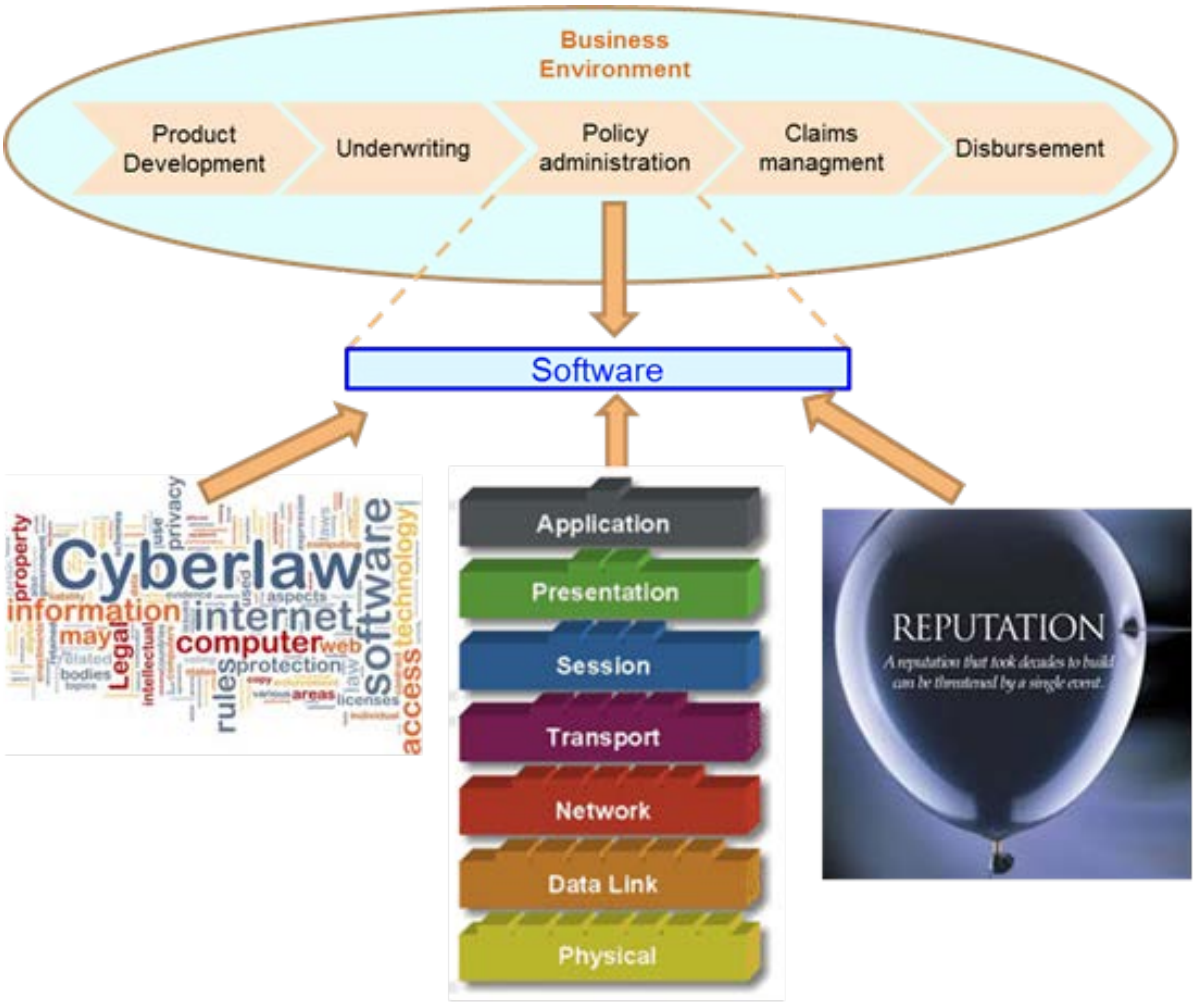
The Influence of Software Vulnerabilities on Business Risks



Four sources of risks relevant for evaluating the influence of software on business risks

D.H.H. Kramer MSc RE

The Influence of Software Vulnerabilities on Business Risks

Hilbrand Kramer

Submitted as part of the requirements for the award of the
MSc in Information Security
at Royal Holloway, University of London.

25 August 2014

Table of content

1 Executive Summary

A manager receives a report from a penetration test on his application indicating a couple of vulnerabilities that are rated as “high risk.” The manager asks you, “What does it really mean for *me*?” Software vulnerabilities are stated in rather technical language, and the manager may not be able to recognize the problem.

This project reviewed literature about secure software development, comparing how software risks are prioritised using a measure of induced business risks. Three commonly-known methodologies, “The Security Development Lifecycle” of Microsoft [14], “Building Security In” of G. McGraw [18], and the ISO/IEC 27034: “Secure techniques – Application Security” [2, 5, 6]. are reviewed in Chapter 3. These methods present and rank the priority of software vulnerabilities in a rather technical way that is not linked with the business processes of the business manager.

Chapter 4 presents the results of the review of material used in the field of detecting computer-related risks in the broad sense. The literature is reviewed to provide an overview of relevant topics to identify threats, likelihood, vulnerabilities and risks.

During the review of the material presented in Chapter 5, four sources of risks that are relevant for evaluating software are determined:

1. Operational risk related to business processes
2. Regulatory risk resulting from non-compliance to the applicable laws and regulations
3. Reputational risk related to the brand of the organization
4. Information Technology (IT) generic risk stemming from the non-compliance of software with technical standards, reducing the overall information security

The project selected the following frameworks to manage information security risk: COBIT5 [3, 4, 15], IRAM [16], ISO 27005 [1] and SABSA [17]. The book *No Excuses:* A *Business Process Approach to Managing Operational Risk* [11] is reviewed to provide material about risks and the processes from which they originate.

Based on a review of the literature, a model is developed to link software vulnerabilities to business risk that consists of four parts:

First, the environment in which the software vulnerability is discovered should be thoroughly understood.

Second, the probability of a threat exploiting the software vulnerability must be assessed. A Threat Score Model is developed to assess the likelihood of a threat exploiting the software vulnerability. This model is used to evaluate whether there is a threat agent and what his capability, motivation, catalyst, and the inhibitors and amplifiers are.

Third, the impact of a probability must be assessed. This will be performed with the IRAM method of ISF to relate software vulnerabilities to the first three sources of risk. COBIT5 is selected to relate software risk to the generic IT risk.

Finally, the resulting risk for the broad business environment should be established. The probability of the threat exploiting the vulnerability and the impact assessed are plotted in a 5x5 matrix. This will reveal the business risk and can be used during the discussions to achieve mutual understanding about the risk.

2 Introduction

2.1 Project background

Imagine that you are an Information Risk Manager at an organization, and a manager approaches you with a report from a penetration test on his application. The report states a couple of vulnerabilities that are rated as “high risk.” The manager asks you, “What does it really mean for *me*? Do I have a severe issue?”

This is a situation I encounter in day-to-day practice. Software vulnerabilities are stated in rather technical language, and the manager may not be able to recognize the problem. Even as an Information Risk Manager, it is often rather difficult to present the business risk. Vulnerabilities seen as a significant problem to the penetration testers can be difficult or impossible to relate to business (process) risks.

During the MSc in Information Security, I followed the Software Security module. The topic of one of the lectures was Building Security In [12]. The method presented used threat modelling to determine the threats of software. Once the threats are determined, they must be ranked based on the risk they expose [12, slide 30].
I noticed that, in my opinion, the methods used for ranking vulnerabilities lack a necessary link to business (process) risks.

2.2 Project objectives

Business people tend to judge vulnerabilities discovered during software security tests as too technical to be understood. This can have the effect that these vulnerabilities do not receive the priority they could receive if the issues were sufficiently understood. The aim of this project is to:

1. Explore whether my understanding is correct that software security methodology presents and ranks the priority of vulnerabilities in a rather technical way that is not recognised by business managers.
2. Investigate whether methodologies from the field of general computer risks and IT Risk/ Security Governance will contribute to a method of classification of vulnerabilities that is understood by business managers setting the priorities.
3. Establish a methodology that will bridge the gap between the technical vulnerabilities and risks understood by the business.

2.3 Relevance from two different perspectives

Relating software risks to business (process) risks in everyday practice is important from two perspectives:

First, it is important to relate the discovered vulnerability in software with a risk that is recognized by business managers. When these managers do not understand the risk of the vulnerabilities, they will not recognise the need to mitigate these vulnerabilities. They might mitigate the vulnerabilities, but only because the IT security staff say so, but may also waste scarce resources on low business risks that are presented to them as “high risk” vulnerabilities.

Second, developing software with the knowledge of the business risk can lead to more secure software. Developers should be able to analyze the threats and analyze the vulnerabilities to be avoided for the software to be built, when they have an insight about what the business (process) risks are for this business application. During the software security testing process, this knowledge of business risk and threats can also be used to test the functionality with the highest risk.

It was not reasonable to develop a method for both perspectives within the time constraints of this project and the permitted length of this thesis. The aim of this project is to address the first perspective, to relate discovered vulnerabilities to risks recognized by business process managers.

2.4 Methodology

The project can be described as a review of the existing literature, combining existing methods to establish a practical method to rank software risk, and a case study. It is based on existing literature and the practical knowledge that has been obtained by the author of this thesis in his working environment. The project consists of the following phases:

- Based on the observations of the author of this thesis during the Software Security module lecture “Building Security In” and observations in day-to-day practice as an Information Risk Manager, the issue of prioritizing software vulnerabilities was recognized.

- A preliminary review of material about the subject on internet forums, internet sites, published reports and articles, and Information Risk/ Security standards is executed.

- The objective of the project is described in the project description form, which presents the objectives and a preliminary outline of the resources and information to be used.

- The conceptual model will involve using knowledge from the fields of secure software development, computer-related risk analysis, and generally-accepted Information Security Risk frameworks to establish a method to rank software vulnerabilities based on their business (process) risk. To achieve this objective, three questions are defined for the project:
    - Do the existing methods for secure software development relate software vulnerabilities to business risks?
    - Can literature about computer-related risks provide input to relate software vulnerabilities to business risks?
    - Do generally-accepted frameworks to manage Information Security Risk provide applicable knowledge to relate software vulnerabilities to business risks?

- Desk-based research is performed to assess the applicability of the relevant literature and frameworks for the conceptual model.

- Based on the observations from the desk-based research, a model is presented with a case study to validate the applicability of the model.

- The results of the research and the case study are presented in this thesis.

2.5 Adding value

Existing knowledge is applied to demonstrate information security skills by exploring the material of lectures during the Master of Science in Information Security. An overview of the areas in which this thesis adds value:

- Augment and extend the subject of ranking software vulnerabilities presented during lectures from the “Building Security In” of the Software Security module, a method to rank vulnerabilities based on their business risk.

- Software security vulnerabilities in most methods are ranked on their technical risk. The ranking stems from a system in which risks are ranked based on technical parameters. The method presented in this project provides a method to relate software vulnerabilities to business risks.

- The method proposed in this thesis, provides a practical tool to assess threats of discovered vulnerabilities and makes the process visible. This can be used as a basis to discuss software vulnerabilities with business managers.

2.6 Definitions

This project will use the following definitions according to NIST [21]:

- A **threat** is any circumstance or event with the potential to adversely impact organizational operations and assets, individuals, other organizations [21, pp. 8].

- A **threat source** is characterized as [21, pp. 8].:
  (i) the intent and method targeted at the exploitation of a vulnerability; or
  (ii) a situation and method that may accidentally exploit a vulnerability

- A **vulnerability** is a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source [21, pp 9].

- The **likelihood** of occurrence is a weighted risk factor based on an analysis of the probability that a given threat is capable of exploiting a given vulnerability (or set of vulnerabilities) [21, pp. 10].

- The level of **impact** from a threat event is the magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability [21, pp. 11].

