A DECISION MATRIX APPROACH

to prioritize holistic security requirements in e-commerce

Albin Zuccato

Karlstad University, Department of Computer Science, Universitetsgatan 2, 65188 Karlstad, Sweden

Abstract: In security management, the concept of security requirements has replaced risk analysis when assessing appropriate measurements. However, it is not clear how elicited requirements can be prioritized? State of the art methods to prioritize the holistic nature of security requirements are applicable only after major revisions. This dilemma is the starting-point for proposing a qualitative decision matrix approach which is quick and where the results are reproducible and sufficiently accurate. This article describes how the parameters for a prioritization are derived and how the prioritization is carried through.

Key words: decision matrix, holistic security requirement, security requirement prioritization

1. INTRODUCTION

In recent years the term security requirement has become more and more popular in the security management community. The purpose of a security requirement is to guide the implementation and ongoing administration in security management [ISO 13335-1, 1996]. In earlier years, a security requirement was mainly interpreted as a factor that had to be derived from a risk analysis process – see [ISO 13335-1, 1996], [ISO 17799, 2000]. The risk value then clearly indicated the importance of the requirement. The more severe the risk was, the higher was the incitement to realize the requirement. In that manner a priority order, dependent on the risk value, can be established and the resources can be dedicated to the most important requirements. This is

necessary as we assume that only limited resources are available which are insufficient for realizing all security requirements.

However, for to e-commerce applications [Zuccato, 2004] suggests that also the stakeholder and the environment can in addition to impacts of risks on assets also provide valuable inputs to the holistic security requirements. This broadening of a security requirement implies that the conventional mechanism for prioritization is no longer suitable. Therefore we propose the decision matrix approach, which relies on a strategic management method in order to prioritize business activities, called the Boston Consulting Group (BCG) Matrix, and adapt it to the security area. The proposed approach is described later in the article, where also an application example is provided.

Apart from the functional demands we proposed in [Zuccato, 2002a] that an approach that works in an e-commerce environment should also fulfill additional demands. One demand for each decision method should be that the results can be reproduced later[4]. Another demand that is specifically important in e-commerce is short time-to-market cycles – therefore a ranking method must be fast.

To justify the proposal of a new approach we will start by discussing related work on requirement prioritization approaches from the security field as well as the software engineering community. Shortcomings that make those approaches unsuitable for the discussed problem will be pointed out.

2. REQUIREMENT PRIORITIZATION TODAY

The concept of requirements is a recent trend and currently heavily influenced by the previous approach of risk analysis. [ISO 17799, 2000] mentions for example security requirements, but has risk analysis as the only source. This implies that risk management concepts can be applied for prioritization. [ISO 17799, 2000] and, based on that, [CCTA CRAMM, 1996], argue that the asset value and the savings indicate the risks that should be mitigated. The problem with this assumption is that risks are taken as the only source for security requirements. [Zuccato, 2004] states that security in e-commerce cannot solely rely on risk analysis. Additional input from business and stakeholder have to be included in order to cover a broader picture. Such requirements are then no longer expressed in terms of risks for an asset. Therefore the old prioritization (higher risks first) is inappropriate.

[4]One argument in favor of that is liability claim to a court. With a reproducible process it is easier to prove an honest and negligent behavior.

As an alternative to risk concept, sometimes business metric systems are used – see [Gordon et al., 2004]. Prominent examples used in the security field are Return of Security Invest (RoSI) [Wei et al., 2001] or Net Present Value (NPV).

RoSI conducts a cost-benefit analysis almost in the same way that we are going to propose it. However, the fundamental difference is that that RoSI was designed to evaluate the effectiveness of security safeguards. The approach chooses a risk and then evaluates in how far a given safeguard prevents it. RoSI implicitly assumes that all risks (or mainly the most prominent risks) are considered. A similar approach is presented in [Pfleeger and Pfleeger, 2003], where risks are processed in order of their magnitude. In [Zuccato, 2002b] we argue that security (requirements) can "earn" money as a business enabler (i.e. generate a positive cash flow) and it would be wise to consider that in the cost-benefit analysis.

