'.

In activity 2 of PLSecAppReq ("**Application Security Artefacts Instantiation and Sec-Deltas Analysis**") application security artefacts from the set of domain security features obtained in the previous activity were instantiated. Throughout the Security Requirements Decision Model and the Security Variability Model the appropriate security artefacts (that is, the security variants) for the specific application (product) which would as far as possible satisfy the application security goals, were selected. The result of this activity was a set of security requirements and their related artefacts, which did not completely fulfill the stakeholders' application requirements. In this example, at the VP 'secure submissions' we selected the security variant 'https' because the stakeholders selected the 'public view' variant and due to the security links (or traceability links) established on the Security Requirements Decision Model of the Public Registry Online Product Line. At the VP 'user authenticity' we selected the security variant 'e-certificate' because the stakeholders

selected the 'online requests' feature for the Public Registry Online of the Institution.

In activity 3 "**Application Specific Security Artefacts Development**" the sec-deltas analysis was performed. The sec-deltas occur when stakeholder security requirements cannot be completely satisfied by security domain requirements artefacts. During the sec-deltas analysis, sec-deltas to the security domain variability model resulting from stakeholders' security features/goals were analyzed. Due to the particular stakeholders needs for the Public Registry Online of the Institution we had to add one more variant to the 'online requests' to allow online requests of 'retirement pension'. This kind of request necessitated the attachment of documentation. Therefore we identified one sec-delta (depicted as a discontinued line in Fig. 2) because the SPL did not provide any security feature to ensure secure attachments; we therefore added one more security variation point for the 'file documentation' to the application variability model, as is shown in Fig. 2. This VP offers the variants: 'signed file' and 'pdf'. Next, the impact of the security variability model sec-deltas on the corresponding security artefacts was analyzed. The results of this analysis were the security application variability model (shown in Fig. 2) along with the security requirements artefacts deltas (assets, threats, etc.).

Finally, these sec-deltas were communicated to the security risk expert who estimated the risks of carrying out or not carrying out the security requirements deltas (activity 4 "**Application Risk Assessment**") as shown in Table 1. For example, the estimated security risk for not carrying out the security variant 'signed files' was 'high' (risk of 4 in a scale of 0 to 5). The first number of each cell in the table is the value of the assets; the second number of each cell is the degradation value of the assets caused by the threat expressed as a percentage; the third value is the accumulated impact to the assets; and the last value is the accumulated risk to the assets, according to Magerit [17] method.

In the "**Application Security Requirements Negotiation and Prioritization**" activity (activity 5 of PLSecAppReq), after the application risk assessment of the sec-deltas was performed, it was communicated to the security architect and to the security requirements engineer who estimated the realisation effort based on the sec-deltas and their associated risks. The stakeholders used this estimation to decide whether or not the security requirements deltas should be carried out and which security standard the application should fulfil. In this example we performed a slight economical analysis by balancing the risk with the economical impact of implementing countermeasures. Thereby we reached an agreement with the stakeholders about taking into account those security requirements associated with those threats that imply high or very high risk (risk of 4 or 5) whatever the conflicts with other requirements. However, for the security requirements with a risk which was lower than high (that is, from 3 to 1, medium to low) we had to reach trade-offs mainly with other non-functional requirements mainly, especially with regard to performance and interface accessibility (as is shown in Fig. 2, the system had to fulfil the WAI, Web Accessibility Initiative, level 'AA'). As a result of this activity, the application security requirements and the corresponding security requirements artefacts and security application variability model were defined.

Next, in the "**Application Security Requirements Specification**" activity (activity 6 of PLSecAppReq) the application security artefacts, the sec-deltas and the traces between application security artefacts and the corresponding domain security artefacts were formally specified and documented. Moreover, the security application variability model and the traceability links of the application security artefacts to the application-specific variability model were documented, such as the security requirement specification in XML shown in Fig. 2. The estimated risk and realisation costs were even related to the sec-deltas in order to ensure that decisions about sec-deltas were traceable.