- **Risk** is a measure of the extent to which an entity is threatened by a potential circumstance or event, and is typically a *function of* [21, pp.6]:
  (i) the adverse *impacts* that would arise if the circumstance or event occurs; and
  (ii) the *likelihood* of occurrence.

The Figure below shows the risk components and how they are related.

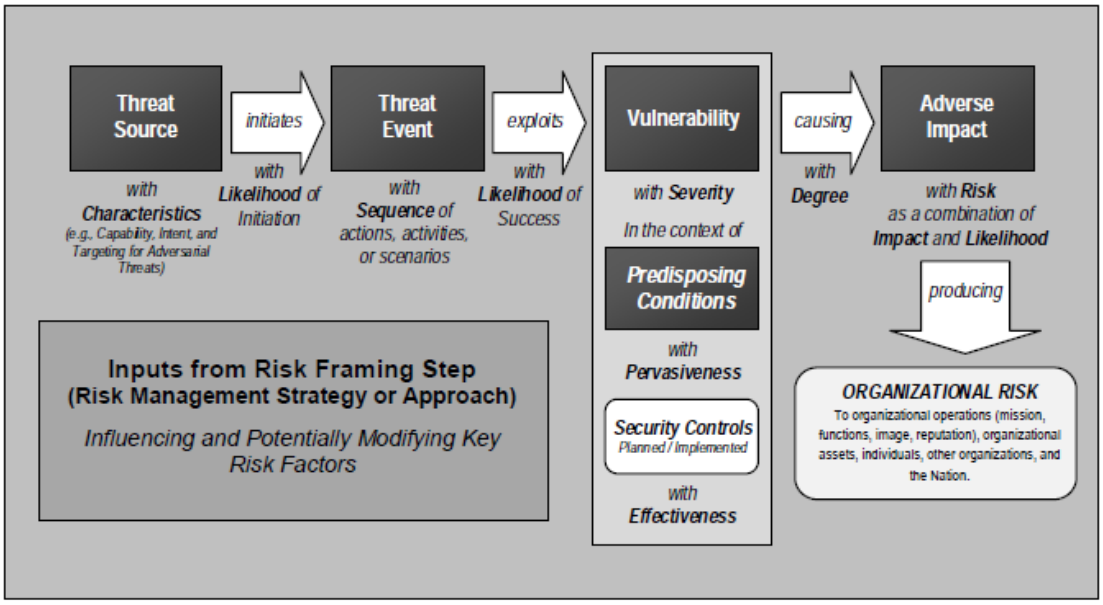


Figure 1: [21, pp. 12], Figure 3: Generic risk model with key risk factors

3 Secure software methods

3.1 Introduction

The internet and the library of the Royal Holloway University of London were searched for relevant material in order to attain an oversight of the material available to develop secure software. The literature was assessed on its potential ability to link software risk with business (process) risk. During the research, many methods were found that describe the software development process from a relatively technical angle. Other material was more focussed on the risk of the software development process for the organization. Other literature described losses or missed goals because a software project fails to deliver the required application within time and budget.

The literature selected is more related to the subject of this project: software (product) security risks and the relation to business risk. Three relevant resources were selected: “The Secure Development Lifecycle” of Microsoft [14], “Building Security In” of G. McGraw [18], and the ISO/IEC 27034: “Secure Techniques – Application Security” [2, 5, 6].

The first resource was selected because it forms the basis of the material of one of the lectures of the Software Security module as part of the MSc in Information Security, and includes a notion of ranking of “risks”.

“Building Security In” was selected because G. McGraw (investigator of NIST) published substantial material about securing software that included a link to business risk. In the book he states: “Only by practicing risk management and factoring in critical business information about impacts will software security escape the realm of the geeks and have an impact on business [18, pp. 79]”. The author of this thesis enthusiastically concurs with him.

ISO/IEC 27034 is a newly-developed framework that states: “Application security protects the critical data computed, used, stored, and transferred by an application as required by the organization” [2, pp. 6]. The application requirements are described in an Organizational Normative Framework. One of the components is the business context that promises a connection of software security with the objectives of the organization. This should be useful to link software vulnerabilities to business risk. During the review of the methods, the focus was on the following parts:

- What the method consists of
- How risk identification/analysis is performed
- How the risk level is captured

In the last paragraph of each method, critical analysis are drawn about the usability of parts of the methods for identifying software risk and relating them to business (process) risk. At the end of the chapter a comparison of the three methods is presented.

3.2 The Security Development Lifecycle (SDL)

On January 15, 2002, Bill Gates issued his memo “Trustworthy Computing” [13, pp. xxiii], which outlined a high-level strategy to deliver a new generation of computer systems, that are more secure and available than those presently used.
Microsoft founded a variety of initiatives to answer the call for more secure software. One of these initiatives was “The Security Development Lifecycle” [14, pp. 53].

3.2.1 The method

The Security Development Lifecycle (SDL): A process for Developing Demonstrably More Secure Software consists of thirteen stages (0-12) [14, pp. 53]:

- Stage 0: Education and Awareness [14, Ch. 5]: *To enable people to understand and be aware of what secure development means.*
- Stage 1: Project Inception [14, Ch. 6]:
  Decide if the project is required to be covered by SDL and if so, which activities need to be undertaken.
- Stage 2: Define and Follow Design Best Practices [14, Ch. 7]
  Best practices about how security goals are achieved. SDL describes Attack Surface Analysis/Reduction (ASR).
- Stage 3: Product Risk Assessment [14, Ch. 8]
  Estimate the effort to build secure software. How much threat modelling, security design reviews, penetration testing, and fuzz testing is needed.
- Stage 4: Risk Analysis [14, Ch. 9]: *See Section 3.2.2*.
- Stage 5: Creating Security Documents, Tools, and Best Practices [14, Ch.10]
- Stage 6: Secure Coding Policies [14, Ch. 11]
  Coding best practices are described for new code and that should be actively analyzed for legacy code.
- Stage 7: Secure Testing Policies [14, Ch. 12]
  Fuzz testing, penetration testing, run-time verification, review of threat model and reevaluation of attack surface.
- Stage 8: The Security Push [14, Ch. 13]
  Primary revisit of legacy code developed before this development cycle.
- Stage 9: The Final Security Review (FSR) [14, Ch. 14]
  Review to verify that SDL is followed during the development lifecycle.
- Stage 10: Security Response Planning [14, Ch. 15]
  A security response process has to be established, to decide which vulnerabilities will be responded to and how to respond.
- Stage 11: Product Release [14, Ch. 16]
  The central security and privacy team, must agree that SDL is sufficiently followed, and the product can be released.
- Stage 12: Security Response Execution [14, Ch. 17]
  Follow-up plan (stage10) to respond to found security vulnerabilities.

For the topic of this project, Stage 2 and particularly Stage 4 are relevant.

3.2.2 How *Microsoft SDL* performs risk identification

Stage 2: Define and Follow Design Best Practices [14, Ch. 7]
Best practices are prescriptive guidelines describing how security goals must be met e.g. Common Secure-Design Principles.
SDL describes Attack Surface Analysis/ Reduction (ASR). ASR focuses on reducing the quantity of code executed by default, restricting who can access code, restricting which identities can access code, and reducing privileges of code.

When e.g. the software requires an extra port on the firewall to be open then the attack surface of the application and the organization will be broadened.

Stage 4: Risk Analysis [14, Ch. 9]
Risk Analysis actually covers what Microsoft refers to as "Threat modeling". A threat is defined as an attacker's objective. The output of this stage is a document describing background information about the application (high-level model), often using Data Flow Diagrams (DFDs), a list of assets that require protection, threats to the system ranked by risk, and an optional list of mitigations.

Threat types are determined using the STRIDE method:

- **S**poofing Identity: attacker to pose as something or someone else
- **T**ampering: malicious modification of data or code
- **R**epudiation: denying having performed an action that other parties can neither confirm nor contradict
- **I**nformation Disclosure: exposure of information to individuals who are not supposed to have access to it
- **D**enial of Service: attack to deny or degrade service to valid users
- **E**levation of privilege: user gains increased capability

Actually this is a list of rather technical threats. To perform a threat analysis for a particular piece of software one would prefer to assess threats that are more linked with the threats of the business process. For example for a payment system: the threat that someone is able to change beneficiary information without the proper authorization.

3.2.3 How *Microsoft SDL* captures the risk level

In the early version of Microsoft's SDL, DREAD was used for characterizing the risk associated with vulnerabilities [24, pp.111]. When using the DREAD method, a threat modeling team calculates security risks by assigning numeric values to each of these five categories:

- **D**amage Potential: Extent of damage occurring if a vulnerability is exploited. *This could be used to provide a notion of impact on business objectives.*
- **R**eproducibility: How often an attempt at exploiting a vulnerability works.
- **E**xploitability: How much effort is required to exploit the vulnerability.
- **A**ffected Users: The ratio of instances of the system that would be affected if an exploit became widely available.
- **D**iscoverability: Likelihood that a vulnerability will be found by external security researchers and hackers if it went unpatched.

Assigning numbers in DREAD is, to some degree, arbitrary [12, slide 35-36]. It appeared that security people rate every risk as high (10), and non-security people tend to rate every risk as low (0-1). As a result most risks were ranked equally as medium risk.

Microsoft decided to retire DREAD from further use and created the bug bar that defines characteristics of a threat and, thereby, the level of risk [14, pp.121]. The risk rankings are derived, in part, from the Security Bulletin Severity Rating System (SBSRS), which defined risk level 1 as highest and 4 as the lowest [12, slide 36 and 19].

3.2.4 Critical analysis of the *Microsoft SDL* method

The Microsoft SDL threat modeling method can be used to discover vulnerabilities in the software. The five categories of the STRIDE methodology actually represent threat types that can lead to vulnerabilities. However these are rather technical threats with no obvious relation with the business process.

The method of drawing DFD's and listing the assets requiring protection is useful when discovered vulnerabilities need to ranked. It provides insight how the software works and what assets need protection.

The SDL methods for capturing the risk level of vulnerability, DREAD and SBSRS are not useful to link software risks with business risks. The only link with a notion of business risk is damage potential in the DREAD method. Risk levels are merely captured from the likelihood that a risk can occur, by evaluating the circumstances or conditions that need to be present. This is the more technical risk.
However, it is questionable whether the "risks" discovered by SDL are a concern of the business manager because the link with business risk is not obvious.

3.3 Software Security: Building Security In

In the introduction of his book, McGraw defines software security as "engineering software so that it continues to function correctly under malicious attack" [18, pp. 3].

The book describes a Three Pillars model of Software Security (see Figure 2) [18, pp. 25-37].

Figure 2:
[18, pp. 26] Software Security: Building Security In, Figure 1-8 The three pillars of software security are risk management, software security touchpoints, and knowledge.

McGraw distinguishes between the notion of tracking and mitigating risks as a full cycle activity in Pillar I and architectural risk analysis (ARA) in Pillar II [18, pp. 26].

3.3.1 The method

3.3.1.1 Pillar I: Applied Risk Management

Successful risk management is a business-level decision-support tool: a way to gather data to make a good judgment call, based on knowledge of vulnerabilities, threats, impacts and probabilities [18, pp. 26]. Applied Risk Management is approached with a risk management framework (RMF). The main purpose of the RMF is to consistently track and handle risks [18, pp. 41].

3.3.1.2 Pillar II: Software Security Touchpoints

McGraw presents the software security touchpoints as a set of best practices. They can be applied for each software process to build software [18, pp. 34].

