RiskREP: Risk-Based Security Requirements Elicitation and Prioritization (extended version)[*] [†]

Andrea Herrmann[1] and Ayşe Moralı[2‡]

[1] University Carolo Wilhelmina, Braunschweig, Germany; now: Axivion GmbH
herrmann (at) axivion.com
[2] University of Twente, Enschede, The Netherlands
ayse.morali (at) utwente.nl

Abstract. Today, companies are required to be in control of the security of their IT assets. This is especially challenging in the presence of limited budgets and conflicting requirements. Here, we present Risk-Based Requirements Elicitation and Prioritization (RiskREP), a method for managing IT security risks by combining the results of a top-down requirements analysis with a bottom-up threat analysis. Top-down, it prioritizes security goals and from there derives verifiable requirements. Bottom-up, it analyzes architectures in order to identify security risks in the form of critical components. Linking these critical components to security requirements helps to analyze the effects of these requirements on business goals, and to prioritize security requirements. The security requirements also are the basis for deriving test cases for security analysis and compliance monitoring.

1 Introduction

The goal of this research is to develop and to validate a method for the systematic elicitation and prioritization of security requirements. We do this by analyzing the effects of changes in the applied security requirements.

System owners need to be in control of their information assets because regulations require it. This becomes difficult as the IT or security architecture of these systems change dynamically. For instance as the physical/logical location of a server or the authentication method changes. In order to keep track of the dynamics of the IT systems a light-weight (cost-effective) requirements elicitation process is necessary which can easily be repeated. Such a process should document requirements and why these requirements are necessary.

[*] This research is supported by the research program Sentinels (http://www.sentinels.nl). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

[†] We would like to thank the TUgether project and especially Stefan Bhme for their friendly and helpful contribution to the case study.

[‡] Author names are in alphabetical order due to equal contribution.

Elicitation of security requirements is a challenging process. During this process, tacit knowledge and expert knowledge from stakeholders in different domains need to be extracted and combined. These domains are business domain, IT domain and security domain. The requirements engineering (RE) frameworks of today aim to gather the requirements related information from system owners without explicitly differentiating which stakeholder knows the most about which domain (see Table 4). This leads to long (costly) and inefficient meetings at which all stakeholders (or their representatives) attend. In order to increase the efficiency of information elicitation and requirements management a systematic and stakeholder specific method is necessary.

As Dubois et al. [6] state, security risk models are largely unable to address cost-effectiveness concerns in a satisfactory manner. A cost-effective security evaluation requires prioritizing possible countermeasures according to their costs and effectiveness. To the best of our knowledge, there is no method that presents the effects of countermeasures on risks levels and link the security risks to business goals.

For the usability of such methods having a systematic process is vital. We call a process a systematic process when it consists of a sequence of activities. For each of these activities it is clearly defined which concepts are derived from which other concepts. Furthermore, results of each activity are documented in a traceable notation.

The importance of integrating security in RE is indicated by many researchers (e.g. [1, 2, 7, 17]). These researchers usually formally extend currently available RE methods and supporting diagrams. However, we believe that they do not have these features which we think are important for eliciting and managing security requirements accurately. We think that these features are: (a) to provide a systematic process for eliciting requirements, in order to make it usable, (b) to consider different stakeholders perspectives when eliciting input information like goals and threats - what means to differentiate between business goals and quality goals of the IT system, (c) and to consider both intentional use and misuse in order to be complete. Furthermore, to prioritize security requirements in a traceable way, a method needs to (a) quantify impact and likelihood (risk), (b) systematically draw diagrams that encore analysts and stakeholders creativity, (c) consider effectiveness of requirements at mitigating incidents, (d) provide trade-offs among contradicting requirements and (e) consider cost of implementing each requirement.

In this paper we present a systematic and practical requirements elicitation model that is supported by a method that prioritizes security requirements according to the risks they counteract. The authors developed this solution by integrating concepts from MOQARE [8] (a method for systematic requirements elicitation) and CRAC++ [11] (a method for practical yet accurate risk assessment). The new method describes step-wise how to identify the quality goals with the top-down approach of MOQARE and link them to security risk of IT assets that are analyzed with the bottom-up approach of CRAC++. The objective of this solution is to identify the most security effective set of requirements, which describe how security as the quality goal can be achieved.

We claim that the main strengths of RiskREP are: (1) Business-IT-alignment by linking business goals traceably to IT requirements and vulnerability analysis. The latter means that the priority of each IT requirement can be traced back to the business goals which it supports. This means that rationale are documented which answer the question "Why is this requirement important in this specific system?". (2) It combines a graphic overview presentation (used on the business perspective of the analysis) with better scalable table presentations on the user perspective. This is important for structuring the information elicitation process. Furthermore, (3) it provides a clear process with phases which demand well-defined knowledge and therefore specific stakeholders. It is clear which phase demands the contribution of management or of a security officer. This is important for gathering information efficiently.

2 Background

In this section, we present the MOQARE [8] and CRAC++ [11] methods. These methods constitute the basis for the requirements elicitation and prioritization method RiskREP that we introduce in this paper.