Finally, in activity 7 ("**Application Security Requirements Inspection**") the security requirements artefacts variability consistency between the application and domain artefacts of the Public Registry Online Product Line was verified. We also verified whether the security requirements satisfied the stakeholders' security needs and application security goals, and whether the security requirements conformed to ISO/IEC 27001 control objectives, to Common Criteria assurance requirements and to the IEEE 830-1998 standard.

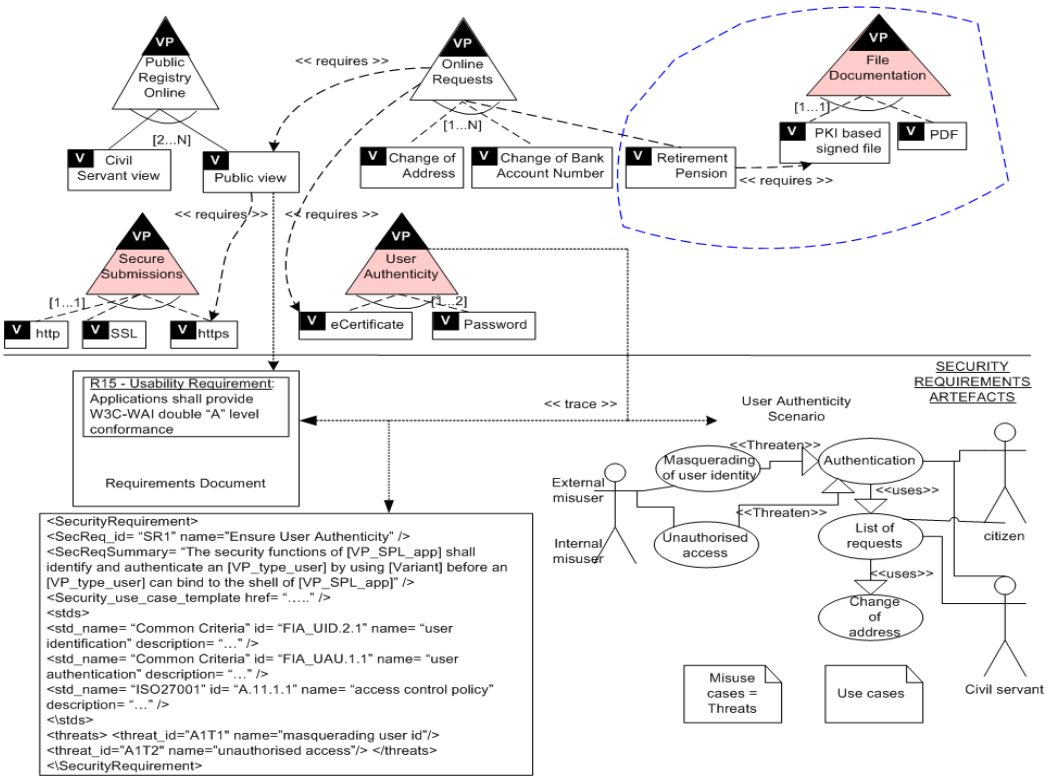

Fig. 2 Example: Public Registry Online security variability model and security artefacts