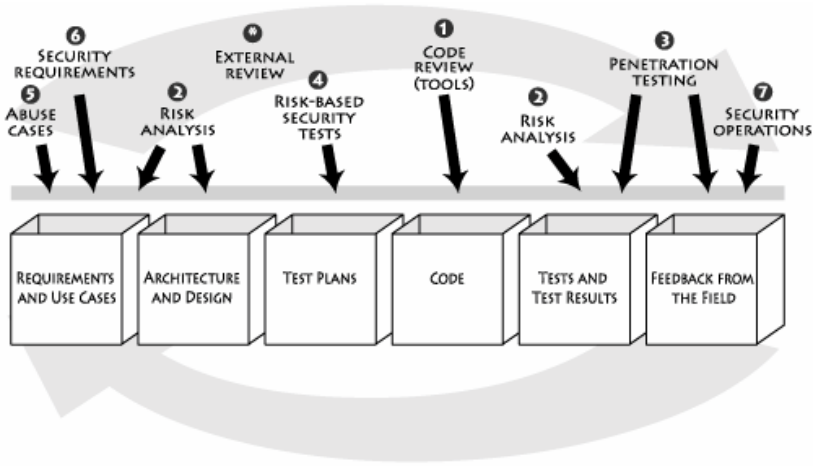


Figure 3:
[18, pp. 84] Software Security: Building Security In Figure 3-1 Lightweight software security best practices called touchpoints are applied to various software artifacts. The best practices are numbered according to effectiveness and importance.

Some touchpoints are, by their nature, more powerful than others. The numbers in Figure 3 present an order of effectiveness [18, pp. 84]. Seven security touchpoints include:

1. **Code review** [18, pp. 86]: Inspect code for implementation bugs. Static analysis tools scan source code for common vulnerabilities.
2. **Architectural risk analysis (ARA)** [18, pp. 86]: Uncover and rank architectural flaws so that mitigation can begin (see Section 3.3.2.2).
3. **Penetration testing** [18, pp. 87]: Understanding of security of fielded software in its real environment.
4. **Risk-based security tests** [18, pp. 87]: Test of security functionality with standard functional testing techniques and risk-based security testing *based on attack patterns*, risk analysis results, and abuse cases.
5. **Abuse cases** [18, pp. 88]: Abuse cases describe the system's behavior under attack (see Section 3.3.2.3).
6. **Security requirements** [18, pp. 88]: Security must be defined at the requirements level. These requirements should address both functional security and emergent characteristics (preventing abuse cases and attack patterns).
7. **Security operations** [18, pp. 88]: Operations staff carefully establish and monitor fielded systems during use to enhance the security posture.

From these touchpoints, points 2 and 5 are especially relevant for this thesis and are explored in Sections 3.3.2.2 and 3.3.2.3.

3.3.1.3 Pillar III: Knowledge

Pillar III involves gathering, encapsulating, and sharing knowledge that can be used to provide a solid foundation for software security practices. This pillar will not be further described in this thesis because it is not directly related with the subject of this project [18, pp. 35].

3.3.2 How *Building Security In* performs risk identification

Risk identification in the "Building Security In" book can be divided into two parts: Pillar I: Risk Management Framework (RMF) and Pillar II: Architectural Risk Analysis and Abuse Cases.

3.3.2.1 Risk Management Framework

The RMF involves identifying, tracking, and mitigating software risk over time. It is designed to manage software-induced business risks [18, pp 39-40].

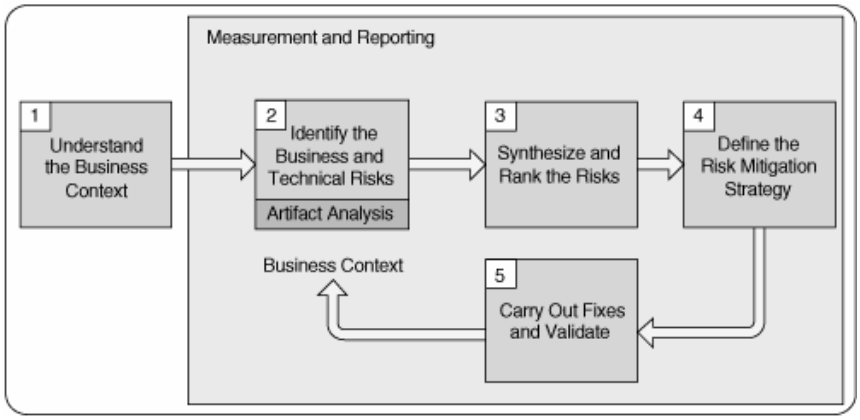


Figure 4: [18, pp. 42] Software Security: Building Security In Figure 2-1 The risk management framework (RMF). In this picture, the RMF is a closed-loop process with five basic activity stages, each of which is numbered.

The RMF [18, pp. 43-46] begins with Understand the Business Context (business goals, priorities, and circumstances), Identify the Business and Technical Risks (discover and describe technical risks and map them through business risks to business goals), Synthesize and Rank the Risks (risk prioritization based on business goals), Define the Risk Mitigation Strategy (optimal risk mitigation based on cost and likelihood of success), and Carry Out Fixes and Validate (obtain evidence that risks have been properly mitigated).

3.3.2.2 Architectural risk analysis

Security analysts will uncover and rank architectural flaws so that mitigation can be initiated [18, pp. 86]. Risk analysis results and risk categories drive both into requirements and into *testing* [18, pp. 140].
Risk analysis should incorporate assistance from risk professionals external to the design team. It relies on an understanding of business impact, requiring the understanding of laws and regulations as well as the business model supported by the software [18, pp. 140].

Three critical steps of the architectural risk analysis [18, pp. 162-169] are:

1. Attack resistance analysis: How does the system fare against known attacks? *E.g. tokens meant to identify a user are easy to guess.*
2. Ambiguity analysis: The creative activity of a couple of experienced analysts to discover new risks. *E.g. protocol authentication problems.*
3. Weakness analysis: Understanding the impact of external software dependencies. *E.g. weaknesses in J2EE, .NET, etc.*

3.3.2.3 Abuse cases

Abuse cases describe the system's behavior under attack. This requires explicit coverage of what should be protected, from whom, and for how long [18, pp. 88]. Abuse cases [18, pp. 213- 217] are to be built by a team of requirements staff and security analysts. These people start with a set of requirements, a set of standard use cases (user stories), and a list of attack patterns.

First, they create a list of anti-requirements, which are functionalities that are not desired in the software. Next, an attack model is created by selecting attack patterns relevant to the system. Finally, abuse cases are built around those attack patterns. The attack model and anti-requirements form abuse cases that describe how the system reacts to an attack and which attacks are likely to occur.

3.3.3 How *Building Security In* captures the risk level

During the Pillar II activities, which include Architectural Risk Analysis and Abuse Cases, the technical threats and risks are assessed. However, the prioritizations of these risks need to be driven by the business risk, which is covered in the RMF. Business risk can be separated into three broad categories [18, pp. 155]:

1. Legal and/or regulatory risk
2. Financial or commercial considerations
3. Contractual considerations

Stage 3 of the RMF states that for synthesis and ranking of the risks [18, pp. 44-45], large numbers of risks will be apparent in any given system. However, the prioritization process will supply an answer in which technical risks have to be mitigated first. This will depend of the business goals that are threatened and the likelihood that the technical risks manifest themselves in a way that impacts the business.
The software development team will now have a list of software risks that are prioritized based on their impact on the business goals.

3.3.4 Critical analysis of the *Building Security In* method

The activities Architectural Risk Analysis and Abuse Cases are useful to identify the technical software risks. They can be used in combination with the software methodology the organization is already using. However, they are not very useful for our purpose of relating discovered vulnerabilities to business risks.

The RMF and the Pillar II are represented as two parts of a method that are related to each other. However, reading this book, it appears to me that the link is desired by the author but not presented clearly.
During the Architectural Risk Analysis and Abuse Cases the technical threats and risks on the software level are discovered. These technical risks are related to business goals via the RMF. A list of software risks is created that are prioritized based on their impact on the business goals.

Although I appreciate this perspective, I think that the step taken is too large to successfully relate technical risks to business goals. As will be shown in Chapter 6, it will be useful to first relate technical risk with the total control environment before relating it to any impact. What I perceived as lacking in the method is the question of whether the vulnerability actually has a threat. The theory on threat analysis will be a good enhancement for the “Building Security In” book [18].

An example given in the book is: If the flimflobble in sector four has a floosblozzle failure, that means we will miss the first quarter number by $2 million. I think that his kind of relation is not realistic. The question is whether there is no other control in the organizations control environment that will compensate the damage of the failure stated. This should be known before an impact can be estimated.

3.4 ISO/IEC 27034: IT – Security techniques – Application security

The purpose of ISO/IEC 27034 [2] is to assist organizations in integrating security seamlessly throughout the lifecycle of their application. The currently developed standard consists of six parts. The project scope includes Parts 1, 2 and 3 [2, 5, 6].

3.4.1 The method

3.4.1.1 Introduction

The ISO/IEC 27034 states that application security is influenced by a defined target environment. The type and scope of application security requirements are determined by the risks to which the application is subjected [2, pp. xii].

3.4.1.2 Part 1: Overview and concepts

ISO/IEC 27034-1 defines the application security scope as: *Application security protects the critical data computed, used, stored and transferred by an application as required by the organization. The criticality of data and other assets should be defined by the organization through its security risk assessment process* [2, pp. 6].

All components, processes and frameworks are part of two overall processes:

- ONF Management process (see Section 3.4.1.3)
- Application Security Management Process (ASMP) (see Section 3.4.1.3)

3.4.1.3 Part 2: Organization normative framework (draft)

Organization Normative Framework (ONF)
The ONF [2, pp. 14-30] is an organization-wide framework containing a subset of the organization's processes and components that are relevant to application security and are normative inside the organization. Every organization has a normative framework. It is a compendium of all regulations, policies, practices, roles and tools used by the organization. See Figure 5

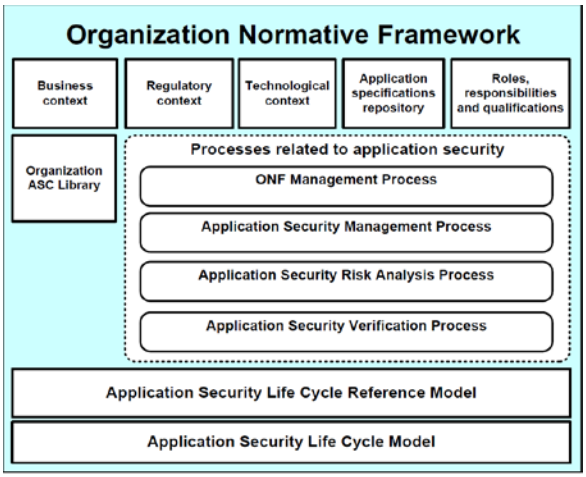


Figure 5: [5, pp. XXV] ONF – simplified graphical representation

Below the relevant parts for this project are described.

Business context [5, pp. xxvi]	The business context is an inventory and documentation of all business processes, standards, and best practices adopted by the organization that can have an impact on application projects *i.e. Business, risk analysis, security policy, information assets with security classification, best practice for programming languages, etc.*
Regulatory context [5, pp. xxvii]	The regulatory context is an inventory and documentation of all laws and regulations that can have an impact on application projects, in any of the organization's business locations *i.e. Applicable law and regulations for all locations.*
Technological context [5, pp. xxxviii]	The technological context is a documentation of the organization's IT components and the organization's best practices and rules that apply to the use of such components *i.e. list of IT components relevant for application security.*
Organization Application Security Controls (ASC) Library [5, pp. xxx]	The ASC Library associated with one or many level(s) of trust. The library is used by an organization for ordering ASCs according to the levels of trust they apply to and for selecting appropriate ASCs in the course of an application project *i.e. list of levels of trust with list of ASCs assigned to each level of trust. E.g. for a C3 application ASC threat modeling is required.*
Application Security Management Process [5, pp. xlii] See Section 3.4.1.4.	
Application Security Risk Analysis Process [5, pp. xliv] See Section 3.4.2.	