The NPV approach in security anticipates the occurrence of future cash flows when a risk is mitigated by a safeguard. Such cash flows would represent the annual spending and the annual savings for the anticipated risks – it would be possible to replace a risk with a requirement. However, apart from the risk related problem mentioned above, we have another problem with NPV which is that future cash flows and future interest-rates (for discounting) must be known in advance. In a highly volatile area that information security constitutes such a long term prognoses seems to be almost impossible[5] .

A third alternative is to rely on the requirement prioritization schemes from the software engineering community. Three of these approaches should be discussed as representatives.

We start with the eXtreme Programming (XP) [Beck, 2000] as a representative for the agile methods. With XP the customer is requested to define a priority for each requirement (called story). When it comes to security this implies a specific problem, namely that many customers do not realize the importance of security [Hitchings, 1995] and therefore rank it very low – as current experiences with security problems indicate. A second problem is that such decisions are hardly reproducible.

The second approach is to ask the stakeholder how (a) satisfied with the availability of a security feature or (b) unhappy with its absence they would be [Robertson and Robertson, 1999]. This approach is better than just simply asking the customer, as it probably mitigates the "dislike -factor" when

[5]NPV is also a quantitative method and as [Moses, 1992] argued, quantitative methods imply problems of data generation in the security field.

distributed to various stakeholders. Regarding the reproducibility, however, it is only slightly better.

Finally, we look at the requirement prioritization carried through in the Unified Process (UP) [Jacobson et al., 1999] as a representative for the monumental processes. [Leffingwell and Widrog, 1992] indicates that two different prioritizations are required in the UP. The first one lies on the customer's side, where he/she has to define the features required. The assumption is, in conformity with approaches presented earlier, that the decision maker possesses some kind of oracle that supports the decision making. However, it can be questioned whether this is true for security, as we assume that the decision maker seldom has enough knowledge to conduct such decisions. The second prioritization in the UP is carried through by the software architect, who decides, based on the first prioritization, which requirements should be implemented first and which ones will be postponed to later iterations or versions. It is therefore necessary to assume that they are initially ranked highly enough when considering security requirements, so that they will be implemented also after the second prioritization. It is obvious that this assumption is doubtful as the same decision restrictions as above can be applied.

These problems with each of the above mentioned methods indicate that they are not entirely suitable and could only be applied after major adoptions. We therefore propose a different approach used in strategic management when deciding which products (features) are required on the market which also is suitable for the security field and security requirements.

3. PORTFOLIO ANALYSIS

In strategic management, one of several important tasks in order to survive in the competitive market and to maximize the profits is to find the optimal product portfolio. As a result of that, the portfolio analysis was proposed in the 1970ies to find out the actual product's position on the market. Based on that information the further steps were planned.

The first approach came from the [[Boston-Consulting-Group, 1972] (BCG) and today, thanks to its simplicity, it is still the most frequently used one, and it will be investigated further on in this article.

3.1 Boston Consulting Matrix

The BCG Matrix is based on two criteria: the reference market's growth rate (acting as an indicator for the attractiveness) and the market share in relation to the firm's largest competitor (measuring competitiveness). A

large market growth means that the product is mostly at the begin of its life cycle and has the potential to get large parts of the market although not having it yet. In the matrix – Fig. 1 – these two criteria form the axes. Additionally the matrix is divided into 4 zones, where each of them intuitively represents the products position on the market.

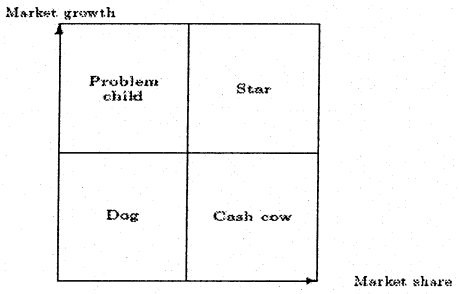

Figure 1. Product portfolio matrix after BCG

After defining the duple for each product, the value pair is going to be drawn in the matrix. Based on the position, different strategies are proposed (see for example [Lambin, 1997]).

