School of Economics and Management
TECHNICAL UNIVERSITY OF LISBON

INJECTING SECURITY INTO INFORMATION SYSTEMS DEVELOPMENT

Michael Lapke
Rhode Island College – School of Management – Department of Accounting and Computer Information Systems

Abstract

This paper will attempt to reconcile the apparent developmental duality (Baskerville, 1992) between Information Systems (IS) development and IS security development. IS Development and IS Security Development each have a substantial foundation of literature on their respective approaches and methodologies. Iivari, Hirschheim and Klein (2001) provide a dynamic framework for classifying IS development approaches and methodologies. Besides providing a method for classification, this framework demonstrates the rich history behind IS Development. Baskerville (1993) likewise provides a valuable literature history for IS security development. We believe part of the cause of the security problems that continue to plague organizations (Dhillon, 2001) is ad hoc security implementation (Baskerville, 1993). This "security after the fact" can lead to an incompatibility between the system and the security of the system. Our argument is that a theoretically grounded and methodological approach is lacking for integrating security with Information Systems Development.

Keywords: Information Systems Development, Security.

1. INTRODUCTION

Security issues continue to plague organizations (Dhillon, 2001) despite steadfast efforts at controlling the problem. According to the 2004 CSI/FBI Computer Crime and Security Survey, total monetary losses due to security breaches topped 141 million dollars (Gordon, Loeb, and Lucyshyn, 2004). Over 50% of organizations surveyed reported unauthorized use of computer systems within the

Correspondence Address: Michael Lapke – Rhode Island College – School of Management – Department of Accounting and Computer Information Systems, 600 Mount Pleasant Avenue – Providence – USA, RI 02908-199. E-mail: mlapke@ric.edu

last 12 months (Gordon *et al.*, 2004). A major contributing factor to this ongoing problem may be that security tends to be an afterthought with systems development. This phenomenon, described by Baskerville (1993) as ad hoc security implementation, can lead to incompatibility between the system and the security of the system. It appears that a theoretically grounded and methodological approach is lacking for integrating security with information systems development (ISD). In this paper, we propose an integration method using the Socio-Technical Design (STD) (Lyytinen, Mathiassen, and Ropponen, 1998) approach.

Both Baskerville (1993) and Siponen (2001) describe the evolution of security development. The first three generations are identified as checklists, mechanistic engineering methods, and logical transformation methods, respectively (Baskerville 1993). Siponen (2001) extends this with a fourth generation that accounts for Dhillon and Backhouse's (2000) responsibility modeling. Both Baskerville (1992) and Siponen (2001) recognize that a developmental duality exists between security development and ISD. It is this very duality that may be at the root of many of the problems still facing IS security.

In examining this developmental duality, this paper will be structured into five parts. First, a review of how other researchers have approached the problem shall be analyzed. The second section will move into a discussion regarding ISD methodologies and will select a methodology to be used to demonstrate a theoretically grounded integration of security. The third part will overview the underlying theoretical framework, the Socio-Technical Model. This will justify the forthcoming proposed method for security integration. The fourth part of the paper will discuss how the theoretical framework works with performing a method of ISD and security integration. The fifth part will demonstrate an example of security integration with the previously chosen ISD methodology. The paper will then conclude by summarizing the findings and identifying possible future research in the area.

2. PREVIOUS RELATED WORK

A large portion of the work we examined was rigorous and well thought out in their ISD and Security integration attempts but they were limited in their scope of the security side. Inmore, Esichaikul, and Batanov (2003) present a security-oriented extension to the object model. They focus on the analysis phase of the Object Oriented Analysis and Design (OOAD) ISD methodology but make the argument that applying and integrating stringent security at this level will permeate the entire process. They do this by changing the very structure of the object class itself to include a security extension.

The major drawback of Inmore *et al.*'s (2003) proposal is that it draws on no literature regarding the security engineering approach they chose. The paper begins with the premise that all of security amounts to Confidentiality, Integrity, and Availability (CIA). By relying solely on CIA, the authors are stuck in what Baskerville (1993) calls the third generation of security development. Soft issues (Dhillon, 2000), (Siponen, 2001) are ignored. This could lead to a system which is technically very secure but completely vulnerable to internal threats.