3.4.1.4 Part 3 Application security management process (draft)

Application-level framework and processes are provided by Part 3: Application Security Management Process (ASMP). The ASMP helps a project team to apply relevant portions of the ONF to a specific application project and to formally record evidence of the processes in an Application Normative Framework (ANF) [6, pp. vi]. The Application Security Management Process contains five steps [6, pp. 3]:

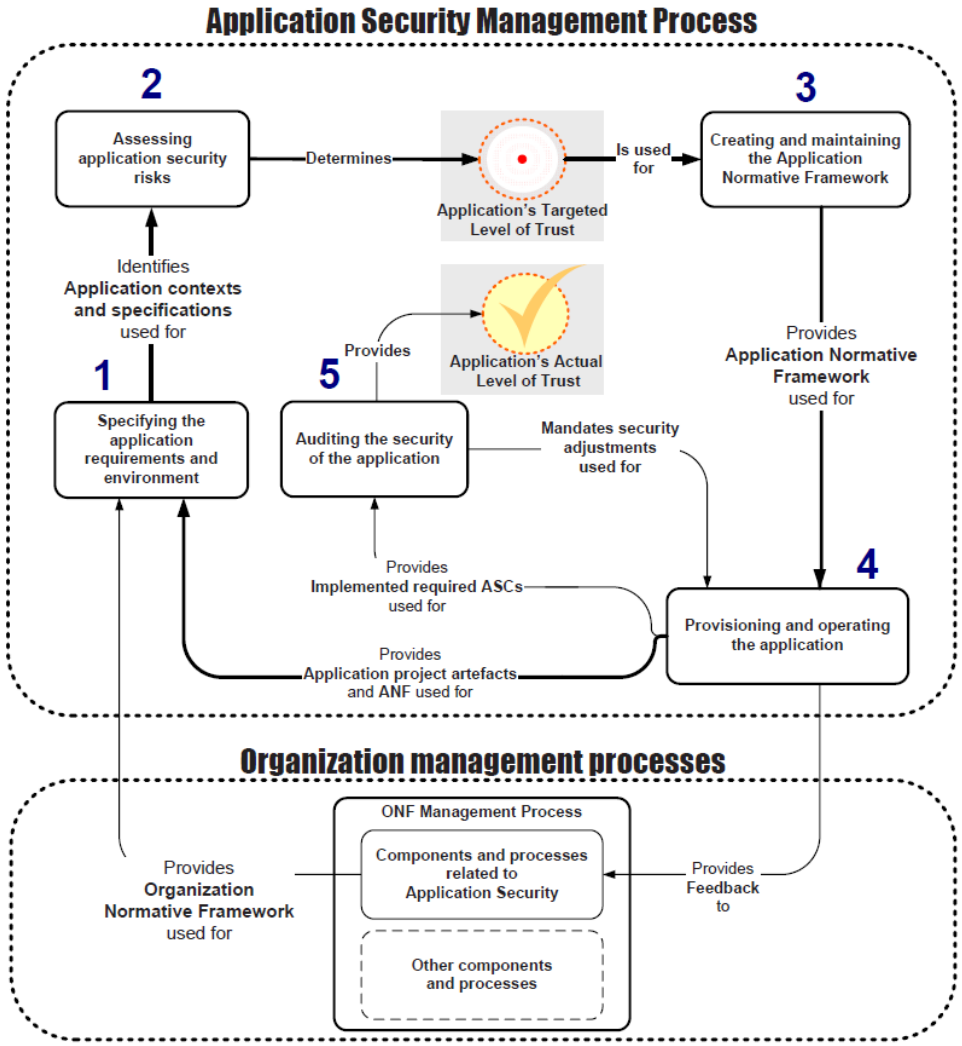


Figure 6: [2, pp. 12] ISO/IEC 27034-Part1, Figure 3 – Organization Management Processes

ASMP Step 1: Identifying the application requirements and environment

General application meta information is recorded including [6, pp. 5-7]:

a) *Actors*: persons or processes that perform an activity or initiate interaction with any process provided by or impacted on by an application.
b) *Specifications*: major sources are regulatory and compliance documents. Also, organizational and business objectives/ visions and organizational security requirement policies e.g. minimum password requirements.
c) *Information*: Understanding the information that flows within an application is a key step in deriving security requirements. Information is derived from various sources including data generated by business and technological context, data pertaining to the application and structure, data stored, data in transmission
d) *Environment*: technological, business and regulatory contexts.

ASMP Step 2: Assessing application security risks

This step assesses the risk for a specific application project and produces security requirements, that are used to obtain the desired level of trust for the application [6, pp. 8]. See Section 3.4.2.

ASMP Step 3: Creating and maintaining the ANF

Relevant elements from the ONF that apply to a specific application project are recorded in the ANF. The organization will select the applicable security controls (ASCs) for the application project [6, pp. 12]. Exceptions should be recorded in the ANF and validated by the verification team when ASCs are not applicable or implementable for an application. Exceptions requiring accepting residual risk should be signed off by the application owner and periodically reviewed [6, pp. 16].

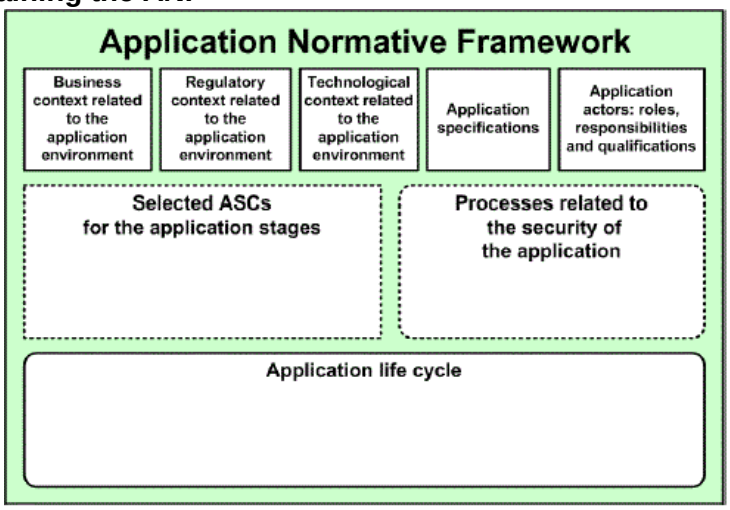


Figure 7: [6, pp. 16] ISO/IEC 27034-Part 3, Figure 3 – ANF

This ANF exists and evolves during the application's life cycle. Changes to the ANF have an impact on the application's security and should be tracked and approved by the application owner [6, pp. 17].

ASMP Step 4: Realizing and operating the application

Step 4 involves the use of the Application Security Controls. The project and verification teams are supplied with the ASCs associated with the Target Level of Trust for their project [6, pp. 17].

The most commonly applied **security activity** components of ASCs are [6, pp. 19]:

- *Attack surface reduction*: reducing risk by providing attackers less opportunity to exploit a potential weak spot or vulnerability.
- *Threat modeling*: allows development teams to analyze the security implications of designs and security issues for the application.
- *Use approved tools*: new security analysis functionality and protections leveraged by approved tools and their associated security checks.
- *Deprecate unsafe functions*: analyze all functions and APIs that will be used and prohibit those that are determined to be unsafe.

The most commonly applied **security measurement** parts of ASCs [6, pp. 20-21] are:

- *Static analysis*: analysis of source code to ensure that secure coding policies are being followed.
- *Dynamic program analysis*: run-time verification of software programs to ensure that a program's functionality works as designed.
- *Fuzz testing*: form of dynamic analysis used to induce program failure by deliberately introducing malformed or random data to an application.
- *Threat model and attack surface review*: revisit the threat models and attack surface measurements of an application when the code is complete to include changes to functionality during development.
- *Exception process*: the verification team will review exception requests, and grant exceptions when appropriate.

ASMP Step 5: Auditing the security of the application

The purpose of the fifth step is to verify and formally record the supporting evidence of the application's Targeted Level of Trust at a specific time. An organization can declare an application secure when its Actual Level of Trust is equal to its Targeted Level of Trust. If some ASCs fail, the organization will take appropriate measures to address the problem [6, pp. 22-24].

3.4.2 How *ISO/IEC 27034* performs risk identification

Risk assessment includes three sub-steps: risk identification, risk analysis and risk evaluation. The risk assessment process produces the security requirements from which the application's Targeted Level of Trust is derived.
This, in turn, becomes the goal for the application's project team [6, pp. 8].
ISO/IEC 27034 refers to ISO/IEC 27005 for the definitions of risk assessment.
ISO/IEC 27034-Part 2 describes how risk analysis can be used to determine the applications Target Level of Trust.

Risk identification
According to ISO/IEC 27005, risk identification determines what could happen to cause a potential loss, and provides insight into how, where and why the loss might happen. It includes [1, pp. 13-16 including relevant parts of ISO/IEC 27034]:

- Identification of assets: anything that has value to the organization and therefore requires protection.
- Identification of threats: the potential to harm critical information in the application scope, and thereby the organization itself. Threats originate from the application's environment and actors [2, pp. 10],[5]
- Identification of existing controls: existing and planned controls should be identified.
- Identification of vulnerabilities: the result of inadequate or nonexistent controls from actors, processes, technological context and specifications [2, pp. 10].
- Identification of consequences/ impact: an impact is the cost to an organization of suffering a breach in availability, integrity or confidentiality of its critical application data [2, pp. 9].

Risk Analysis
The application risk analysis process includes two general steps [6, pp. 10]:

a) High-level risk analysis *(usually performed during the project initiation phase):* The high-level risk analysis defines, in a simple heuristic manner, the application's Targeted Level of Trust of the application according to information provided in the ANF (*e.g. internal users or www users, credit card data involved)*
b) Detailed risk analysis *(usually performed in the realization phase)* The goal of the detailed risk analysis is to more precisely identify the application specifications and contexts (*e.g. has the application an own authentication mechanism or authenticated against active directory).*

Risk evaluation
Risk evaluation uses the understanding of risk obtained by risk analysis to make decisions about future actions: whether an activity should be undertaken and priorities for risk treatment.
The application owner has the responsibility of accepting the residual risks of a specific application after the risk evaluation has been performed [6, pp. 11].

3.4.3 How *ISO/IEC 27034* captures the risk level

The ISO/IEC 27034 does not include a method for capturing a risk level but refers to the ISO/IEC 27005 for risk assessment.
The risk assessment step of the ASMP results in identifying the application's Targeted Level of Trust. Level of Trust is not complementary to risk. Rather, it is similar to the concept of a "security plan", which is a set of controls authorized by an organization in order to reduce the risk that was determined by a risk analysis [2, pp. 19]. However, these Targeted Levels of Trust provide an indication of whether an application needs to be highly protected or not.

3.4.4 Critical analysis of the *ISO/IEC 27034* method

The ISO/IEC 27034 describes a proper process that can be used to develop secure software. The method also includes other phases of the Application Security Lifecycle that can be used to cover the entire application lifecycle.

The input from business, regulatory and technological contexts promised to be useful material for this project. These contexts can be used to discover the environment of the vulnerability as input for the threat and impact assessment.
Business context activities (see Section 3.4.1.3), cause risks, and the organization will determine security requirements for mitigating those risks [5, pp. xxvi]. However, how the context is related to risks that can occur in software is not described. Instead, the method defines a Targeted Level of Trust. The Targeted Level of Trust for an application can be used to select the required security activities that need to be performed for the given application. A lot of the ASC activities to be performed are also included in the Microsoft SDL method.

Only high-level threat modeling is mentioned as a method to discuss the security implications of designs. However, how this should be accomplished is not mentioned. For the purpose of this project, the method lacks a connection from the security objectives to the risk that will ultimately affect the business processes.

3.5 Critical comparison of three presented methods

The overall conclusion, of this project, from comparing the three development standards is that the Microsoft SDL method describes the secure software development process on a rather technical level and lacks the connection with business risks. The other two methods mention the link with business processes, but they describe the secure software development process on a rather high level, with no practical implementable linkage of technical risk to business risks.