Cash cow (a well situated, profitable product) The priority strategy is to earn money.

Dog (an old product for divestment) The priority strategy is to divest.

Star (a young product with market potential) Investment is recommended to make the product a cash cow.

Problem child (product in start-up phase, which needs placement) Depending on the relative position in the quadrant, two strategies are possible: an investment strategy to make the product a cash cow, or a divestment strategy to make the product a dog.)

[Lambin, 1997] argues that although the initial assumptions may be restrictive – but assumably correct – an accurate and valuable recommendation can be generated. An advantage worth to emphasize is that the matrix is straight forward and intuitive and therefore easy to understand and apply.

4. DECISION MATRIX FOR SECURITY REQUIREMENTS

In the previous section, the BCG matrix was introduced as a tool when deciding how a portfolio should be developed further. The problem is

similar when it comes to security requirements: how do we decide which ones should be developed first and which ones can wait? To conduct this decision we first need to position the requirements in the matrix. Then we can derive a priority list. Additionally, the position in the matrix can suggest a course of action for the treatment.

The positioning mentioned is the difficult part of the approach as it is the non-mechanistic (creative) one. Corresponding parameters to attractiveness and competitiveness must be derived for each requirement. When the requirements are parameterized, the mechanistic part of the priority generation must be conducted. Before going into more detail for each step we will provide an overview for our approach.

4.1 Approach

To begin with, it is necessary to assess a requirement according to its potential, i.e. to generate something similar to the tuple of attractiveness and competitiveness used in the BCG Matrix. Each requirement should be represented as a tuple containing the perceived security benefit and cost-complexity of the realization.

Security benefit To reflect competitiveness of a requirement we propose to use the perceived security benefit. Security benefit should mean either (a) that the requirement provides high protection of own resources and/or (b) that the requirement will increase the security benefit as it enables business. This is based on the underlying assumption for holistic security requirements, where they not only insure company resources but also enable the selling of the product because of a competitive advantage gained from the satisfaction of security needs from customers – for a more elaborate discussion of these security drivers see [Zuccato, 2002b]. Then we can say that the higher the security benefit of a requirement is, the more competitive it is in respect to other requirements.

Cost/Complexity We think that attractiveness of the requirement is represented best by its costs of realization and the associated complexity. These factors represent in how far the requirement is likely to fulfill its perceived function. The more it costs and the harder it is to realize, the higher the stake is. However, the cost-complexity measure makes only sense in relation to the intended security level. It is important to mention that a requirement that is easier and cheaper to enforce than a second one with he same benefit should be prioritized, and it most definitely does not mean that the cheap and easy way is always the best solution.

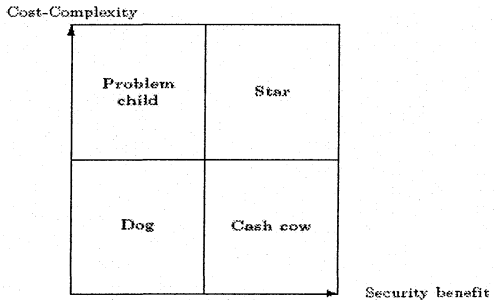

Figure 2. Requirement prioritization matrix

Before applying the matrix concept, the meaning of the quadrants needs to be set into relation to the input values. This is necessary as the complexity and costs are indirect proportional to the benefit. More benefits and limited costs are preferable. Additionally the quadrants must be redefined to reflect the scope of security requirements.

Dog means that not only the complexity and the costs are low, but also are the benefits. The requirement has an indifferent potential.

Problem child means that the complexity and the costs are high and the expected benefit is low. The potential of the requirement is low.