MOQARE (Misuse-Oriented Quality Requirements Engineering) is a method for the top-down elicitation of quality requirements. Its fundamental principle is to combine the elicitation of wanted elements and unwanted elements. The MOQARE analysis starts with the business perspective where business goals are identified, i.e. the reasons for developing the system. Business damages (which are defined in Section 4) also are identified. The business damages are partly caused by quality deficiencies of the system, and these are analyzed further by defining quality goals for the system. Threats to these quality goals are then elicited in the form of misuse cases [15]. Misuse cases are similar to use cases, but in scenario form describe what must not happen, such as intentional attacks or user errors. Then, one seeks for countermeasures, which are requirements on system, development or use which can detect, prevent or mitigate the misuse case. MOQARE has shown it's merits of systematically supporting the creative process of deriving realizable non-functional requirements from abstract quality goals (like data security) and documenting rationale of the requirements. This process is supported by check-lists for threat agent types, potential assets and their vulnerabilities and threats which endanger them. However, the prioritization of the misuse cases and requirements is left to experts and not supported by the MOQARE process.

CRAC++ (Confidentiality Risk Assessment and Comparison) is a method for the bottom-up specification of confidentiality requirements according to risk assessment (RA) results. Its fundamental principle is analyzing how functional and operational information can flow through an IT architecture, and how unauthorized persons can get access to information available on nodes of an IT architecture. Possible information flow determines the information that can be present in a node of the architecture, and therefore allows the risk expert to assess the severity of a confidentiality breach (information disclosure) at that node. Analysis of possible movement of unauthorized persons through the network allows the risk expert to asses the ease of an unauthorized person accessing information assets on a node. Combining this information allows the risk expert to assess the risk of confidentiality breach per node. Since quality requirements (countermeasures) aim to mitigate confidentiality breaches, their application to the nodes affect the risk. Based on this change in the risk the CRAC++ method ranks the relevance of requirements. However, it assumes that a list of possible requirements is provided and leaves their elicitation to requirements experts.

3 Case Study Description

In this section, we describe the case study that we use to validate the feasibility of RiskREP. The target system that we analyze is the student administration portal developed at the University Braunschweig (TU). The project team's motivation for participating in our security analysis case study was to learn more about the security risk level of the system and get ideas for potential improvements, but also the knowledge about which risks already are mitigated sufficiently in the current system.

The TU Braunschweig (Germany) offers to their students and teaching personnel many services which are available online, such as several e-learning platforms, registration for exams and seminars, download of documents and templates, time tables of the next semester and room reservation plans, menu plan of the university restaurant, a content management system and an email account. The project TUgether integrates all these services within one portal in order to establish an "online campus". TUgether also is the name of the portal which allows their students single-sign-on to all services of the TU Braunschweig via one individually configurable interface. The portal itself does not offer any content but just an entry point to data and services which are offered by other systems. These services stay accessible also without the portal. The project wants to offer as much added value to the users as is possible within the project budget. This means that development effort must be optimized. The objective is that all students use the portal and the portal must be adaptable to all changes in the universitys technical and organizational infrastructure. The cooperation with other universities is another objective of the project.

The first phase of the TUgether project - taking place in 2009 - meant to choose the portal framework product which best satisfies the requirements. More than 80 requirements for choosing this framework were specified and about 70 products were taken into consideration. Finally, three products were installed and tested. Out of the 80 requirements, 9 were security-related requirements, like “privacy”, but also technical means like “backup possibility”. At the point of time of this case study, the portal was in pilot operation.

In this project, we applied RiskREP in order to validate the method on a realistic example. The project teams motivation for participating in our security analysis case study was to learn more about security in general and get ideas for potential improvements, but also the knowledge about which risks already are mitigated sufficiently in the current system.

The scope of this case study is limited to security (confidentiality, integrity and availability) requirements of student information that is managed by or accessed via the portal. The case study example is a real software project, but not too complex in terms of low number of stakeholders and only one software application to be analyzed (although several more are involved). This allows to apply the RiskREP method as a proof of concept in a real project. Later-on, it will be applied to more complex case studies.

4 Meta Model

In this section, we briefly present the concepts of RiskREP and how they are connected to each other on the meta level. The meta model consist of concepts belonging to three perspectives, i.e. business perspective, user perspective and technical perspective. (See Fig. 1 for illustration.) Our main motivation for choosing these concepts is to benefit from both practical and academic acceptance of CRAC++ and MOQARE methods. Accordingly, we integrated the concepts of these methods into a method that aims to solve the identified problems. It's concepts are defined as follows:

Business goals are desired properties of the business. They are stakeholder-relative and might be supported by an IT system. Business goals finally justify system requirements. Such goals can be ”efficient business processes”.

A *business damage* is a state or activity of the business that *violates* a business goal. The business damage completes the business view by asking what should not happen.

Quality goals are desired qualities of the IT system, i.e. a desired state of the systemthe system. They are non-functional system goals that *support* business goals. These goals are furthermore high level quality requirements that consist of a quality attribute and an asset, e.g. confidentiality of data. They help to focus the analysis of an IT system's quality on the most important quality attributes and parts of the system.

Quality attributes are attributes of the system to be protected. They describe an aspect or characteristic of quality, e.g. confidentiality. We use the quality attributes of the ISO 9216 [4] and assume that these completely categorize all relevant aspects of an IT systems quality.

Assets are parts of the system that are valuable for the organization, e.g. information, software, and hardware. They need to be protected from malicious activities in order to achieve business goals.

Value quantifies the criticality of each quality goal to the business and prioritizes the quality goals against each other. It is determined by the *impact* the compromise of an asset would cause to the business.