3.5.1 Comparison of risk identification/ analysis within the methods

All methods of risk identification are rather technical risk identification methods. The methods begin with the technical threats. The “Building Security In” method presents a model of how to link the technical risks back to the business risk; however, this relationship should be reversed. To guide the security software assessment, threats from the broad business area should be evaluated and used as input for the technical threat assessment. By proceeding in this way, the technical assessment will be guided by issues the business sees as a real risk.
For this project, material to identify and analyze risk is too high-level to be applicable for the relation of discovered vulnerabilities to business risk.

3.5.2 Comparison of capturing of the risk level within the three methods

The risk levels in the three methods are not discovered in such a way that they can be used for the relation of discovered vulnerabilities to business risk, because they lack a proper connection with the business risks.
The Microsoft SDL method does not capture risk to rank vulnerabilities; rather, it uses a measure of whether they are easy to exploit. The “Building Security In” method presents a framework to link technical to business risks, but it lacks a proper threat assessment to be useful to attain an understanding of risks at the business level. The ISO/IEC 27034 establishes target levels of trust and assesses whether the project is applied to this level of trust; it is not useful to relate discovered vulnerabilities to business risk.

The conclusion can be drawn that my understanding is correct that software security methodology presents and ranks the priority of vulnerabilities in a rather technical way that is not able to be recognized by business managers.

4 Detection of computer related risk

4.1 Introduction

Literature addressing the identification of computer-related risk can be found in a broad variety. The library of the RHUL is used to research relevant literature about this subject. The NIST 800-30: Guide for Conducting Risk Assessments (See Section 2.6) is used a guiding resource to research this corpus of material.

The aim of this part of the project is to investigate whether methodologies from the field of general computer risks can contribute to a method of classification of vulnerabilities that is understood by business managers who establish priorities. The following literature has been selected to obtain relevant material to identify threats and vulnerabilities:

- Blyth and G.L. Kovacich. Information Assurance: Security in the Information Environment [9] provides a thorough overview of the subjects that form risk. The overview of threat components and their relationships is useful material.
- E.G. Amoroso. Fundamentals of Computer Security Technology [7] Although this book was not recently written, the material presented is still relevant for the IT Security community. He presents threat trees that show, for every sub-threat, a criticality, effort and resulting risk. This method can be used to determine where the protections for threats can best be placed.
- P.G. Neumann. Computer Related Risks [20] presents numerous examples of software risk that emerged in sectors such as space aviation. This material is usable background information for the planning process addressing how to relate software risk to business risks.
- R.N. Charette. Applications Strategies for Risk Analysis [10] provides relevant information addressing how to relate individual software vulnerabilities to systems and the broader context of an application.

In the following paragraphs, relevant parts of the literature investigated will be provided as groups of knowledge. This review begins by defining the object of assessment. The generic risk model of NIST (see Section 2.6) will be used to present an overview of relevant material concerning risk that was encountered in the literature.

4.2 Defining the object

Vital to a risk analysis is to select the proper **system** of interest and **environment** of interest. This will include the business, systems engineering, political, legal, technological, development, and operational environments [10, pp. 86].

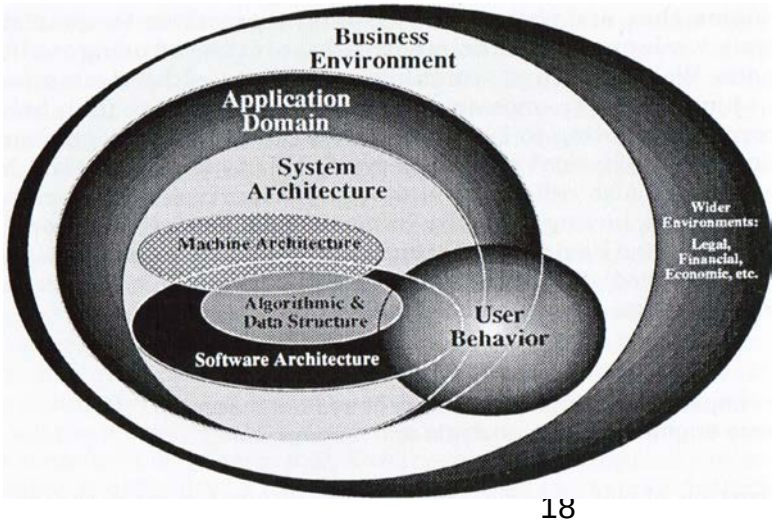


Figure 8:
[10, pp. 72] Applications strategies for Risk Analysis, Figure 2.3 Environments of Interest

The structure of systems may be very complex. Attributes of complexity include: significant numbers of system interactions; a large number of system elements; nonlinear system behavior; asymmetry of system processes and parts of a system, not operating under the regime of a central control mechanism [10, pp. 70].
In a distributed system, it is not always clear which hidden dependencies exist, and which of those involve dependence on less trustworthy systems and networks. Distributed systems exacerbate the need for different system components to be able to authenticate one another and the users of one another [20, pp. 210].
Taking this into account, to perform a risk analysis well, one must use a systematic point of view. The risk analyst must [10, pp. 72]:

1. Understand where the system belongs to; what the greater context is.
2. Understand what the system does; what the elements, functions are.
3. Understand how the system acts; data and control flow of the (sub)system.

When the risk analyst oversimplifies the models used to describe the system, the risk will not be observed. On the other hand, too much detail will cause the analysis to be bogged down in nonessential trivia [10, pp. 86].

An application has a process component, an architecture component (conceptual structure and functional behavior), and a physical instantiation called an application realization (i.e. a system). The programs are composed of data structures and algorithms [10, pp. 300].

4.3 Risk

NIST: ***Risk*** *is a measure of the extent to which an entity is threatened by a potential circumstance or event, and is typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.*

It is essential to understand the vulnerabilities and threats before we can undertake a meaningful analysis of the risks [20, pp. 257]. Risk is caused by the lack of information, the lack of control, and/or the lack of time. To reduce or avert a risk, one or more of these causes must be changed [10, pp. 15-19].

Information systems have increasingly become an integral part of the business environment. This implies that one must start with the risks from the business environment [10, pp. 38].

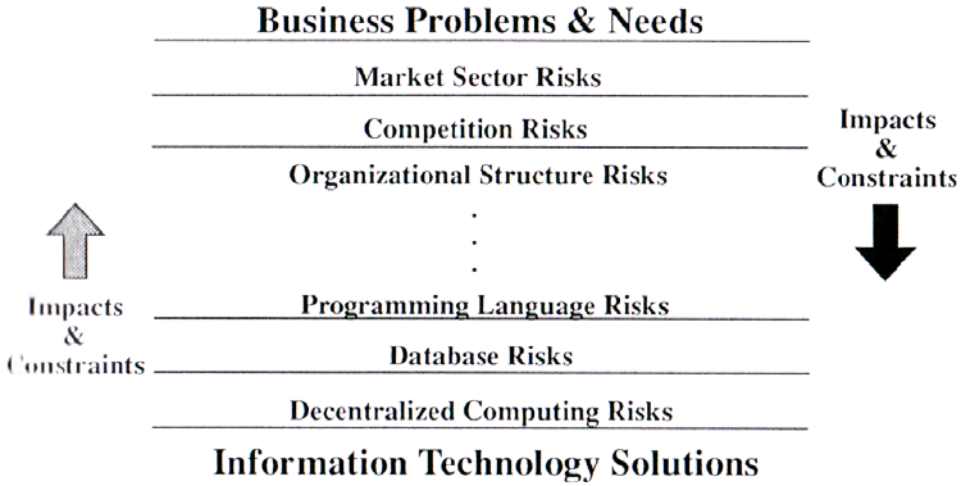


Figure 9: [10, pp. 39] Applications strategies for Risk Analysis, Figure 1.17 Flow of Risks

The risks, before they reach the development project, flow down through many “layers”, each changing or impacting the original risk [10, pp. 126]. System engineering is primarily concerned with building the desired product, whereas software engineering is concerned with building it correctly [10, pp. 128]. Within the system engineering environment, risks coming down via the business environment and software risks coming up from the software engineering environment will be balanced [10, pp. 130].

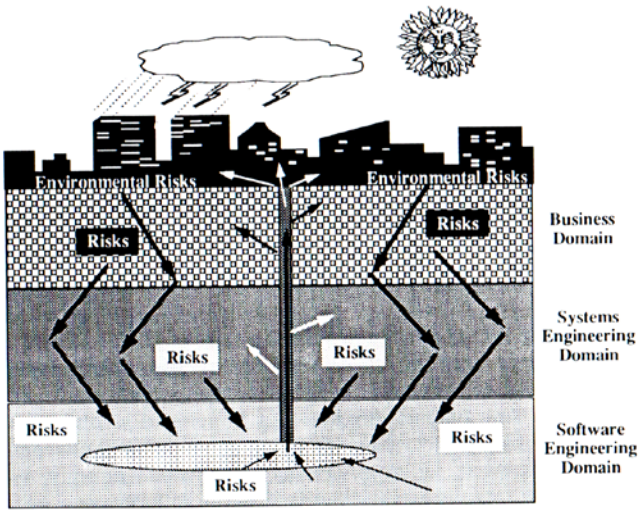


Figure 10: [10, pp. 127] Applications strategies for Risk Analysis, Figure 2.18 Risk environment

Threat trees can be used to consider the optimal placement of security protections to reduce the risk [7, pp.23-24].

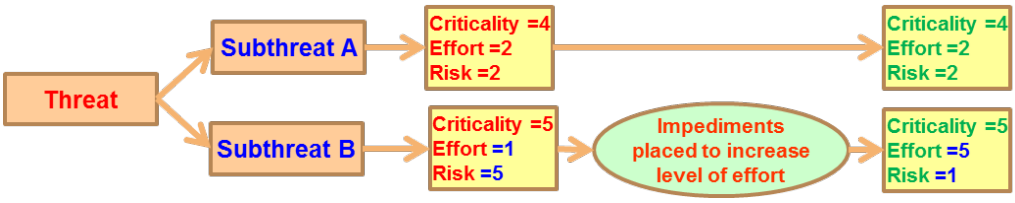


Figure 11: [7, pp. 23-24] Fundamentals of Computers Security Technology, Figure 2.8 & 2.9

When building information systems, the risks with which one is confronted appear much different at the business level than at the system engineering or software engineering levels. Refer to the overview of reality and perception pairs in Figure 12 below [10, pp. 455-469].

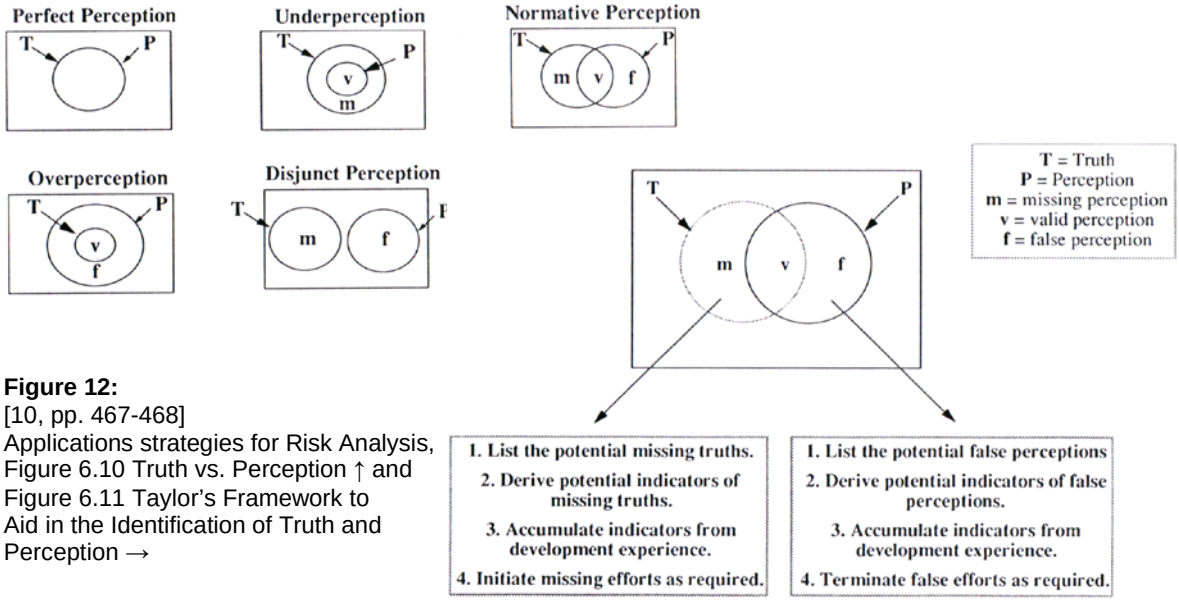


Figure 12:
[10, pp. 467-468]
Applications strategies for Risk Analysis, Figure 6.10 Truth vs. Perception ↑ and Figure 6.11 Taylor’s Framework to Aid in the Identification of Truth and Perception →

4.4 Risk analysis methods

In performing security risk analysis, the driving factor is the underlying level of security desired as defined by the company's and/ or project's security policy. Without such a policy, there is nothing to measure the risk against [10, pp. 401].