Cash Cow means that the complexity and the costs are low but the expected benefit is high. Such a requirement is very promising to realize as it has high potential.

Star means that both the cost-complexity and the benefits are high. Although such a requirement is interesting its realization is also highly risky. Therefore, as for the dog, the potential is indifferent.

4.2 Input data elicitation

The approach for every security requirement is to elicit the perceived security benefit and the cost-complexity level. Due to several reasons of impracticability of quantitative methods we will use a qualitative approach. Firstly we think that the required empirical data for a quantitative estimation is hard to provide due to the high dynamics in the security field – see [Moses, 1992]. Secondly, we think that most quantitative estimates require an "oracle" – most likely statistical prediction or a simulation – which would not necessarily provide more accuracy than a qualitative estimate (i.e. an expert guess). And thirdly, we think that an empirical method is more prone to violate our quickness requirement for a prioritization method.

However, as the goal is to achieve reproducibility and acceptable accuracy, we propose the conduction of a structured elicitation. We suggest the use of the Delphi method [Dalkey and Helmer, 1963] in order to predict the security benefits and the cost-complexity parameters. Delphi is a method that is used to support judgmental or heuristic decision-making – i.e. creative or informed decision making. According to [Adler and Ziglio, 1995], Delphi is a suitable method when "(a) the problem does not lend itself to precise analytical techniques; (b) the problem at hand has no monitored history nor adequate information on its present and future development [and]; (c) addressing the problem requires the exploration and assessment of numerous issues". We think that all these factors are accurate in concern of our elicitation problem. Alternative approaches to Delphi could be brainstorming or questionnaires. However, both alternatives can create problems in the reliability and are eventually subjects to the "dislike" problem mentioned above.

"The Delphi method is based on a structured process for collecting and distilling knowledge from a group of experts by means of a series of questionnaires interspersed with controlled opinion feedback" [Adler and Ziglio, 1995]. In the beginning a questionnaire is sent to selected experts. The filled-in questionnaires are collected and aggregated as a second step. Different ways to derive the aggregates are possible, but here a mean value approach has been used. The mean-value should then be rounded to the next integer to avoid positioning problems in the evaluation. The aggregates constitute feedback to the experts, and in case of to big variation – decided by the method performer – the experts are requested to further state or revise their opinions. This process is conducted until the intended accuracy is achieved. Note that the higher the accuracy demand is, the higher the cost will be – which holds true for all decision methods.

The design of the questionnaire mentioned above is important in order to achieve satisfactory inputs for the result generation – i.e. the requirement prioritization. To perform the subsequent prioritization process efficiently we need to have sufficient parameter information without adding much complexity to the prediction – which would require additional time. We therefore propose the use of an ordinal scale for the parameter. To derive the scale, according to [Fowler, 1995], one must design the granularity to (a) achieve validity, and (b) make the elements of the distribution distinguishable. This would indicate that the higher the granularity is, the better. However, [Fowler, 1995] says that 5 to 7 categories are probably as many categories as most respondents can use meaningfully. This means that we will aim for a six value scale as our scale must be a multiple of two to correspond with the quadrant structure of the matrix. The quadrants should be made explicit to the respondents by introducing a neutral point in between,

as [Fowler, 1995] says that neutral points help to reduce ambiguity. Therefore we introduce a neutral point between 3 and 4 where 1 - 3 represent low and 4 - 6 represent high. [Fowler, 1995] suggests to use numbers for reliability reasons, but to provide adjectives for clarification of the categories' meaning. Based on that we propose the following scale for each parameter:

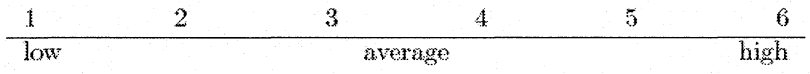

Figure 3. Ordinal scale for the survey

To derive the categories for each requirement, two questions should be asked for each requirement.

- • How much security benefit do you associate with the requirement?
- • How much complexity and cost do you associate with the realization of the requirement?