			Security Objectives / Security Dimensions						
(A) Assets (T) Threats		Frecu	[D] Availab	[I] Integrity	[C] Confide	[A_S] Authe	[A_D] Auth	[T_S] Accou	[T_D] Accou
[BS] Business Services									
(A)	[BS_Pension] Request Retirement Per		5; 70%; 5; **4**			7; 100%; 7; **5**		6; 100%; 6; **5**	
(A)	[BS_Address] Request change addres		5; 50%; 4; **4**			7; 100%; 7; **5**		6; 100%; 6; **5**	
(A)	[BS_BankNum] Request change bank		5; 50%; 4; **4**			7; 100%; 7; **5**		6; 100%; 6; **5**	
(A)	[BS_ReqManage] Requests Managem		3; 50%; 2; **3**			5; 100%; 5; **5**		5; 100%; 5; **4**	
[BD] Business Data									
(A)	[D_SS] Files Social Security		[5]; 90%; 5; **5**	5; 50%; 4; **4**	7; 100%; 7; **5**	[7]; 100%; 7; **5**	6; 100%; 6; **3**	[6]; 100%; 6; **3**	5; 100%; 5; **3**
(A)	[D_Personal] Citizen Personal Data		[5]; 90%; 5; **5**	5; 50%; 4; **4**	7; 100%; 7; **5**	[7]; 100%; 7; **5**	6; 100%; 6; **3**	[6]; 100%; 6; **3**	5; 100%; 5; **3**
(A)	[D_FileAttach] FileAttachRequest		[5]; 90%; 5; **5**	1; 50%; 0; **2**	5; 100%; 5; **5**	[7]; 100%; 7; **5**	6; 100%; 6; **3**	[6]; 100%; 6; **3**	5; 100%; 5; **3**
(A)	[D_FileAttach2] FileAttachRequestSec		[5]; 90%; 5; **5**	5; 50%; 4; **4**	7; 100%; 7; **5**	[7]; 100%; 7; **5**	6; 100%; 6; **3**	[6]; 100%; 6; **3**	5; 100%; 5; **3**
	(T) Manipulation of config	0,1	50%; 4; **2**	10%; 2; **2**	50%; 6; **2**	100%; 7; **4**	100%; 6; **3**	100%; 6; **3**	100%; 5; **3**
	(T) Masquerading of user i	100				100%; 7; **5**			
	(T) Modification of data	20		50%; 4; **5**					
	(T) Eavesdropping	10			50%; 6; **4**				
	(T) Unauthorised access	100	70%; 5; **5**	10%; 2; **3**	50%; 6; **5**	50%; 6; **5**			
[IS] Internal Services									
(A)	[IS_Auth] Login Service		[5]; 70%; 5; **4**	[5]; 50%; 4; **5**	[7]; 50%; 6; **5**	[7]; 100%; 7; **5**	[6]; 100%; 6; **5**	[6]; 100%; 6; **5**	[5]; 100%; 5; **4**
(A)	[IS_VirtualOffice] Internet Portal		[5]; 70%; 5; **4**	[5]; 50%; 4; **5**	[7]; 50%; 6; **5**	[7]; 100%; 7; **5**	[6]; 100%; 6; **5**	[6]; 100%; 6; **5**	[5]; 100%; 5; **4**
(A)	[IS_Intranet] Intranet for civil servants		[5]; 70%; 5; **4**	[5]; 50%; 4; **5**	[7]; 50%; 6; **5**	[7]; 100%; 7; **5**	[6]; 100%; 6; **5**	[6]; 100%; 6; **5**	[5]; 100%; 5; **4**

Table 1 Part of the risk assessment of the Public Registry Online

4. Conclusions

Security requirements issues are extremely important in SPL because a weakness in security can cause problems throughout the lifecycle of a line. Although there have been several attempts to fill the gap between requirements engineering and SPL requirements engineering, no systematic approach with which to define security quality requirements and to manage their variability and their related security artefacts to the models of an SPL is available.

The contribution of this work is that of providing a systematic approach for the management of the security requirements and their variability from the early stages of product line development, in order to facilitate the conformance of the SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408 (Common Criteria). Our proposal defines a systematic process based on a security requirements decision model driven by

security standards to assist in SPL security requirements definition and to facilitate products security certification. Moreover, a security variability model with which to manage the variability and traceability of the security requirements related artefacts of the SPL and its products is proposed and preliminarily validated in a case study. Consequently, our proposal allows us to make security variants selection in the requirements level instead of in the design level, as well as providing a cross-cutting view of the security variability across all security development artefacts and assisting in maintaining the different views of variable security requirements artefacts consistent.