Risks associated with software applications arise from three areas: the application knowledge, the logical application architecture, and the physical realization of the application architecture [10, pp. 293].

Issues that should be accounted for during (technical) risk analysis are that risks will vary due to: the difficulty of the problem; the relationship in time between data and processing; the number of simultaneous tasks to be performed; the relative difficulty of data, control and algorithmic aspects of the problem; and whether the application is deterministic or nondeterministic [10, pp. 330].

4.5 Threats

NIST: *A **threat** is any circumstance or event with the potential to adversely impact organizational operations and assets, individuals, other organizations.*

Software can be vulnerable to deletion, interruption of execution, interception of software in transit, and modification. Hardware is vulnerable to theft and interruption of service. Information is vulnerable to interruption (loss), interception, modification and fabrication [9, pp. 38].

Systems need not only be fault tolerant and secure against malicious misuse, and more predictable in their behavior, but also less sensitive to the mistakes of people involved in system development and use [20, pp. 263].
Computer systems become untrustworthy because of the actions of individuals. The principle of separation of duties and least privilege provide an aid to designing systems and applications so that only critical portions of the systems are required to be trustworthy. The system should be capable or protecting itself against both intentional and accidental misuse [20, pp. 270].

A weak link is a point in a system at which a failure or an attack can cause the system to become incapable of continuing to satisfy its requirements. It is desirable to protect against multiple causes, because we can never guarantee that only a single failure or attack will occur at any one time [20, pp. 120].

4.6 Likelihood

NIST: *The **likelihood** of occurrence is a weighted risk factor based on an analysis of the probability that a given threat is capable of exploiting a given vulnerability (or set of vulnerabilities)*

The likelihood of a threat being able to exploit a vulnerability is assessed in Figure 13 below that presents an overview of the components of a threat:

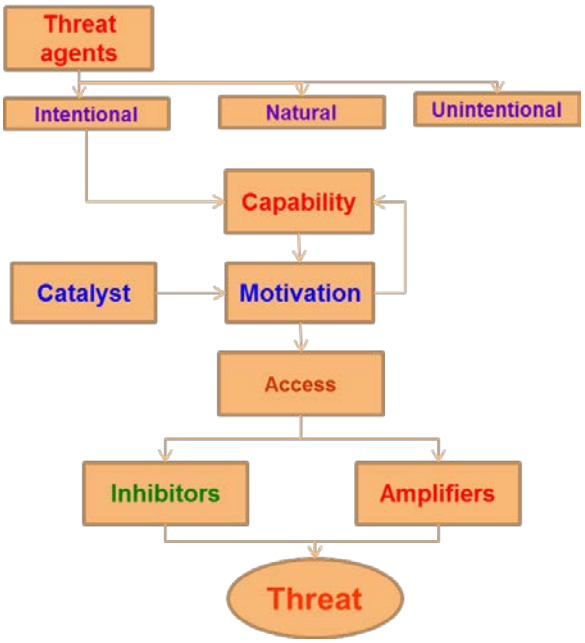


Figure 13:
[9, pp. 34 Information Assurance: Security in the Information Environment, Figure 3.4 (Information added). The Threat Components and Their Relationships.

Components of a malicious threat:

Component:	**Description:**	**Examples:**
Natural threat agents	Acts of God	Fire, flood, power failures, rock movements, and other.
Unintentional threat agents	Parties that cause damage or loss of service without direct intent.	Employees of the organisation or external parties. For example, a software flaw during the use of software.
Intentional threat agents	Those parties that would knowingly seek to make a threat manifest	Criminals, terrorists, subversive or secret groups, state sponsored, disaffected employees. Hackers, pressure groups, and commercial groups.
Capability	Capability to conduct and sustain an attack or totally destroy the system and any replacement	Software, technology, facilities, education and training, methods and books and manuals.
Threat catalyst	Those factors or actions that cause an attack to be initiated at the time and on the target selected.	• Events that influence the threat agent to trigger pre-determined actions. • Technology change: use of shortcomings. • Personal circumstances: change in personal circumstances of the threat agent.
Threat agent motivators	Factors and influence s that motivate a threat agent.	Could be; Political, Secular, Personal gain, Religion, Power, Terrorism or Curiosity
Threat inhibitors	Any factor that decreases either the likelihood of an attack taking place or being successful	Fear of capture, Fear of failure, Level of technical difficulty, Cost of undertaking the attack, or Sensitivity to public perception
Threat amplifiers	Any factor that increases either the likelihood of an attack taking place or being successful	Peer pressure: status in peer group, Fame, Access to information, Changing high technology, De-skilling through scripting, Skills and education levels, Law enforcement activities, Target vulnerability, Target profile or Public perception

Figure 14: [9, pp. 34-38 Information Assurance: Security in the Information Environment, Resume paragraph 3.3 Threat components applying to Malicious Threats.

Threat trees are a top-down technique to assist system designers during the security requirement analysis phase of computer development. They are derived from fault trees analysis (FTA) in system reliability engineering, to prevent system failures due to errors (see Figure 15 below).
The complete set of threats that exists for a given system is the root of the tree. The next level underneath will show sub-threats, and so on. The resulting analysis documents provide the rationale for each identified threat [7, pp.15-18].

The weakness of FTA is that the tree can become extremely complex and dependencies between events are difficult to ensure [10, pp. 373-374].

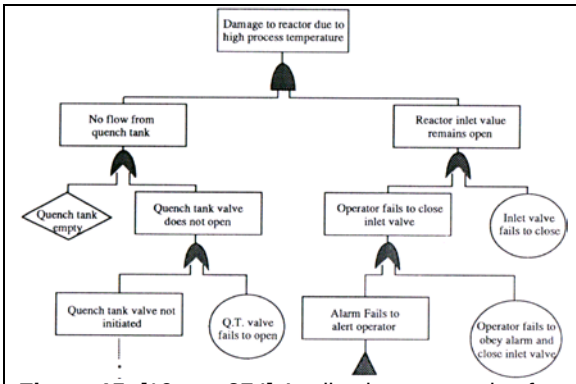


Figure 15: [10, pp. 374] Applications strategies for Risk Analysis, Figure 5.4b Partial Fault Tree

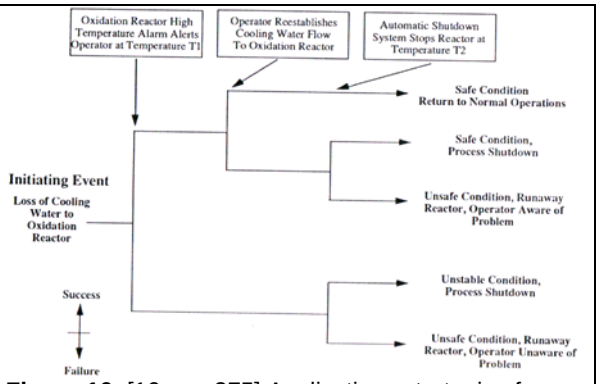


Figure 16: [10, pp. 375] Applications strategies for Risk Analysis, Figure 5.5 Sample Event Tree

Event Tree Analysis (ETA) identifies potential accidents by means of “forward analysis” from an initiating event. ETA is bottom-up and uses deduction to trace from an undesired event back to its basic causes [10, pp. 374-375].

4.7 Threat source

NIST: *A threat source is characterized as:*
- the intent and method targeted at the exploitation of a vulnerability; or
- a situation and method that may accidentally exploit a vulnerability

An example of an unintentional threat agent is a previously unknown software flaw in a widely deployed General Electric system. The problem is that software can never be exhaustively tested for bugs, and the larger the piece of software the greater the chance of encountering a bug when using the software [9, pp. 30-34].

An issue that greatly complicates the prevention of a threat is that even if an attack is observed, the motivation to perform the attack cannot be identified. Even when e.g. a malicious programmer inserts a statement to be used for an attack and is caught, he can claim that it was an innocent mistake [7, pp. 5]

Vulnerabilities can manifest themselves in different types of misuse:
External misuse of an information system is related to the creation, manipulation, and destruction of information by a user within the organisation.
Hardware misuse of an information system is primarily concerned with the Information Assurance of the physical devices.
Masquerading misuse of an information system is primarily concerned with the authentication of information, its sources, destinations, and users.
Pest programs are primarily concerned with the availability of information system services and their expected behaviour (Trojan horses, Logic bombs, viruses, etc.).

Bypasses are types of misuse of an information system concerned with authorisation and configuration management (e.g. improper initialisation or termination).
Active misuse of an information system is primarily concerned with modification of information, or entering false or misleading information.
Passive misuse of an information system is primarily concerned with exploiting the information within the system for the purpose of conducting analysis and making inferences about the existence of sensitive data.
Inaction misuse of an information system is primarily concerned with wilfully failing to perform expected duties or committing errors of omission.
Indirect misuse of an information system is primarily concerned with preparing for subsequent misuses, as in off-line pre-encryption matching, or factoring large numbers to obtain private keys [9, pp. 38-43].

4.8 Vulnerability

NIST: *A **vulnerability** is a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source*

Figure 17 below describes the stages of software development with the implication that errors can be made in each stage. Errors in the early stage are more severe than error in later stages [10, pp. 32].

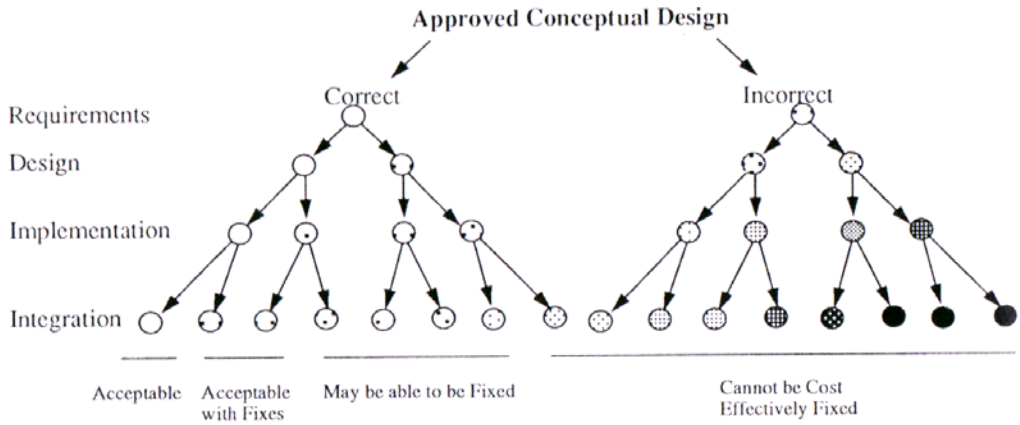


Figure 17: [10, pp. 32] Application strategies for Risk Analysis, Figure 1.12 Flow of Errors

Integrity concerns maintaining the correct and consistent condition of the system and system data (system integrity), and of (user) data entities (data integrity). Internal consistency means that the inside of a system is accurate, and external consistency means that the internal components of the system accurately reflect the outside world [20, pp. 96-97].