Jürjens, Popp, and Wimmel (2003) took a similar approach to Inmore *et al.* (2003) by creating a security extension to a particular facet of OOAD and assuming it will permeate the entire methodological process. Instead of extending the definition of the actual class as Inmore *et al.* (2003) did, they propose a method by which one can express security-related information within the diagrams in a UML system specification. They do this by providing four new stereotypes with descriptive tags. The stereotypes are secrecy, integrity, high, and critical and describe various levels and types of security. Unfortunately, Jürjens *et al.* (2003) suffers from the same problems that Inmore *et al.* (2003) do. Their view of security is relies solely on CIA. Despite their solid choice in methodologies, the lacking security foundation will lead to vulnerabilities.

Other research that has attempted to produce an integration method but is limited to the technical aspects of security (CIA) include Jones and Rastogi (2004), Breu, Burger, Hafner, and Popp (2004), and Mouratidis, Giorgini, and Manson (2003). On the other hand, some research in the area has grounded itself in a security foundation (such as the Systems Security Engineering Capability Maturity Model (SSE-CMM)) in order to give itself a solid foundation for the security side of the integration attempt. Lee, Lee, and Lee (2002) and Chan and Kwok (2001) are two examples of such attempts.

Lee *et al.* (2002) propose an integration model that intertwines all the process activities and deliverables of a Systems Development Lifecycle (SDLC) with Security Engineering (SE) activities. Though based in the SSE-CMM (and several other security models), Lee *et al.* (2002) hand-picked the SE components they thought were the most important to include in their integration model. Despite the fact that Lee *et al.* (2003) validated their model *post facto* with nine experts, the method of integration was not based in a rigorous method. It was hand crafted and allows for the incorporation of organizational and supporting processes. Lee *et al.* (2003) provided a notable end result with their efforts but this paper seeks to improve on their work by grounding our integration method on a theoretical framework.

Chan and Kwok (2001) performed a similar analysis using the SSE-CMM but attempted the integration with an OOAD methodology. Interestingly, like Lee *et al.* (2002), they hand-picked what portions of the SSE-CMM should be integrated without relying on any underlying theoretical framework. Like the differentiation between Lee *et al.*'s (2003) work and this paper, it is hoped that grounding this paper in a theoretical framework will give it the power to add to the literature.

3. ISD METHODOLOGY

Besides the techno-centric security development focus (CIA) and lack of theoretical grounding for integration method, a prevailing trend in the previously discussed literature was a lack of discussion on the justification for choice of ISD methodology. Ulrich (2003) points out this lack of critical thinking that often accompanies the decision making process of ISD methodology choice. This section of the paper will provide a brief overview of ISD methodologies and justify our choice of methodology for integration. The section will be driven by Iivari, Hirschheim, and Klein's (2001) philosophical approach to ISD methodology classification. Iivari *et al.* (2001) begin their classification at an ontological and epistemological level but this paper will focus towards the end point of their classification: methodologies.

An appropriate starting point might be the exhaustive list of methodologies provided by Iivari *et al.* (2001). Of their own list, Iivari *et al.* (2001), choose 11 methodologies which were representative of all of the four paradigms in which to explore. While admirable from a philosophical perspective, this approach is not pragmatic. For example, one of the four paradigms was identified as radical structuralism which contained a single methodology, "trade unionist." Outside of this article, no mention of this methodology can be found in any of the major IS journals. "Major IS journals" would include MIS Quarterly, Information Systems Research, Communications of the ACM, Journal of MIS, and Management Science. A second paradigm included was neo-humanism. This too had one methodology that is available, speech-act based information analysis methodology with computer-aided tools. Like the trade unionist approach, this is not a much cited methodology. Hence, these methodologies shall be eliminated from the analysis presented in this paper.

The two remaining paradigms that Iivari *et al.* (2001) provided were social relativism and functionalism. The social relativist paradigm is relatively obscure, at least in the United States, but it does have a noted methodology. This is Soft Systems Methodology (SSM) (Checkland, and Scholes, 1990). The functionalist paradigm, on the other hand has been a dominant ISD paradigm in the U.S. There are several methodologies within this paradigm that stand out, based on the great number of citations in the major IS journals and their frequent use in practice. These are Structured Analysis and Design (SAD) (Randell, 1969), OOAD (Booch, Rumbaugh, and Jacobson, 1999), and ETHICS (Hirschheim and Klein, 1994).