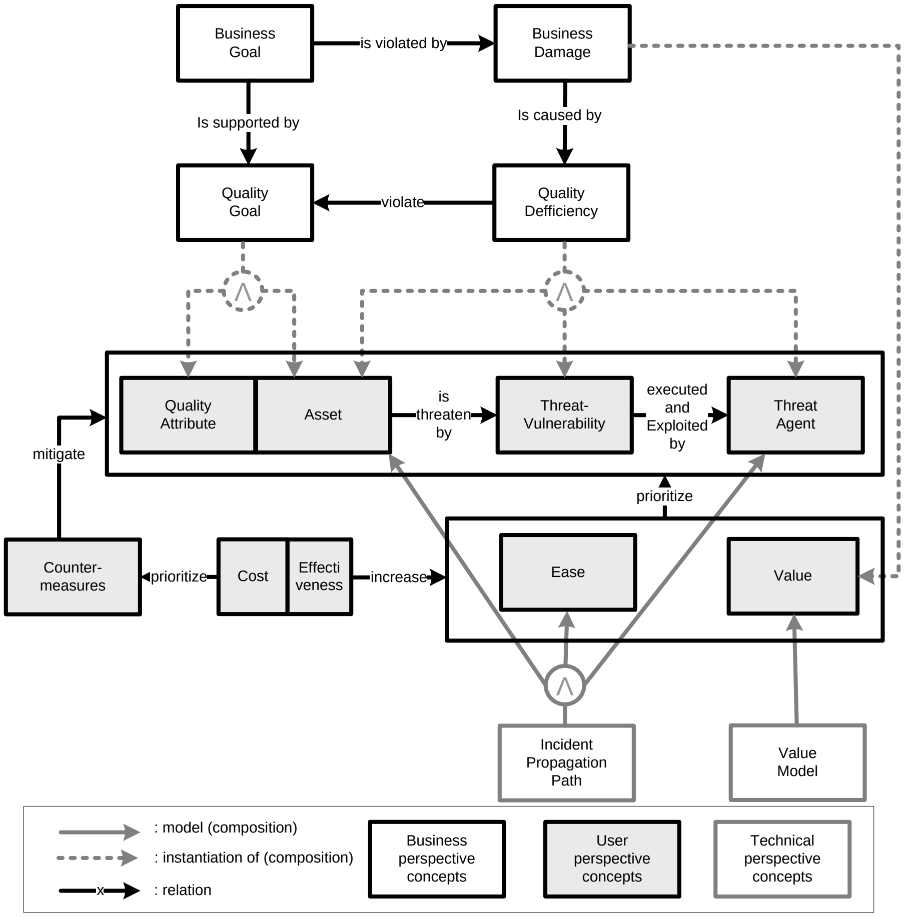

Fig. 1. Meta-model showing the concepts and their interrelations.

Quality deficiency is a lack of quality attribute for an asset (a flaw or a side-effect of an otherwise wanted roperty) that *violates* quality goals and *causes* a business damage, e.g. confidentiality problems cause loss of trust of the users.

Threat agent is a person, i.e. an insider or an outsider, that intentionally or unintentionally executes a threat, e.g. an employee at the outsourced data center. A threat agent can be characterized in terms of his motivation, goal and attribute, e.g. disgruntled employee.

Threats are actions which cause a quality deficiency that degrades the satisfaction of a quality goal, e.g. data theft threatens the confidentiality of data. A threat is often made possible or more probable by a vulnerability.

Vulnerabilities are a property of the assets or the IT system or it's environment that can be exploited by threat agents who execute a threat. A vulnerability can be misused w.r.t. a quality goal. Identifying the vulnerabilities and determining the assets that are threatened by them help analysts determine the effectiveness of countermeasures that mitigate them. Vulnerabilities can be "lack of technical change management" or also wanted properties of the system like "Single-Sign On", if they can enhance the ease of execution of a threat.

Misuse Cases (MCs) describe as scenarios how a threat agent may cause a quality deficiency. The MC takes the perspective of the user and describes what happens at the interface between user and system. The MCs are prioritized based on the *ease* which they can be executed with and the *impact* which they cause to the asset(s).

Incident Propagation Paths (IPPs) are descriptions of MC using the technical perspective. In some cases, an IPP consists of several interconnected steps. That is a threat agent causes a quality deficiency on an asset by executing one or many threats, which exploit vulnerabilities of several assets. Such IPP scenarios are important for humans to imagine the flow of events including causes and consequences and support creativity. Like the MCs, the IPPs are prioritized based on their ease of execution and the impact caused. It is possible that there are several IPPs realizing the same MC. Then, ease of the easiest IPP is the ease of the MC.

Ease of an attack is determined by the effort needed for exploiting the hardest vulnerability of the asset for a threat agent. This effort correlates to the likelihood that this threat is executed.

Countermeasures are mitigation, detection or prevention mechanisms. They partly or completely counteract a threat-vulnerability pair or the threat agent, and transform the asset that they apply to into a more secure asset. The countermeasures specify quality requirements on the IT system.

Cost of a countermeasure quantifies all costs of a countermeasure, including implementation cost and cost of ownership. Depending on the depth of the assessment we either use partially ordered scale or the real costs. In case the real costs are used then the risk expert may calculate the implementation cost based on required man hours and average salary per hour.

Effectiveness of a countermeasure is given by the expected risk reduction it achieves. Most countermeasures either influence impact or ease of an MC respectively IPP.

5 RiskREP Method and it's Application

In this section, we describe the steps of the RiskREP method and the activities associated with each step. Furthermore, to illustrate how each step can be applied in practice, we present a running example based on the case that we describe in Section 3. We suppose that the security requirements elicitation can build upon the written specifications of the systems functionalities e.g. in the form of use cases and IT architecture. Due to the dynamic nature of the IT systems, security requirements elicitation and prioritization is an iterative process. This process is executed in five steps:

Step 1: Finding quality goals;
Step 2: Analyzing security risks;
Step 3: Defining countermeasures;
Step 4: Prioritizing countermeasures; and
Step 5: Applying countermeasures and re-starting the cycle from Step 2.