Reliability implies that a system performs according to the functional specifications, and does so consistently over time. A distinction is made among faults, errors and failures in systems. A fault is a condition that could cause the system to fail. An error is a deviation from expected behavior (some can be ignored and/or overcome). A failure is an error that is not acceptable. Problems related to time and date (e.g. summer time) and computing errors (e.g. floating point accuracy) are examples that can cause reliability issues [20, pp. 12].
Accidental losses of data or accidental physical damage to computer hardware caused by natural disaster or other circumstances can be defined as software disaster recovery. Accidental disclosure of data or malicious damage of physical computing assets is considered to be a security issue [10, pp. 387].

Security
Security refers to protection against human misuse, and excludes protection against malfunction. Three aspects of computing-- processing, storage and transmission -- and the security problem posed by the transition of data from one aspect to another must be taken into account when reviewing security. In each stage, the three basic elements of security must be assured: confidentiality, integrity and availability. The impact must be assessed based on the value of information [10, pp. 391].

4.9 Selection of useful material form computer risk theory

The aim of this portion of the project was to investigation whether methodologies from the field of general computer risks can contribute to a method of classification of vulnerabilities that is understood by business managers who establish priorities. The investigated literature stated that the different perceptions of risk at the business and software engineering level is to be taken into account [10, pp. 455-469]. Based on this observation, we can conclude that there is a need to make visible the estimates made during risk assessment. When they are visible, they can be used as background information for discussing levels of risk.

The theory for defining an object with its different environments [10, pp. 72] can be used to relate a discovered software vulnerability back to the business process or broader business environment (e.g. a software defect is part of a function that is a component of a system that supports a business process).
The notion that impacts originate from business problems and needs, and propagate down to IT solutions, and vice versa, has also to be taken into account when relating business and IT risks.

Threat trees can be used to identify threats on the business process level and to define the threats at the software levels below them [7, pp.15-18]. This method will be used to evaluate the threats that will exploit a vulnerability (e.g. connect the threat of changing an account number in an application to the business threat of stealing money).

The model to identify the likelihood of a threat being able to exploit a vulnerability by evaluating the components of a threat is a method that this project will use. The components of the likelihood of the software vulnerability occurring can be made visible to discuss these with the business managers (e.g. when the threat assessment provides insight that the capability of an attacker to exploit the vulnerability is set as high, then this estimate is visible and can be discussed between the assessor and the business manager).

5 Relating IT risk to business risk

There is a broad range of methods and frameworks that addresses the relationship between IT risk and business risk/objectives. The literature selected is commonly known and is internationally accepted as good practice. The book “No Excuses” was selected because of the link between Operational Risk and Business Process Management.
During the review of the literature, the concepts linking IT risks to risks relevant for management were assessed.
Two examples will be used by each in this chapter (mentioned in italics):

- *Breach of sensitive client data by software flaw*
- *Inaccurate calculation of premium due to software bug*

This chapter concludes with a discussion of which parts can be used for this project.

Why this literature?
COBIT5 is presented on the website of ISACA as: the only business framework for the governance and management of enterprise IT. It incorporates the latest thinking in enterprise governance and management techniques, and provides globally accepted principles, practices, analytical tools and models to increase the trust in, and value from, information systems. It is a framework that is constantly updated according to new views on the subject. Additionally, a previous version was used as the basic resource during a former study of mine to become a certified IT auditor.
IRAM is an acronym for Information Risk Analysis Methodology (IRAM) and is presented as an essential business tool that helps organizations to identify, analyze and manage information risk throughout the enterprise.
We use a former version of IRAM within the company that employs me. A strong point is that the method starts with a business impact assessment to determine the business requirements for confidentiality, integrity and availability of an application.
The **ISO 27005** is used during the core subject Security Management of the MSc Information Security at the RHUL. It is also a well-known and heavily used standard when evaluating information security in everyday practice.
These standards state that Information security risk management should contribute to the risks being assessed in terms of their consequences to the business and the likelihood of their occurrence.
SABSA is presented as a proven methodology for developing business-driven, risk- and opportunity-focused Security Architectures at both the enterprise and solutions levels, traceably supporting the business objectives. The business-driven risk approach for security should be useful input for this project.
A business process approach to managing Operational Risk stated that it describes how operational risk affects the bottom line, shareholder value, reputation, and even survival. It provided an overview of risk and operational risk management (ORM) and how ORM and Business Process Management can be integrated. It can be useful material to relate IT risks via operational risk to business processes that are of the concern of the manager of these business processes.

5.1 COBIT5

COBIT5 [3, 4, 15] uses a value creation approach. This entails realizing benefits at an optimal resource cost while optimizing risk. Risk is defined as the combination of the probability of an event and its consequence [4, pp. 17].
IT risk is defined as business risk, associated with the use, ownership, operation, involvement, influence and adoption of IT within an enterprise.
IT risk consists of IT-related events that could potentially impact the business. IT risk can occur with both uncertain frequency and impact, and it creates challenges in meeting strategic goals and objectives [4, pp. 17].

COBIT5 is not risk-oriented, but rather is goal-oriented. Goals are cascaded from the stakeholders to enable goals. There are 17 enterprise goals defined that are developed using the four Balance Scorecard dimensions: Financial, Customer, Internal, and Learning and Growth. These enterprise goals are supported by IT-related goals (Figure 19 below) requiring the application and use of enablers to be achieved. Enablers include processes, organizational structures, and information. For each enabler, a set of goals is defined in support of the IT-related goals [8, pp. 21-24].

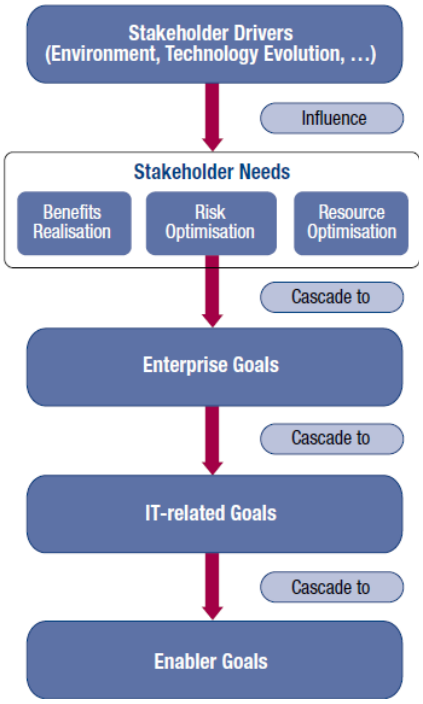


Figure 18: [3, pp. 18] COBIT5 Framework, Figure 4 The Governance Objective: Value Creation

IT BSC Dimension	Information and Related Technology Goal	
Financial	01	Alignment of IT and business strategy
	02	IT compliance and support for business compliance with external laws and regulations
	03	Commitment of executive management for making IT-related decisions
	04	Managed IT-related business risk
	05	Realised benefits from IT-enabled investments and services portfolio
	06	Transparency of IT costs, benefits and risk
Customer	07	Delivery of IT services in line with business requirements
	08	Adequate use of applications, information and technology solutions
Internal	09	IT agility
	10	Security of information, processing infrastructure and applications
	11	Optimisation of IT assets, resources and capabilities
	12	Enablement and support of business processes by integrating applications and technology into business processes
	13	Delivery of programmes delivering benefits, on time, on budget, and meeting requirements and quality standards
	14	Availability of reliable and useful information for decision making
	15	IT compliance with internal policies
Learning and Growth	16	Competent and motivated business and IT personnel
	17	Knowledge, expertise and initiatives for business innovation

Figure 19: [4, pp. 13] COBIT5 Enabling processes, Figure 5 IT-related goals

The core processes for COBIT5 from Risk Management Perspective are:

COBIT 5 Process Identification	Reasoning
EDM03 Ensure Risk Optimisation	This process covers the understanding, articulation and communication of the enterprise risk appetite and tolerance and ensures identification and management of risk to the enterprise value that is related to IT use and its impact. The goals of this process are to: • Define and communicate risk thresholds and make sure that key IT-related risk is known. • Effectively and efficiently manage critical IT-related enterprise risk. • Ensure IT-related enterprise risk does not exceed risk appetite.
APO12 Manage Risk	This process covers the continuous identification, assessment and reduction of IT-related risk within levels of tolerance set by enterprise executive management. Management of IT-related enterprise risk should be integrated with overall ERM. The costs and benefits of managing IT-related enterprise risk should be balanced by: • Collecting appropriate data and analysing risk • Maintaining the risk profile of the enterprise and articulating risk • Defining the risk management action portfolio and responding to risk

Figure 20: [15, pp. 57] COBIT5 for Risk, Figure 33 Core risk processes

A key information item of “AP012 Manage Risk” is the risk scenario. A possible event will have an uncertain impact on the achievement of the enterprise’s objectives. Two different or a combination of these approaches can be used to establish the risk scenarios for the enterprise :

- Top-down, where one starts from the overall enterprise objectives and performs an analysis of the most relevant and probable IT risk scenarios impacting the enterprise objectives.
- Bottom-up, where a list of generic scenarios is used to define a set of more relevant and customized scenarios, applied to the individual enterprise situation [15, pp. 59].

For the last approach a list of example Risk scenarios is included in COBIT5, with negative and positive example scenarios. An example of positive and negative example scenarios:

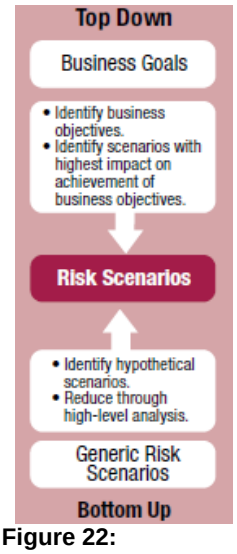


Figure 22: [15, pp. 57] COBIT5 for Risk, Figure 34 (partly) Risk Scenario Overview

Negative Example Scenarios	Positive Example Scenarios
Regular software malfunctioning of critical application software occurs.	Appropriate testing is conducted before the go-live decision to ensure the availability and proper functioning of the software.
Intermittent software problems with important system software occur.	

Figure 23: [15, pp. 71] COBIT5 for Risk, part of Figure 38 Example Risk Scenarios

Part of the Risk Management Perspective is “APO12: Manage Risk”. Below the *Respond to Risk* portion is shown.

APO12 Process Practices, Inputs/Outputs and Activities *(cont.)*

Management Practice	Inputs		Outputs	
APO12.06 Respond to risk. Respond in a timely manner with effective measures to limit the magnitude of loss from IT-related events.	From	Description	Description	To
	EDM03.03	Remedial actions to address risk management deviations	Risk-related incident response plans	DSS02.05
			Risk impact communications	APO01.04 APO08.04 DSS04.02
			Risk-related root causes	DSS02.03 DSS03.01 DSS03.02 DSS04.02 MEA02.04 MEA02.07 MEA02.08

Activities
1. Prepare, maintain and test plans that document the specific steps to take when a risk event may cause a significant operational or development incident with serious business impact. Ensure that plans include pathways of escalation across the enterprise.
2. Categorise incidents, and compare actual exposures against risk tolerance thresholds. Communicate business impacts to decision makers as part of reporting, and update the risk profile.
3. Apply the appropriate response plan to minimise the impact when risk incidents occur.
4. Examine past adverse events/losses and missed opportunities and determine root causes. Communicate root cause, additional risk response requirements and process improvements to appropriate decision makers and ensure that the cause, response requirements and process improvement are included in risk governance processes.

Figure 24: [4, pp. 111] COBIT5 Enabling processes, APO12 Process Practices, Inputs/Outputs and Activities (cont.)