The result is then represented as a tuple:

Requirement(Benefit, Cost-Complexity)

4.3 The prioritization activity

The aim of the decision matrix is to derive, which requirements should be implemented at first. The position in the matrix suggests the priority of the requirements.

To derive the priority, we suggest two different methods which should be used dependent on the accuracy demand, the quality of the inputs and the application place. For the first method we suggest the use of the quadrants. Based on the result a priority can be derived. The second method will rely on a more formal prioritization that eventually could be automatized.

4.3.1 Informal prioritization

For the start, we assume that the requirements are placed in the matrix. The quadrants can then be used to derive a requirement priority list. This list suggests which requirements should be considered first.

In general we can say that the closer a requirement is to the right lower corner, the more preferable it is. Given the quadrants we therefore suggest the following prioritization:

Requirement list = (Cash Cows, lower Stars, lower Dogs, higher Stars, higher Dogs, Problem child)

Problem child These requirements are likely to be problematic in the implementation. The expected benefit will not justify that and they will end up low in the priority list.

Cash Cow These requirements are of great priority as much benefit is expected for the associated costs and complexity. They will all end up high in the priority list.

Star We have already mentioned that this quadrant suggests indifference. However, we can derive a priority in the way that we imagine a diagonal from the source to the upper right corner. All requirements that are below will have higher priority than the requirements above it. Therefore the "lower Stars" will follow directly after the Cash cows and the "higher Stars", and end up in the middle of the priority list.

Dog The "Dog requirements" are similar to the stars when it comes to indifference. The same diagonal as mentioned above can be used to derive priorities. "lower Dogs" will come after the "lower Stars" and "higher Dogs" after the "higher Stars" just before the "problem child" requirements.

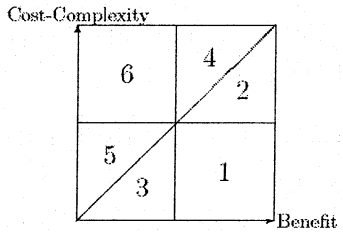

Figure 4. Informal prioritization matrix

Although the informal method can be less accurate we propose it because:

1. it is a good way to visualize the requirement prioritization for the decision maker;
2. in cases where the input variables do not provide high accuracy – because the Delphi method was abandon in favor of a faster or more suitable method in specific situations – the informal method do not introduce a fictive accuracy and;
3. in some situations – e.g. a requirement engineering workshop with stakeholders [Zuccato, 2004] – a visual and less technology dependent method is preferable

4.3.2 Formal prioritization

The formal approach starts by deriving a value for each requirement, which defines the position in the matrix. This value is calculated by dividing the Benefit with the Cost-Complexity – Equ. 1. To position that in the matrix ($a_{i,j}$) we assume that *i=Benefit* and *j=CostComplexity*.

$$a_{i,j} = \frac{Benefit}{CostComplexity} \tag{1}$$

For a 6×6 matrix we can construct generic values as shown in table 1.

Table 1. Priorities for a 6x6 Matrix

Cost \ Benefit	**1**	**2**	**3**	**4**	**5**	**6**
6	0.16	0.33	0.5	0.66	0.83	1
5	0.2	0.4	0.6	0.8	1	1.2
4	0.25	0.5	0.75	1	1.25	1.5
3	0.33	0.66	1	1.33	1.66	2
2	0.5	1	1.5	2	2.5	3
1	1	2	3	4	5	6

For prioritization we construct – as in the informal approach – a preference set. We compare two requirements with each other until we have processed all requirements[6] . This comparison leads to a preference set where $a \succ b$ means that a is preferred to b, and $a \sim b$ means that they are indifferent.