The information that the RiskREP method uses is elicited from three stakeholder categories, i.e. business owner, IT manager and security officer. Two further stakeholders are the RE expert and the risk expert. These experts elicit the necessary information from semi-structural interviews with the other stakeholders and execute the RiskREP method. In our case study, the experts are two authors of this paper.

5.1 Step 1: Finding Quality goals

RiskREP begins with identifying the business goals (BG). For this, the RE expert asks the business owners to define their goals. After identifying the BGs, RE expert defines business damages (BD) by estimating what may violate the achievement of the business goals. Then, she identifies quality deficiencies (QD) that may cause a BD. For this, she analyzes quality-related deficiencies of the IT system in the context that it will be used. Once the QDs are identified, she derives quality attributes (QA) that need to be protected from the QDs. Finally, she drives quality goals (QG) from QAs by relating affected assets.

Running Example 1: Finding Quality goals

The graph in Fig. 2 plots the connections between the security-related business perspective concepts of the system described in Section 3.

The BGs had been defined by the management, before we started our case study. Therefore, we could extract them from a project report. In this context,

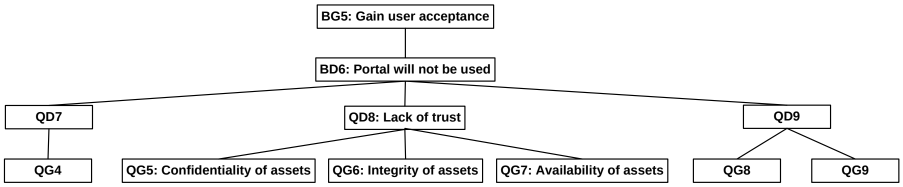
BG5: Gain user acceptance
BD6: Portal will not be used
QD7
QD8: Lack of trust
QD9
QG4
QG5: Confidentiality of assets
QG6: Integrity of assets
QG7: Availability of assets
QG8
QG9

Fig. 2. The down way of eliciting business perspective concepts of RiskREP.

the management is composed of the project management (i.e. the responsible professor and one of her staff members), a committee being responsible for the use of information technology in teaching and the vice president for teaching and studies.

In the project report there is only one security related business goal *BG5: Gain user acceptance*. Starting with this BG we constructed the goal and damage graph (see Fig. 2) by connecting it to the BGs that threaten it. The BG is related to one BD: *Portal will not be used (BD6)*. Then, the RE expert identified three quality deficiencies that may cause BD6, i.e. *User unfriendliness (QD7)*, *Lack of trust (QD8)*, and *Lack of added value (QD9)*. Because of the scope of our case study we analyzed only QD8 further. Lack of QAs confidentiality, integrity and availability may lead to the QD8. Accordingly, the expert derived three high level quality goals, i.e. *Confidentiality of assets (QG5)*, *Integrity of assets (QG6)*, and *Availability of assets (QG7)*. These high level QGs are instantiated at the user perspective.

5.2 Step 2: Analyzing Security Risks

The aim of this step is to analyze the security risks related to each QG. The risk expert starts by identifying possible misuse cases (MC) that may threaten the QGs and then estimates their ease and impact. For identifying the MCs, she brainstorms together with the security officer based on a list of possible threat agents, threats and vulnerabilities. She furthermore analyzes documents delivered by the IT manager (e.g. IT architectural drawings and system specifications), lists relevant information assets and IT assets and forms IPPs.

For estimating the ease, the risk expert together with the security officer first identify the vulnerabilities of the IT assets and the threat that exploits them, as well as estimate the threat agents that may execute them, and their motivation. Then they estimate how incidents might propagate. We model different ways an incident might propagate with IPPs. The risk expert forms IPPs based on a structured thinking process. That is she first draws the assets representing the entry points of the system. Then, gradually connects further assets under consideration of physical and logical connections among the assets and presence of vulnerability-threat pairs associated with the destination component. We consider an IPP complete when the asset, that the MC addresses, is reached. Finally, the risk expert calculates how easy it is for each threat agent to accomplish the

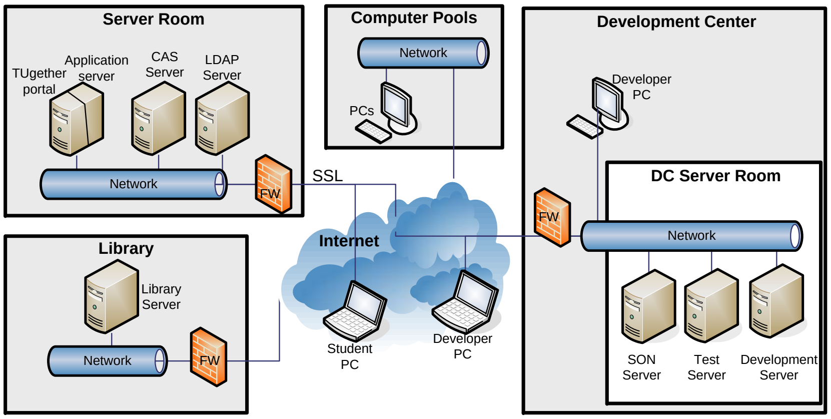

Fig. 3. IT architecture of the student portal. (FW: Firewall, DC: Data Center, CAS: Central Authentication Service, and SON: Personal Development Server)

IPPs. We call this the *ease* of an IPP, which equals to the ease of the least difficult propagation on the IPP. We furthermore set the ease of each propagation equal to exploiting the easiest vulnerability-threat pair of the destination asset. For less complicated systems it is possible to chose a more practical approach and guess the ease of each MCs without forming IPPs. In this case, the vulnerabilities are linked to MCs.