Because it is based in the same socio-technical epistemology as the upcoming theoretical framework, it is tempting to use Hirschheim *et al.*'s (1994) ETHICS methodology. It is also tempting to use Checkland *et al.*'s (1994) SSM methodology due to its heavy focus on soft issues. Both of these methodologies will be rejected though because of their relative obscurity in the U.S. practitioner world. Between the remaining two methodologies, SAD and OOAD, OOAD is the best choice. This is because OOAD is the emerging dominant ISD methodology that is consistently making inroads to the mature SAD paradigm (George, Batra, Valacich, and Hoffer 2004).

4. SOCIO-TECHNICAL THEORY

The notion of socio-technical system emerged from the labor studies conducted by the Tavistock institute in British coal mining industry in the 1950s. The concept of socio-technical system emphasizes the inter-relationship between humans and technology in an organization. The focus is to enhance efficiency without ignoring human work or social conditions. It is a general approach to the analysis and design of organizational structures. The major sources of influences on the socio-technical perspective have been the concept of socio-technical system, research on small groups, system theory, and principles of job design.

An open socio-technical systems framework, as influenced by system theory, considers work operations as (Badham *et al.* 2000):

- Systems with interdependent parts
- Open systems adapting to and pursuing goals in external environment
- Open socio-technical systems possessing an internal environment made up of separate but interdependent technical and social sub-systems.
- Open socio-technical systems with equifinality, that is, in which system goals can be achieved by different means.
- Open socio-technical systems in which performance depends on jointly-optimizing the technical and social sub-systems.

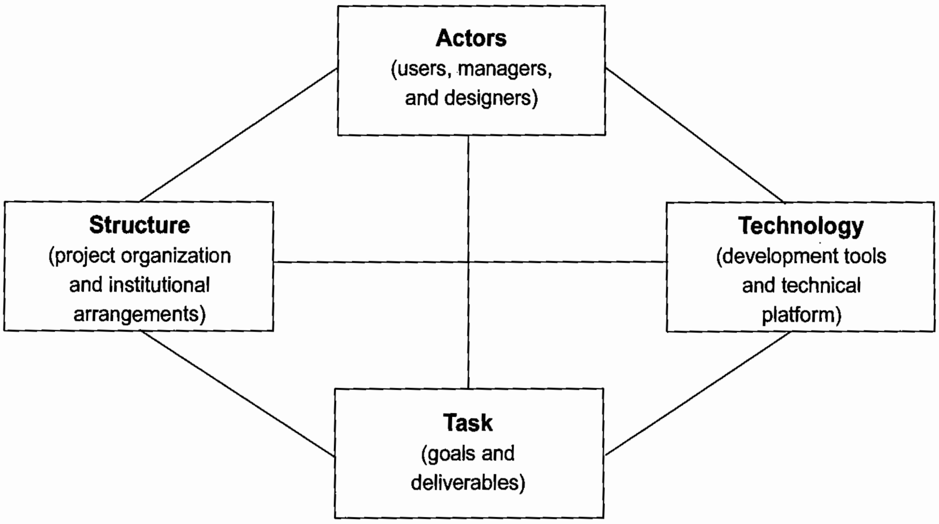
The socio-technical design principles can be used to guide the individual jobs, technology, work processes and organizational structure. Cherns (1987) has advocated a comprehensive set of these principles. The first principle of compatibility means that the process of design should be compatible with the design objectives. The second principle of minimal critical specification implies that the means of achieving the objectives should be specified, while the objectives should be. The third principle refers to variance control, where the variances should be controlled at the source. The fourth principle is the boundary control. The boundaries should not be drawn so as to impede sharing of information, knowledge or learning. The principle of information flow requires that information should be provided to those who require it when they need it. According to the principle of power and authority, people should have access to resources and authority to command them in order to carry out their responsibilities. The multifunctional principle deals with the multiple roles of individuals and teams to increase their response repertoires. The principle of support congruence implies the need of congruency between supporting systems and sub-systems. According to the principle of transitional organization, periods of transition require planning and design, and transitional organizations may be different from the old and the new systems, and are themselves subject to socio-technical design. Finally, the principle of incompletion implies that redesign is continuous and is the function of self-regulating teams.