When the prioritization value of one requirement ($a_{i,j}$) is different to the other requirement ($a_{k,l}$), we construct a preference order by following the Equ. 2.

$$a_{i,j} > a_{k,l} \Rightarrow a_{i,j} \succ a_{k,l} \tag{2}$$

The prioritization value can be equal under two circumstances. In these cases a preference order should be achieved dependent on the requirement

[6]Note that this is a classical sorting problem. Therefore sort algorithms should be used to process all requirements.

parameter. If the parameters are equal, the requirements are indifferent and receive the same priority[7] – see Equ. 3.

$$a_{i,j} = a_{k,l} \wedge i = k \wedge j = l \Rightarrow a_{i,j} \sim a_{k,l} \quad (3)$$

If the parameters are different from each other, we define that more security benefit ($i>k$) is preferable, as our overall goal is to improve the system security – see Equ. 4. However, when having a limited budget this interpretation must not correspond with the truth and could be reconsidered ($j<l$).

$$a_{i,j} = a_{k,l} \wedge i > k \Rightarrow a_{i,j} \succ a_{k,l} \quad (4)$$

5. E-COMMERCE SCENARIO

We start by looking at some security requirements proposed in [Zuccato, 2004]. Those requirements have an Internet-banking scenario as a background, where the customers access their accounts and make money transfers.

1. Sensitive user data (passwords, keys ...) in a database needs to be stored bi-directionally (not hashed) encrypted due to requirements of the voice recognition system.
2. A demand of internal audit means that audit logs for the intrusion detection system must be stored for three months.
3. An activity log for each transaction should be kept for six months.
4. When saving personal information for statistical purposes, user pseudonyms should be used whenever possible to comply with the data protection legislation.
5. User authentication for accessing bank accounts and services via the internet is necessary.
6. The privacy policy must define customer profiling as one purpose for the activity logs.

We start by preparing the questionnaire for the Delphi method. Then we select some experts representing different areas to cover all aspects of the

[7]Note that this is an intended behavior as those requirements then form a priority group, where the requirements are of the same importance and the selection can be conducted based on project planning considerations.

holistic requirements. A few examples could be: a security officer, a product owner, a bank manager, a security implementer...

In this example we assume that we will receive the following parameter values after a number of Delphi iterations.

Requ.	Benefit	CostComp.
1	4	2
2	2	2
3	4	4

Requ.	Benefit	CostComp
4	2	6
5	6	2
6	2	1

Informal method

When we conduct the informal approach we must transfer the requirements to the matrix – see Fig. 5.

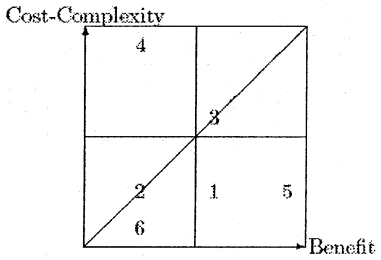

Figure 5. Qualitative security requirement decision matrix

From there we can follow our algorithm and derive a priority list. We get the following priority set: ({5,1},3,{6,2},4). Note that according to this priority list there is no priority between 5 and 1 and 6 and 2.

Formal method

To conduct the formal approach we need to calculate the values for each requirement.

Requ.	Value
1	2
2	1
3	1

Req.	Value
4	0.33
5	3
6	2

When applying the algorithm we will end up with a priority set as follows: (5,1,6,3,2,4). The difference to the informal method is that due to the higher granularity we achieved a greater accuracy in the result. Requirement 6 would have gained less attention in the informal approach than in the formal one as it is in the wrong quadrant.

6. CONCLUSIONS

It is a difficult task to make requirement prioritization easily understandable and reconstructible. In a market environment, where time-to-market cycles are measured in weeks instead of months, the speed of such a method is of considerable importance. The method presented in this article is supposed to solve these problems by enabling a prioritization based on a matrix approach common in strategic management.

By choosing this matrix approach, large parts of the prioritization work become mechanistic and therefore easy to reproduce. The non-mechanistic part uses an established prediction method to derive parameter values and can therefore be more easily reproduced. Concerning the speed, a final judgment can only be made after extensive testing. However, from a theoretical perspective, properties as simplicity and the mechanistic prioritization imply acceptable speed behavior.