Finally, she assesses the *value* of each QG by following *value models*, e.g. TD model [18] for availability, DCRA model [12] for confidentiality. These values depend on the related BGs and the degree at which each QG contributes to the satisfaction of a BG. These values are the basis for estimating the *impact* or damage caused by the MC respectively IPP to these QGs.

Running Example 2: Analyzing Security Risks

One of the MCs that threatens QG6 is *Manipulation of account data (MC5)*. The vulnerabilities, threats and threat agents that we used result from vulnerability and threat check-lists that we put together during previous case studies and by literature research. For this case, the risk expert agreed on with the security officer to use five threat agents, i.e. *user, hacker, portal admin, portal developer* and *service developer*.

Then, the risk expert and the IT manager draw the IT architecture of the system in scope (see Fig. 3) based on which they conducted a list of information assets and a list of IT assets to be protected. The IT assets of the TUgether portal related with this MC5 are *TUgether portal server, LDAP server* and *Development server*.

Finally, the IT manager estimated the impact of each MC based on the information assets that might be retrieved from the MC related IT assets. At this case using more sophisticated value models was not necessary. Therefore, we

Table 1. MCs and their attributes.

MC ID	**risk (ease,impact)**	**Threat agent**	**Threat**	**Vulnerability**
MC1: non-compliant modification of an included service	(2, 2)	service developer	modifies service non-compliant to standard	Portal does not manage data and therefore data synchronization between portal and services is necessary
...	...	...	...	...
MC5: manipulation of account data	(1.5,1)	hacker	data get lost or are manipulated during transfer	Portal does not manage data and therefore data synchronization between portal and services is necessary
...	...	...	...	...
MC9: no logout in computer pool allows others to use this account	(1,3)	user	does not log out after having used the portal on a computer in the public computer pool	no access control to computer pools

estimated the impact of MC intuitively and in a scale from 1 (low) to 3 (high). For instance, the impact of MC5 is *1.5.* Account data is stored outside the portal and is transported to the portal when needed, so a hacker might manipulate it when it is transferred.

Furthermore, although we have discussed IPPs when specifying the MCs, because IPPs are self-evident in this not too complex system architecture we did not specify them in this case. We estimated the ease of each MC intuitively on a scale from 1 (low) to 3 (high). This estimation demanded knowledge about technical infrastructure and context of use.

In total, related to QG6, we identified ten MCs. Two of which are presented in Table 1. The table shows which threat agent, vulnerability, and threat combination constitutes each MC and plots their risk. For instance, *Non-compliant modification of an included service (MC1)* might be realized by only one threat agent, i.e. service developer, who develops the connected services.

5.3 Step 3: Defining Countermeasures

At this step, we conduct a list of countermeasures for each MC. The countermeasures either completely or partially mitigate or detect either a threat or a vulnerability, or act against a vulnerability. The stakeholders involved at this step are the security officers, who knows effects of countermeasures on threat and vulnerability pairs, and the RE expert.

We first composed a set of *countermeasures* by extracting countermeasures from existing check-lists. These check-lists are part of RiskREP and contain general countermeasures for 167 threat-vulnerability pairs. In this step of RiskREP, one brings these general measures to a concrete, realizable level by specifying to which component it applies and how. We determine which countermeasure can mitigate, prevent, or detect which MCs (and to which level) by refereing to the

Table 2. Countermeasures for mitigating MCs, example from the case study.

	Cost	**Misuse Cases**				
		MC 1	MC 2	...	MC 9	MC 10
C1: standardized interfaces (LDAP, CMS,...)	2	mitigates				
C2: timeout and login of user	1				partially mitigates	
...		...	...	...	...	...
C10: security measures taken by the included services	0				partially mitigates	

threats and vulnerabilities of MCs. There are n-m-relationships among MCs and countermeasures, which are best presented in a table. Finally based on the cost of implementation and of ownership of each countermeasure we quantify the cost of each countermeasure.

Running Example 3: Defining Countermeasures

Table 2 shows the results of this step for the portal. Here, we quantified the cost of each countermeasure on a scale of 0 to 3 points, where 0 stands for no cost, 1 for the cost of changing the settings of applications, 2 for the cost of installing and maintaining freely available countermeasures and 3 the cost of for purchasing, installing and maintaining countermeasures.

5.4 Step 4: Prioritizing Countermeasures

At this step, we prioritize the MCs and the countermeasures. We prioritize the MCs based on their *risk*, whereas we prioritize countermeasures based on their *added value*, i.e. effectiveness and cost. Since a countermeasure's added value is created by reducing MC risk, we approximate the value it adds based on the ease reduction and the impact reduction it achieves and the costs it causes. The risk reduction (effectiveness) is estimated by imagining the system with the countermeasure applied and without. The added value is the higher, the more ease and impact are reduced and the lower the countermeasure cost.

Ease and impact, as well as effectiveness and cost are incomparable entities. Thus, we do not add, multiply or subtract them from each other, like many other authors do. Instead, we say that the risk of an MC mc_i is superior to the risk of another MC mc_j if both ease and impact of mc_i are superior to the ease and impact of mc_j; and the added value of a countermeasure c_i is superior to the added value of another countermeasure c_j if risk reduction by c_i is higher than the risk reduction by c_j and/or the cost of c_i is lower than c_j's cost. In case the ease of mc_i and the impact of mc_j are superior (or vice versa), then we consult the stakeholders' opinion to determine the superior MC. Similar applies to countermeasures.