4.1. Socio-technical model for security considerations

For the purposes of this paper, we adopt Lyytinen *et al.* (1998) socio-technical model of systems development to analyze the information systems (IS) security issues. This model was adapted from Leavitt's (1964) open system model of organizational change. Leavitt's model viewed organizations as comprised of four interacting components – task, structure, actor, and technology. Lyytinen *et al.* translated these components into elements of systems development. The socio-technical model of systems development is presented in figure 1. The following discussion is based upon Lyytinen *et al.* explanation of the socio-technical model.

FIGURE 1

A Socio-Technical Model of Systems Development


Actors
(users, managers, and designers)
Structure
(project organization and institutional arrangements)
Technology
(development tools and technical platform)
Task
(goals and deliverables)

The actor component of the model includes various stakeholders in an organization including users, managers, developers and designers. The structure component involves systems of communication, authority and work flow. It includes both the normative (values, norms) and the behavioral dimension. The technology involves different types of tools, methods, hardware and software platforms utilized to develop and implement a system. The task component signifies expected outcomes in terms of goals and deliverables.

An important consideration in this model is that all the four components are related to each other. As such, any changes in one component would have an effect on the others as well. Consequently, any unwarranted condition or state at one component would have an adverse affect on the other components as well as on the entire system. Therefore, the goal is to control any adverse variations in the system and maintain the systems in balance. In the light of the above, the interdependencies between actors, structure, technology, and task assume an important role.

Actor-Structure interdependencies focus on interactions between the structure and the actors. As per Lyytinnen *et al.* typical concerns are: incentive schemes and sanctions, values and beliefs, and how actors' behaviors are in concordance with the prevailing organizational structure. *Actor-Technology interdependencies* deal with the variations created by the misalignment between people and technology. This might arise from implementing untested technologies or mismatching people with inappropriate technology.

Technology-Structure interdependencies address the interactions between technology and the organizational structure. The variations arise as a result of conflict and disparity between the implemented technologies and the existing structures. *Task-Technology interdependencies* focus on the technological fit with the task. *Task-Structure interdependencies* deal with the interactions between task and organizational structure. Structures should be aligned with the organizational goals. A misalignment between the two would lead to inefficient outcomes.

Based on the preliminary work inspired by our understanding of the socio-technical model we derive a security framework, which is presented in Table 1. This framework accounts for the techno-centric security aspects covered by many other integration literature (CIA) but also includes the soft issues that so many of them lack. These include Responsibility, Integrity, Trust, and Ethicality (Dhillon *et al.* 2000), and culture, norms, and beliefs (Backhouse *et al.,* 1996). Furthermore, it also includes the critical backbone of any secure environment, the security policy (Baskerville *et al.*, 2002). This holistic view of security, grounded in the socio-technical model will provide the foundation for any security integration efforts.

TABLE 1

Security STD framework

Components	Security Issues	Seminal Work
Actors	Responsibility Integrity Trust Ethicality	Dhillon and Backhouse (2000)
Structure	Culture Norms Beliefs	Backhouse and Dhillon (1996)
Task	Security policy	Baskerville and Siponen (2002)
Technology	Confidentiality Integrity Availability Non-repudiation	Howard (1995)

5. USE OF THE FRAMEWORK FOR ISD-SECURITY METHODOLOGICAL INTEGRATION

As per Lyytinen *et al.*, "a change in any socio-technical component or relation in a systems development process can create variations which, in the extreme, can lead to a failure of the system development (system), otherwise known as a loss." The argument of this research is that IS security should be addressed in terms of the four components of the socio-technical model and their interdependencies. This would lead to an efficient and effective integration of security in the information systems development approaches. The security issues need to be derived from applicable theoretical basis in the IS security research literature. In order to minimize the loss or avoid system failure, these security concerns should be adapted to integrate with the socio-technical model of system development.