In future research it would be interesting to transform this qualitative method into a quantitative one by providing means to derive the input parameters by calculatory means – as we hope that the progress in security management will provide a sufficient database for statistical prediction. It would be of great interest to learn whether this could enhance accuracy further by not decreasing speed and simplicity significantly.

A predecessor of this method was, as described above, applied once in an Internet banking environment. However, as this approach has changed partially, we plan further application in order to verify the presented ideas in this article.

REFERENCES

[Adler and Ziglio, 1995] Adler, M. and Ziglio, E. (1995). Gazing into the oracle: the Delphi method and its application to social policy and public health. London - Jessica Kingsley.

[Beck, 2000] Beck, K. (2000). extreme programming explained. Addison Wesley.

[Boston-Consulting-Group, 1972] Boston-Consulting-Group (1972). Perspectives and Experience. Boston, Mass.

[CCTA CRAMM, 1996] CCTA CRAMM (1996). CCTA Risk Analysis and Management Method. Central Computer and Telecommunication Agency, United Kingdom, user manual edition.

[Dalkey and Helmer, 1963] Dalkey, N. and Helmer, O. (1963). An experimental application of the delphi method to the use of experts. Management Science, (9): 458–467.

[Fowler, 1995] Fowler, F. (1995). Improving Survey Questions, volume 38 of Applied Social Research Methods Series. Sage Publications.

[Gordon et al., 2004] Gordon, L. A., Loeb, M. P., Lucyshyn, W., and Richardson, R. (2004). 2004 csi/fbi computer crime and security survey. Technical report, CSI/FBI.

[Hitchings, 1995] Hitchings, J. (1995). Achieving an integrated design: the way forward for information security. In Eloff and von Solms, editors, Information security – the next decade, pages 369 – 383. IFIP, Chapman & Hall.

[ISO 13335-1, 1996] ISO 13335-1 (1996). ISO/IEC TR 13335-1: 1996 Information technology – Guidelines for the management of IT Security – Part 1: Concepts and models for IT Security. International Standard Organization.

[ISO 17799, 2000] ISO 17799 (2000). ISO/IEC 17799: 2000, Information technology – Code of practice for information security management. International Standard Organization.

[Jacobson et al., 1999] Jacobson, I., Booch, G., and Rumbaugh, J. (1999). The Unified Software Development Process. Object technology series. Addison Wesley Longman.

[Lambin, 1997] Lambin, J.-J. (1997). Strategic marketing management. McGraw-Hill.

[Leffingwell and Widrog, 1992]Leffingwell, D. and Widrog, D. (2000). Managing software requirements: a unified approach. Addison Wesley.

[Moses, 1992] Moses, R. H. (1992). Risk Analysis and Management. In K.M.Jackson and J.Hruska, editors, Computer Security Reference Book, pages 227 – 263. Butterworth Heinemann.

[Pfleeger and Pfleeger, 2003] Pfleeger, C. and Pfleeger, S. L. (2003). Security in Computing. Addison & Wesley, 3ed edition.

[Robertson and Robertson, 1999] Robertson, S. and Robertson, J. (1999). Mastering the requirements process. Addison-Weseley.

[Wei et al., 2001] Wei, H., Frinke, D., Carter, O., and Ritter, C. (2001). Cost-benefit analysis for network intrusion detection systems. In 28th Annual Computer Security Conference.

[Zuccato, 2002a] Zuccato, A. (2002a). A modified mean value approach to assess security risks. In Labuschagne, L. and Eloff, M., editors, 2nd annual conference of Information Security for South Africa – ISSA-2 Proceedings. South African Computer society.

[Zuccato, 2002b] Zuccato, A. (2002b). Towards a systemic-holistic security management. Licenciate thesis, Karlstad Unviersity Studies.

[Zuccato, 2004] Zuccato, A. (2004). Holistic security requirement engineering for electronic commerce. Computers & Security, 23/1: 63 – 76.