By applying countermeasures on MCs, we reduce the risk. However, applying countermeasures usually means increased spending. Therefore, RiskREP aims at finding the ideal set of countermeasures to be applied in addition to the countermeasures that are implemented in the current system. The best set of countermeasures is the set of not yet implemented countermeasures with minimum total cost and maximum risk reduction. These values can be optimized execratively testing several sets of countermeasures. The security budget of the system is the main delimiter for the ideal set of countermeasures.

Countermeasures interact with each other. Some need to be implemented together or some can replace each other or reduce the effectiveness of another countermeasure. Therefore, RiskREP also estimates the direction of these interactions in order to identify the ideal set of countermeasures to be implemented. We call the effectiveness of a set of countermeasures when applied together the *combined effect* of that set of countermeasures. For determining the combined effect of two countermeasures, we interview the security officer. We furthermore address the combined effects of more then two countermeasures by flattening them into pairs of countermeasures. That is, assuming that we have three countermeasures c_1, c_2 and c_3, we argue that the combined effect of applying c_1, c_2 and c_3 together equals to adding the combined effect of c_1 and c_2 with the combined effects of c_2 and c_3, and of c_1 and c_3.

For using the predicted effects of countermeasure interactions, we not only need the current system as a reference system, but also a vision of the system to be implemented. Vision of the system contains the countermeasures that are foreseen for implementation. Supported by an automated tool, different sets of countermeasures can tentatively be foreseen for implementation and the value added by this set of countermeasures can be calculated and optimized.

Running Example 4: Prioritizing Countermeasures

In the case study, we used the simplest scales for cost, ease and impact, i.e. -1, 0, or +1. This way it is easy to estimate and less prone to mistakes. If necessary, RiskREP allows using more sophisticated scales. We furthermore used a shorthand notation for quantifying a countermeasure's added value, i.e. cost ease impact. For instance, C1's added value is indicated with -00, i.e. it reduces cost, but does not influence ease and impact of integrity-related MCs, on the other hand C10's added value is 0-0, i.e. it is cost neutral and reduces ease of some MCs but does not affect its impact. We furthermore defined a countermeasure's effectiveness as follows: if a countermeasure affects neither impact nor ease of an MC, then it's effectiveness is 0; if it decreases either impact or ease, then it's effectiveness is 1; if it decreases both, it is 2. For those which influence both we approximate the effectiveness as follows: - - counts as effectiveness = -2; + + counts as +2; and + - or - + counts as 1.

Table 3 shows the combined effects of countermeasures that the security officer estimated for TUgether. The table is sparse. In the case study, it contains 10 interactions, while among the 10 countermeasures 90 would be possible.

For determining which countermeasures should be implemented next, i.e. to prioritize them, we applied a heuristic approach using categories of MC risks and countermeasures added values. Here we used spreadsheets.

Table 3. Combined effects of countermeasures.

Countermeasure	C1	C2	...	C10
C1			...	
C2			...	
...	...	...	...	...
C10	- - 0		...	

When prioritizing the MCs according to their risk, i.e. ease and impact, we want to distinguish between those which have low ease and cause high damage and vice versa. Therefore, we use the following categories:

- **ignore:** ease and impact are low;
- **rare, but detrimental:** ease is low, but impact is high;
- **frequent, but harmless:** ease is high, but impact is low;
- **catastrophic:** both are high, or one is average and the other high; and
- **average:** both are average, or one is average and the other low.

For categorizing the countermeasures effects, we chose four categories, based on effectiveness and cost. Since countermeasures either increase the ease or reduce the impact, or both, we built our categories based on these changes. These categories are:

- **contra-effective:** both ease and impact increases or reduces simultaneously;
- **ease increase:** ease increases, but impact remains the same;
- **impact reduction:** impact reduced, but ease remain the same;
- **counter-effective:** ease increases as impact reduces;
- **low hanging fruit:** cost is 0 and either only ease increases or only impact reduces or while ease increases impact reduces;
- **cost-efficient:** cost is 1 and either only ease increases or only impact reduces or while ease increases impact reduces;
- **cost-effective:** cost is 2 and while ease increases impact reduces; and
- **expensive:** cost is 2 and either only ease increases or only impact reduces.

To note that for deciding to which category a countermeasure belongs to, we proceed in the above given order of categories.

For choosing the optimal set of countermeasures, we did not use a formula which optimizes the systems added value automatically, but rather decided for a countermeasure selection strategy together with the stakeholders. In this case the strategy is on countermeasure effectiveness and cost. Accordingly we suggested the stakeholder to implementing all “low hanging fruit” countermeasures. Furthermore, since defining the categories also influences the strategy, we asked for stakeholders’ approve after defining them. This way of choosing the countermeasures to be implemented is a heuristical one which allows to make decisions transparently and based on objective criteria, but still is simple and easy to execute.

5.5 Step 5: Re-starting the Cycle

After having applied the countermeasures, the requirements elicitation and prioritization process starts from Step 2 with the updated set of countermeasures and changed IT-architecture.

6 Validation

The case study showed that RiskREP has a systematic process which guides the security-related analysis of the system current system and requirements on it's future versions. One important observation is that the functional requirements and the system architecture are the necessary precondition for the RiskREP analysis. The permitted actions of users, administrators and developers and how data are exchanged between the system components must be known. This knowledge is the basis for analyzing where data can be lost, manipulated or disclosed to unauthorized persons.