STD approach is adopted in this research as it concentrates on both the technical and social sub-systems of an organization. This takes into account not only the formal aspects of an organization but informal aspects like norms, culture as well. Any negative variation in either of these sub-systems or their interactions would lead to an adverse impact on the effectiveness of the organization. Further, socio-technical model takes into account the organizational structures and reminds us of the importance of the alignment with business objectives. The core focus of the socio-technical model is on the working organization. This objective fits the concerns raised in this paper as the goal of IS security is also towards a working organization.

Based on the proposed security STD framework, we would show how it can be incorporated in a given systems development methodology. To do this, one must add an empty column to the previously mentioned framework. In this column, the discrete components that make up a given ISD methodology would be appropriately placed. This is not something that can be done in a non-rigorous fashion but must be a verified and iterative process. Certainly, for any given methodology, this opens up a door to a plethora of future research. The verification of a methodology within this framework and the resultant integrated methodology would occupy several iterations of work.

6. IMPLEMENTATION OF THE INTEGRATION METHOD

In this paper, we shall be presenting the first iteration of ISD-Security integration, grounded in the STD theoretical framework. The ISD to be used is OOAD, whose characteristics are grounded in George *et al.*'s (2004) work. According to George *et al.* (2004), OOAD is driven by the Rational Unified Process (RUP). RUP's phases of development are inception, elaboration, construction, and transition (George *et al.* 2004). Each of these phases contain all of the SDLC phases, which include planning, analysis, design, and implementation, and operation. RUP is an iterative process by which the entire SDLC is cycled through for each phases of the RUP process. Depending on the RUP phase, more or less of an emphasis is placed on any given SDLC phase (George *et al.* 2004).

In the inception phase, most of the focus is on the analysis subphase. There is still some activity in design and implementation but it is only a fraction of analysis. In the elaboration phase, emphasis on analysis drops by about 50% and design increases dramatically to take center stage. Implementation and operation are still very low in this phase but slightly higher than the inception phase. In the construction phase, analysis drops by another 50%, as does design. Implementation becomes the main focus with operation rising to equal status to design. In the final transition phase, analysis and design are very low and implementation drops off slightly. In this phase, operation becomes the primary focus. These varying levels of focus are illustrated in Figure 2.

With the basic characteristics of OOAD outlined, it is time to attempt the first iteration of security integration into the methodology. This attempt will be presented in table 2 and discussed in the following paragraph.

After determining the appropriate placement of ISD methodological components, as is attempted in table 2, it is time to begin the verification process. It is here that "the devil in the details" comes out. For example, how would the security issues for the actor component be integrated into the management described in the RUP process? A decision would have to be made for the actual implemen-

FIGURE 2

The Rational Unified Process

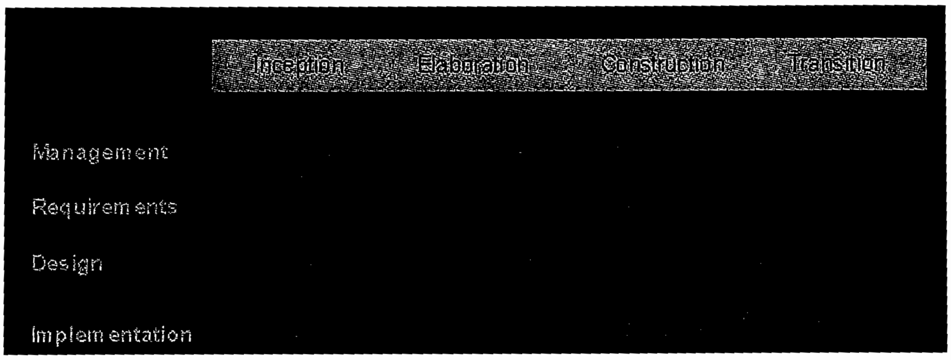


TABLE 2

Security STD framework Guiding an OOAD Security Integration

Components	Security Issues	Seminal Work	OOAD Methodology
Actors	Responsibility Integrity Trust Ethicality	Dhillon and Backhouse (2000)	Management Iterative Nature
Structure	Culture Norms Beliefs	Backhouse and Dhillon (1996)	Transition
Task	Security policy	Baskerville and Siponen (2002)	Inception Elaboration
Technology	Confidentiality Integrity Availability Non-repudiation	Howard (1995)	Construction

tation for each STD area. It may be determined that simple awareness training of responsibility, integrity, trust and ethicality for the project managers might satisfy this component. How could one verify that soft security issues relating to culture, norms, and beliefs be upheld during RUP's transition phase?