Finally, further work is also required to refine the prototype of our CARE (Computer Aided Requirements Engineering) tool which we have developed to support SREPPLine and the Security Resources Repository (which was one of the lessons learned in the case study performed at the Spanish Public Administration partially described in this paper), in order to assist in the complex management and maintainability of the variability and traceability relations. Furthermore, we shall carry out a refinement of our approach by proving it with a complete and exhaustive real case study of SREPPLine and its CARE-tool in order to validate and illustrate SREPPLine in far greater depth, with the aim of providing an holistic framework for security requirements engineering in SPL.

5. Acknowledgments

This paper is part of the ESFINGE (TIN2006-15175-C05-05) and ELEPES (TIN2006-27690-E) projects of the Ministry of Education and Science (Spain), and of the MISTICO (PBC-06-0082) project of the Castilla – La Mancha Regional Government.

8. References

[1] A. Birk and G. Heller, "Challenges for requirements engineering and management in software product line development", *International Conference on Requirements Engineering (REFSQ 2007)*, pp. 300-305, 2007.

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

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

[4] D. G. Firesmith, "Engineering Security Requirements", *Journal of Object Technology*, vol. 2, pp. 53-68, 2003.

[5] ISO/IEC, "ISO/IEC 13335 Information technology - Security techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management", 2004.

[6] ISO/IEC, "ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0)", 2005.

[7] ISO/IEC, "ISO/IEC 27001 Information technology -- Security techniques -- Information security management systems -- Requirements." 2006.

[8] 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.

[9] F. López, M. A. Amutio, J. Candau, and J. A. Mañas, *Methodology for Information Systems Risk Analysis and Management*: Ministry of Public Administration, 2005.

[10] D. Mellado, E. Fernández-Medina, and M. Piattini, "Applying a Security Requirements Engineering Process", *11th European Symposium on Research in Computer Security (ESORICS 2006)*, vol. Springer LNCS 4189, pp. 192-206, 2006.

[11] D. Mellado, E. Fernández-Medina, and M. Piattini, "Towards security requriements management for software product lines: a security domain requirements engineering process", in *Computer Standards & Interfaces*, vol. 30, 2008, pp. 361-371.

[12] E. Niemelä and A. Immonen, "Capturing quality requirements of product family architecture", in *Information & Software Technology*, vol. 49, 2007, pp. 1107-1120.

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

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

IEEE Computer Society Conference Publications Operations Committee

IEEE Computer Society Publications

The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at *http://www.computer.org/portal/site/store/index.jsp* for a list of products.

IEEE Computer Society *Conference Publishing Services* (CPS)

The IEEE Computer Society produces conference publications for more than 250 acclaimed international conferences each year in a variety of formats, including books, CD-ROMs, USB Drives, and on-line publications. For information about the IEEE Computer Society's *Conference Publishing Services* (CPS), please e-mail: cps@computer.org or telephone +1-714-821-8380. Fax +1-714-761-1784. Additional information about *Conference Publishing Services* (CPS) can be accessed from our web site at: *http://www.computer.org/cps*

IEEE Computer Society / Wiley Partnership

The IEEE Computer Society and Wiley partnership allows the CS Press *Authored Book* program to produce a number of exciting new titles in areas of computer science and engineering with a special focus on software engineering. IEEE Computer Society members continue to receive a 15% discount on these titles when purchased through Wiley or at: *http://wiley.com/ieeecs*. To submit questions about the program or send proposals, please e-mail jwilson@computer.org or telephone +1-714-816-2112. Additional information regarding the Computer Society's authored book program can also be accessed from our web site at:
http://www.computer.org/portal/pages/ieeecs/publications/books/about.html

Revised: 21 January 2008

CPS Online is our innovative online collaborative conference publishing system designed to speed the delivery of price quotations and provide conferences with real-time access to all of a project's publication materials during production, including the final papers. The ***CPS Online*** workspace gives a conference the opportunity to upload files through any Web browser, check status and scheduling on their project, make changes to the Table of Contents and Front Matter, approve editorial changes and proofs, and communicate with their CPS editor through discussion forums, chat tools, commenting tools and e-mail.

The following is the URL link to the ***CPS Online*** Publishing Inquiry Form:
http://www.ieeeconfpublishing.org/cpir/inquiry/cps_inquiry.html