It took us four hours for jointly analyzing and prioritizing the integrity requirements. Considering the large amount of information gathered during this time, we consider RiskREP to be an efficient and effective method.

In this case study, we could not observe whether RiskREP indeed helps to separate the communication with business owner, IT manager and security officer, because our contact person could cover all these perspectives. He is project manager and developer equally. We could find most of the information needed for step 1 in the projects report which is written from a management perspective by the project manager and the management above him. This shows that step 1 in fact models the information which is relevant for management.

RiskREP helps to structure the discussion. The templates and check-lists helped to not forget anything important. Our contact person said that the scenarios were very helpful for the analysis, and the analysis gave them new ideas, while all the results of their former discussions were found by the RiskREP analysis also. The case study was supported by simple tools: drawing tools for the tree graphic produced in step 1 and for presenting the system architecture, several spreadsheet tables for the qualitative and quantitative analysis of MCs and countermeasures. These tables also support the testing of different sets of countermeasures.

7 Related Work

In this section, we compare widely known RE and RA methods. Table 4 presents an overview of this comparison. By developing RiskREP we aimed to have a method that satisfies all of the features used in this comparison.

The importance of considering security at requirements elicitation and prioritization phase is accounted by many researchers [1, 2, 7, 17]. To systematically elicit security requirements Elahi et al. [7] and Stamatis [17] propose to derive requirements from high level goals. This is especially important for the completeness of the requirements and applicability of the method. However, a requirements elicitation method should differentiate between business goals and quality (security) goals. Despite the fact that most of the approaches [2,7,10,13] in Table 4 differentiate between functional and non-functional goals, none of them differentiate between business and quality goals.

As security became a concern for the requirements engineers the recently developed methods, e.g. [2,7,9,16], consider both intentional uses and misuses of system components. However, security requirements elicitation process requires expertise of stakeholders from different fields. Although most of the approaches that we compare acknowledges this challenge, only a few [5,9,10,16] express how different stakeholder views can be considered by eliciting information.

Due to the limited security budget only a subset of the identified requirements can be satisfied. Determining the most effective subset requires systematic estimating and quantifying risks, considering different effectiveness levels of requirements and the trade-off among them, as well as their costs and effectiveness. However, only some RE methods assess risk, like FMEA [17], Tropos based approaches [3, 7], GSRM IsHo10, Attack Graphs [14], or extended KAOS [2]. Stamatis [17] assesses the effects of failure and failure occurrence frequencies. Tropos based approaches [3, 7] analyze incident likelihood based on level of evidence that supports or prevents the occurrence of security events. GSRM IsHo10 estimates likelihood and impact of risk events on a scale of low/ medium/ high. Furthermore, Phillips and Swiler [14] provide a method that allows quantifying incident likelihood according to probability, average time or cost/effort. Lamsweerde et al. [2] also quantifies the incident likelihood, but he aims to use this quantification for determining the necessary granularity of requirements to be elicited.

The methods that consider effectiveness levels of requirements refer to different attributes of the IT system that is analyzed. Elahi et al [7] differentiate among three levels according whether the countermeasure alleviates the effects of vulnerabilities, patches them or prevents malicious tasks. Goal-Risk Model [3] differentiates between four levels based on contribution relations between security events and goals. Finally, FMEA [17] differentiates according to incident detection rate.

As we discussed in Section 5, when applied together, requirements may contradict with each other or support each other. Elahi et al [7], NFR framework [13], and Asnar and Giorgini [3] consider these combined effects and prioritize the system requirements accordingly. ATAM [10] also considers how countermeasures affect each other and refer to it as “tradeoff points”.

8 Conclusion and Future Work

This publication presents RiskREP, a new method for the systematic elicitation and prioritization of security (quality) requirements. It has been constructed by integrating the methods MOQARE and CRAC++. We have applied it to a web portal in order to assess the portal’s security and to identify potential security improvement measures. The precondition for such an analysis is a model of the system architecture and the specification of it's functional requirements.

RiskREP has the following features of requirements elicitation: systematic process; differentiation between business and quality goals; considering both intentional use and misuse; and considering different stakeholder views. Furthermore the following features for requirements prioritization: systematic estimation of asset value and incident likelihood; requirements prioritization based on costs; considering requirements' effectiveness; and considering combined effects of requirements. These features showed to have positive effects on the analysis. The strength of RiskREP are: step-by-step guidance of the analysis; check-lists of threats, but also case study specific lists of system agents, system components and use cases support the results to be more complete than mere brainstorming results; time-efficient analysis; and transparent prioritization of security requirements. This is our first case study with this new method. As the case study is not too complex, RiskREP could be simplified in several steps. In the future, more complex cases shall be analyzed, in order to investigate the method's scaleability. Currently, we conduct our analysis by the support of a set of connected spread sheets. To increase the usability of the method, we are planning to provide more specific tool support.

Security requirements can be used to derive test cases for security analysis and compliance monitoring. RiskREPs countermeasures describe what the system shall do and therefore can be used as test criteria. The MCs and IPPs describe misuse scenarios from the user perspective or the technical perspective respectively. These scenarios end in a system misuse and some sort of damage, when a threat is executed. When the countermeasures are effective, they prevent this damage or reduce its ease or the damage caused. Consequently, the MCs can be used as test cases for security-related black box tests and the IPPs as test cases for white box tests. Measuring ease and damage also is important in order to verify whether the implemented countermeasure has the effect which had been expected. The test cases priorities are related to the risks of the corresponding MC or IPP: the higher its risk, the more important it is to test a scenario. In future work, we want derive security test cases and monitoring criteria from MCs and IPPs, in order to see how easy and straightforward this can be done and whether these test cases make sense for security testing and monitoring.