A hot spot that jumps out with this first pass through of integration is security policy. With inception and elaboration being the OOAD area where this security issue is integrated, a significant change to the typical process is introduced. Instead of fashioning a security policy in a separate or *post facto* fashion, it is now required to be integrated into very early phases of the ISD methodology. Requiring a synchronous security policy creation across inception and elaboration forces serious security considerations into the ISD methodology an early and critical point.

Another hot spot that emerges is the inclusion of the techno-centric security considerations in the final phase of the methodology, construction. This follows the work of Siponen (2001) and Dhillon *et al.* (2000) who acknowledge the criticality of the technical aspect of security but only after soft issues have been addressed. Instead of worrying about encryption, access control, and password protection in the critical early phases, these issues are relegated to the end of the process. If the security policy and actor issues are addressed soon in the process, the technical aspects will be more likely to be better engineered.

This section barely scratches the surface of this first iteration of an actual security integration. The purpose of the paper however was simply to provide a theoretically grounded method by which one could perform security integration with an ISD methodology. The example provided is only supposed to illustrate how one might begin the process. It is beyond the scope of the paper to provide a complete illustration.

7. CONCLUSION

We view this paper as a starting point for future work on integrating security considerations into IS development, and believe that it contributes toward a more sound foundation. It is hoped that this starting point will reduce the perceived ad hoc nature of ISD security research. Given that the research is attempting to alleviate the problem of ad hoc security development, this is a step in the right direction.

Our proposed security STD framework answers many of the questions posed by the critiques of the previous research. A solid theoretical foundation, in the STD framework, provides the rigor while the applicability the integration method provides the relevance. Based on the thorough review of existing literature, it is quite clear that a theoretically grounded and methodological approach is lacking for integrating security with ISD.

Future research could take these concepts in several directions. The most obvious area of future research would be to tackle a full integration effort, based on the proposed theoretical framework. One could even continue with the OOAD example and follow the iterative process through to a fully functional secure OOAD methodology. This new methodology (which could be called OOAD-sec) could then be tested in a case study setting. One could also take a different methodology such as SAD or SSM and perform a full security integration.

References

Backhouse, J. and Dhillon, G. (1996) "Structures of responsibility and security of information systems". *European Journal of Information Systems*, 5 (1), pp. 2-9.

Badham, R., Clegg, C., and Wall, T. (2000). "Socio-technical theory." In W. Karwowski (Ed.), *Handbook of Ergonomics*. New York: John Wiley.

Baskerville, R. (1993). Information Systems Security Design Methods: Implications for Information Systems Development. *ACM Computing Surveys*, 25 (4).

Baskerville, R. (1992). The Developmental Duality of IS security. *Journal of Management Systems*, 4 (1), pp. 1-12.

Baskerville, R., and M. Siponen (2002). "An information security meta-policy for emergent organizations." *Logistic Information Management*, 15 (5/6).

Breu, R., Burger, K., Hafner, M., and Popp, G. (2004). "Towards a systematic development of secure systems." *Information Systems Security*, May/June, pp. 5-13.

Chan, M., and Kwok, L. (2001) Integrating security design into the software development process for e-commerce systems. *Information Management & Computer Security*, 9 (2/3).

Checkland P., and Scholes J. (1990). Soft Systems Methodology in Action. Wiley: Chichester

Cherns, A. (1987). "Principles of socio-technical design revisited." *Human Relations*, 40, pp. 153-162.

Dhillon, G. (2001). "Challenges in Managing Information Security in the New Millennium." *Information Security Management: Global Challenges in the New Millennium*: 1-8.

Dhillon, G. and Backhouse, J. (2000). Information System Security Management in the New Millennium. *Communications of the ACM*, 43 (7).

Gordon, L., Loeb, M., Lucyshyn, W., and Richardson, R. (2004). The State of Information Security, 2004. *CIO Magazine and PricewaterhouseCoopers.*

Hirschheim, R., and Klein, H. (1994). Realizing Emancipatory Principles in Information Systems Development: The Case for ETHICS. *MIS Quarterly*, 18 (1), pp. 83-109

Howard, J. (1995). Analysis of Security Incidents on the Internet 1989-1995. Doctoral dissertation.

George, J., Batra, D., Valacich, J., and Hoffer, J. (2004). Object-oriented Systems.Analysis and Design. Upper Saddle River, New Jersey, Pearson Education, Inc.

Iivari, J., Hirschheim, R., and Klein, H. (2001). "A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches." *Journal of Management Information Systems*, 17 (3), pp. 179-218

Inmore, S., Esichaikul, V., and Batanov, D. (2003). A Security-Oriented Extension of the Object Model for the Development of an Information System. *Information Systems Security*, May/June.

Jones, R.L., and Rastogi, A. (2004). "Secure Coding: Building security into the Software Development Life Cycle." *Information Systems Security*, November/December, pp. 29-39.

Jürjens, J., Popp, G., and Wimmel, G. (2003). Use Case Oriented Development of Security-Critical Systems. *Information Security Bulletin*, 8 (2), pp. 55-60.

Karabacak, B., and Sogukpinar, I. (2005). "ISRAM: Information security risk analysis method." *Computers & Security*, 24, pp. 147-159.

Kwok, L., and Longley, D. (1999). "Information security management and modeling." *Information Management & Computer Security*, 7 (1), p. 30.

Lee, Y., Lee, J., and Lee, Z. (2002) Integrating Lifecycle Process Standards with Security Engineering. *Computers & Security*, 21 (4), pp. 345-355.

Lyytinen, K., Mathiassen, L., and Ropponen, J. (1998). Attention Shaping and Software Risk-A Categorical Analysis of Four Classical Risk Management Approaches. *Information Systems Research*, 9 (3), pp. 233-255.

Mouratidis, H., Giorgini, P., and Manson, G. (2003). "Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems." Lecture Notes in Computer Science, 2681, pp. 63-78.

Schultz, E.E., Proctor, R.W., Lien, M., and Salvendy, G. (2001). "Usability and security: An appraisal of usability issues in information security methods." *Computers & Security*, 20 (7), pp. 620-634.

Siponen, M. (2001). An analysis of the recent IS security development approaches: descriptive and prescriptive implications. Chapter 8 in Dhillon, G (2001). Information Security Management: Global Challenges in the New Millennium. Hershey: Idea Group.

Ulrich, W. (2003). Beyond methodology choice: critical systems thinking as critically systemic discourse. *Journal of the Operational Research Society*, 54 (4), pp. 325-342.

Yourden, E., and DeMarco T. (1979) Structured Analysis and Systems Specification. Englewood Cliffs, NJ. Prentice Hall.

Resumo

Este artigo procura conciliar a aparente dualidade (Baskerville, 1992) entre o desenvolvimento dos Sistemas de Informação (SI) e o desenvolvimento da segurança dos SI. Tanto o desenvolvimento dos SI como o desenvolvimento da segurança dos SI possuem abordagens e metodologias teóricas robustas. Iivari, Hirschheim e Klein (2001) apresentam um framework dinámico para a classificação das diferentes abordagens e metodologias sobre o desenvolvimento dos SI. Baskerville (1993) oferece um importante levantamento teórico sobre o desenvolvimento da segurança dos SI. Acreditamos que parte dos motivos dos problemas relacionados a segurança, que continuam a afligir as organizações (Dhillon, 2001), sejam as implementações *ad hoc* de segurança (Baskerville, 1993). Esse tipo de segurança "após o facto ter ocorrido" pode levar a uma incompatibilidade entre o sistema e a segurança do sistema. Nosso argumento é que está a faltar uma fundamentação teórica e uma abordagem metodológica para a integração de segurança ao desenvolvimento de Sistemas de Informação.

Palavras-chave: Desenvolvimento de Sistemas de Informação, Segurança