References

1. R. J. Ellison A. P. Moore and R. C. Linger. Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, 2001.
2. R. De Landtsheer A. van Lamsweerde, S. Brohez and D. Janssens. From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering. In *Proc. of RHAS Workshop*, pages 49–56. Essener Informatik Beitraege, Bd.6, 2003.
3. Y. Asnar and P. Giorgini. Modelling risk and identifying countermeasures in organizations. In *CRITIS'06: Proc. of 1st Int. Workshop on Critical Information Infrastructures Security*, pages 55–66. Springer, 2006.

4. International Electrotechnical Commission. ISO (Int. Standards Org.): Int. Standard ISO/IEC 9126, Information technology - Software product evaluation - Quality characteristics and guidelines for their use.
5. F. den Braber, T. Dimitrakos, B.A. Gran, M.S. Lund, K. Stølen, and J. Aagedal. The CORAS methodology: model-based risk assessment using UML and UP. pages 332–357, 2003.
6. E. Dubois, P. Heymans, N. Mayer, and R. Matulevicius. A systematic approach to define the domain of information system security risk management. In S. Nurcan et al., editor, *Intentional Perspectives on Information Systems Engineering*, pages 289 – 306. Springer-Verlag, 2010.
7. G. Elahi, E. Yu, and N. Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requirements Engineering*, 15(1):41–62, 2010.
8. A. Herrmann and B. Paech. MOQARE: misuse-oriented quality requirements engineering. *Requir. Eng.*, 13(1):73–86, 2008.
9. S. Islam and S.H. Houmb. Integrating risk management activities into requirements engineering. In *RCIS'10: Proc. of the 4th Int. Conf., on Research Challenges in Information Science*. IEEE, 2010.
10. R. Kazman, M. Klein, P. Clements, and N.L. Compton. Atam: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, 2000.
11. A. Morali and R. J. Wieringa. Confidentiality Requirements Engineering for Outsourced IT-Systems. In *RE'10: Proc. of the 18th IEEE Int. Requirements Engineering Conf.* IEEE Computer Society Press, 2010.
12. A. Morali, E. Zambon, S. Etalle, and P. Overbeek. It confidentiality risk assessment for an architecture-based approach. In *BDIM'08: Proc. of 3rd IEEE Int. Workshop on Business-Driven IT Management*, pages 31–40. IEEE Computer Society Press, 2008.
13. J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. *IEEE Software*, 18:92–96, 2001.
14. C. Phillips and L.P. Swiler. A graph-based system for network-vulnerability analysis. In *NSPW '98: Proc. of the 1998 workshop on New security paradigms*, pages 71–79. ACM, 1998.
15. G. Sindre and A.L. Opdahl. Templates for misuse case description. In *REFSQ'01: Proc. of the 7th Int. Workshop on Requirements Engineering: Foundation of Software Quality*, pages 125–136. Essener Informatik Beitraege, Bd.6, 2001.
16. G. Sindre and A.L. Opdahl. Eliciting security requirements with misuse cases. *Requir. Eng.*, 10(1):34–44, 2005.
17. D.H. Stamatis. Failure mode and effect analysisFMEA from theory to execution. 2003.
18. E. Zambon, D. Bolzoni, S. Etalle, and M. Salvato. Model-based mitigation of availability risks. In *Second IEEE/IFIP Int. Workshop on Business-Driven IT Management*, pages 75–83. IEEE Computer Society Press, 2007.

Table 4. Comparison of RiskRep with widely known RE and RA methods. Please note that we do not present RiskREP as a column in this table because RiskREP supports all of these features.

	Elahi, Yu and Zannone [7]	Misuse Cases [16]	extended KAOS [2]	ATAM [10]	NFR framework [13]	FMEA [17]	Attack Graphs [14]	CORAS [5]	Goal-Risk Model [3]	Goal-driven Software development Risk Management model (GSRM) [9]
Requirements elicitation										
Systematic process	drives requirements (soft-goals) from goals	no	no	no	drives requirements (soft-goals) from goals	yes	no	no	no	yes
Differentiation between business and quality goals	goals and soft-goals	no	functional goals and non-functional goals	yes	technical objective and business objective	no	no	no	strategic layer, event layer and treatment layer	project goals and sub-goals
Considering both intentional use and misuse	intentional use and misuse	use cases and misuse cases	goal and unti-goal	no	no	no	no	no	no	risk events and tasks
Considering different Stakeholder views	no	yes	no	yes	no	no	no	yes	no	yes: user representative, business analyst, requirements engineer, risk manager
Requirements prioritization										
Systematic estimation of impact	no	no	no	no	no	failure effect	no	depends on selected model	no	risk impact
Systematic estimation of incident likelihood	level of evidence	no	for determining the granularity	no	no	occurrence of failure	probability, average time or cost/effort	depends on the model	level of evidence	risk likelihood
Prioritization based on monetary costs of requirements	no	real cost	no	volume of change	no	no	financial loss or loss of system	no	yes	no
Considering effectiveness levels of requirements	3 level differentiation	no	no	no	no	detection rate	no	no	contribution relation	effectiveness
Considering combined effects of requirements	between soft-goals	no	no	trade-of points	between soft goals	no	no	no	between goals	no