How can the two example of software vulnerabilities handled by using COBIT5? APO12 states that business impact must be communicated to decision makers. The “impact” can be assessed using the risk scenario structure. The Actor, Threat Type, Event, Asset/Resource and Time can be described. The link with assets/resources can provide a notion regarding what form of “impact” the security breach or the inaccurate calculation has on the goals of the organization that are supported by the asset.

5.2 IRAM

The ISF Information Risk Analysis Methodology (IRAM) method contains three phases: Business Impact Assessment (BIA), Threats &Vulnerabilities Analysis (T&VA), and Control Selection (CS).
Before the actual BIA can be initiated, a business Impact Reference table must first be defined for the organization (once for the entire organization). For each of the business impact types, impact ratings are defined, namely :

F1	Loss of sales, orders or contracts	Financial impact	*20% +*	*11% to 20%*	*6% to 10%*	*1% to 5%*	*Less than 1%*

Figure 25: [16] IRAM, F1 part of BIA sheet.

At the start of the BIA, a system profile is described that identifies information of the system, type of system, number of users, and other factors.
The BIA focusses on three aspects of information: confidentiality, integrity and availability. For each of these aspects, the business impact is defined in four areas: Financial, Operational, Customer-related and Employee-related (in line with Balance Scorecard perspectives). For each business impact, its type is assessed for what the impact will be (e.g. the loss of sales orders when unauthorized information is disclosed). The most serious business impact for a type will result in the overall rating of the system (e.g. if the highest impact is for loss of sales, between 6% to 10%, then the overall rating will be C, or medium-impact).

Example: For the application assessed, a BIA is performed stating what the impact of an event will be relating to Confidentiality, Integrity and Availability. Assume that the Confidentiality and Integrity of the system are rated as high.

Overall rating					
	A	*B*	*C*	*D*	*E*
In summary, taking into account the ratings noted above and any other consequence, what is the most serious impact which would arise from unintended or unauthorised disclosure of information? (This would normally be at least as high as the highest individual rating)	Very high	High	Medium	Low	Very low
			X		

Figure 26: [16] IRAM, Overall rating of BIA sheet.

When the overall rating for the impact of confidentiality, integrity and availability is assessed, the system has a rating that is input for the T&VA that can result in a detailed or a standard T&VA.
The T&VA is a rather structured approach of 49 threat types in the areas of external attack, internal misuse and abuse, theft, system malfunction, service interruption, human error, and unforeseen effects of change.
The likelihood rating is determined from the threat rating and the vulnerability rating.

User guide — **Likelihood Rating** — Edit Likelihood reference table

Ref.	Threat type	Overall threat rating (From Threat Assessment form)	Overall vulnerability rating (From Vulnerability Assessment form)	Likelihood rating (The likelihood of an information incident occurring (automatically determined using the Likelihood reference table)	Explanatory comments
External attack					
T2	Hacking	Medium	High	High	

Figure 27: [16] IRAM, Likelihood Rating part of T&VA sheet.

The breach of sensitive client data can be related to the threat "Unauthorized disclosure of information". The inaccurate calculation can be related to the threat "Software malfunction". For the example, assume that they are both rated as high. When the likelihood is also high, we can derive from the IRAM tool that such a vulnerability will have a high impact on the business.
However, is this true? Imagine that some less-sensitive data were disclosed. Then you cannot refer to the overall business impact, but have to review it more in detail.

5.3 ISO 27005

The information security risk management process phases are shown in the Figure 28 at the right.

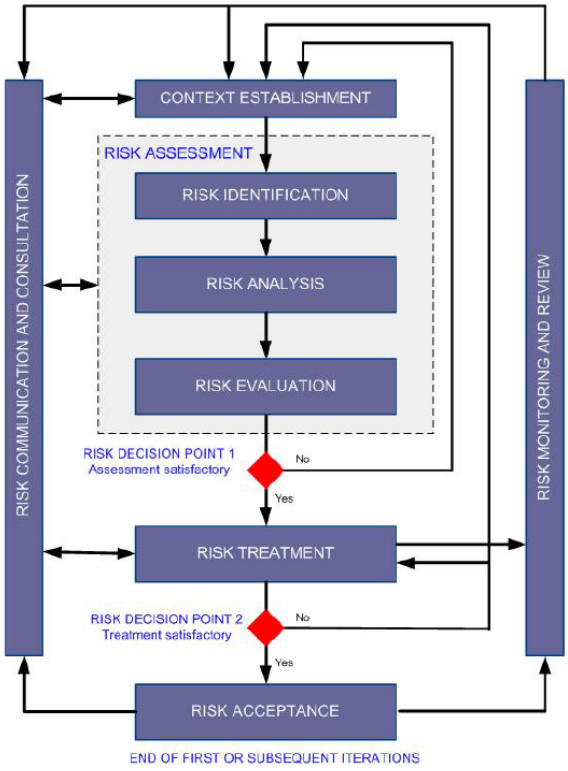


Figure 28: [1, pp 8] ISO 27005, Figure2 Illustration of an information security risk management process

In the context-establishment phase, the impact criteria should be developed and specified in terms of the degree of damage to the organization caused by the event [1, pp. 11]:

- Level of classification of the impacted information asset
- Breaches of information security (e.g. loss of confidentiality, integrity and availability)
- Impaired operations (internal or third parties)
- Loss of business and financial value
- Disruption of plans and deadlines
- Damage of reputation
- Breaches of legal, regulatory or contractual requirements

In the risk identification phase, the consequences are identified. `
A consequence can be loss of effectiveness, adverse operating conditions, loss of business, reputation, damage, or other effects [1, pp. 16].
During the risk analysis, the business impact upon the organization that might result from possible or actual information security incidents should be assessed, taking into account the consequences of a breach of information security such as loss of confidentiality, integrity or availability of the assets [1, pp. 18].
Risk evaluation considerations should include [1, pp. 20]:

- Information security properties: if one criterion is not relevant for the organization (e.g. loss of confidentiality), then all risks impacting this criterion may not be relevant
- The importance of the business process or activity supported by a particular asset or set of assets: if the process is determined to be of low importance, risks associated with it should be given a lower consideration than risks that impact more important processes or activities

Example:
In the Risk Identification phase, the consequences of the vulnerabilities in the software must be identified. The breach of sensitive data will lead to reputation loss, and the inaccurate calculation will result in adverse operating conditions.
Risk analysis phase: the business impact upon the organization from loss of confidentiality of customer data and integrity of calculation of premium has to be assessed.
During the risk evaluation, it should be considered how important the business process is that is supported by the software having the vulnerabilities.

5.4 SABSA

The SABSA Model is built of six layers describing how security architecture is created and based on the Zachman model for enterprise architecture [17, pp. 33]. For this chapter, where we analyze how IT and business risk can be related, the Contextual and Conceptual Architectures are most relevant. These layers contain the link relating IT to business risk.

Business View	Contextual Architecture
Architect's View	Conceptual Architecture
Designer's View	Logical Architecture
Builder's View	Physical Architecture
Tradesman's View	Component Architecture
Service Manager's View	Operational Architecture

Figure 29: [17, pp. 33] SABSA, Table 3-1
The SABSA Model Layered Architecture views

	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECURE	Business Decisions	Business Risk	Business Processes	Business Governance	Business Geography	Business Time Dependence
	Taxonomy of Business Assets, including Goals & Objectives	Opportunities & Threats Inventory	Inventory of Operational Processes	Organisational Structure & the Extended Enterprise	Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Time dependencies of business objectives
CONCEPTUAL ARCHITECTURE	Business Knowledge & Risk Strategy	Risk Management Objectives	Strategies for Process Assurance	Roles & Responsibilities	Domain Framework	Time Management Framework
	Business Attributes Profile	Enablement & Control Objectives; Policy Architecture	Process Mapping Framework; Architectural Strategies for ICT	Owners, Custodians and Users; Service Providers & Customers	Security Domain Concepts & Framework	Through-Life Risk Management Framework

Figure 30: [23, pp. 16] SABSA White Paper, Table **3:** SABSA MATRIX

The complexity of business environments can prove destabilizing for a business if the dependencies of the processes and mechanisms are not understood [17, pp. 58]. SABSA uses a system approach to decompose business information systems by presenting a relatively simplified view on each level of decomposition [17, pp. 56-58]. The system approach is used to focus on the critical concepts that assist in understanding a system. Additionally, the objectives of the application should be aligned with the overall strategic business requirements [17, pp. 62].

Relevant parts of the Contextual phase [23, pp.10]:

- **What?** The business, its assets to be protected (brand, reputation, etc.) and the business needs for information security (security as a business enabler, secure electronic business, operational continuity and stability, compliance with the law, etc*.). Example: Reputation and operational stability.*
- **Why?** The business risks expressed in terms of business opportunities and the threats to business assets. These business risks drive the need for business security (enabling eBusiness, brand enhancement and protection). *Example: Threats are disclosure of confidential data and operational process malfunction.*
- **How?** The business processes that require security (business interactions and transactions, business communications, etc.). *Example: Business process policy administration requires security.*

Relevant components of the Conceptual phase [23 pp.11]

- **What** you want to protect, expressed in terms of Business Attributes.
  Business risk is expressed as the opposite of a business virtue, which is defined as something to be protected and upheld. In SABSA, these are called Business Attributes and are arranged in seven classes: User, Management, Operational, Risk Management, Legal and Regulatory, Technical Strategy, and Business Strategy Attributes [19, pp. 87-88]. *Example: The Risk Management Attributes of Confidential and Integrity-Assured and/or the Operational Attribute Error-free should be included in the conceptual architecture.*

- **Why** the protection is important. Control and enablement objectives are derived directly from an analysis of business operational risks and are a conceptualization of business motivation for security. *Example: Control objectives should enhance the attributes Confidential and Integrity-Assured.*
- **How** you want to achieve the protection, in terms of high-level technical and management security strategies and a process-mapping framework through which to describe business processes. Fit these all together to fulfil the overall strategic goals of the business. *How is Confidential and Integrity-Assured incorporated?*

The SABSA Risk Assessment Method contains the following steps within the Contextual architecture [17, pp. 205- 209]:

1. Business Drivers and Business Attributes (Assets) are translated to a Business requirement for security
2. High-Level Threat Assessment: threats considered relevant for the business.
3. Business Impact Assessment: describe the result from each threat being realized and then rate this on a qualitative scale
4. Vulnerability Assessment: Assessment of the strengths and weaknesses of the systems, processes and culture, without taking controls into account.
5. Risk Category. The risk derived from the impact value and the vulnerability value, without taking any controls into consideration.

In the Conceptual architecture the last three steps are covered [17, pp. 219]:

6. High-Level Control Objectives: Decide which control objectives best express the organization's requirements for security and control.
7. Target vulnerability value: record the target vulnerability after the planed risk mitigation has taken place.
8. Risk mitigation category: New overall risk category that results from the new reduced vulnerability level.

Business risk model											
Business driver	Business Attributes	Business Requirement	High-level Threat	Business Impact	Impact Value	Potential High-Level Vulnerability	Green field Vuln. Value	Green field Risk Category	High Level Control Objectives	Target Vuln. Value	Mitigated Risk Category
The customer is king											
Customer experience impacts competitive advantage or disadvantage	Integrity-Assured	Customers who receive a premium invoice must rely be able to rely that it is accurate.	Customer will doubt if a information received is accurate	Wide loss of customer confidence Regulators investigate operating procedures	H	Inaccurate calculation procedures and lack of control to validate	H	Severe	Establish accurate procedures to test the calculation. Install proper procedures used when caluclations have to be changed.	L	Acceptable
We must comply with the law											
Data protection legislation	Confidential	Must comply with data protection legislation	Customer details disclosed to unauthorised parteis, and this becomes genrally known	Wide loss of customer confidence Prosecution by the regulators	H	Inadequate control over privacy of information	H	Severe	Establish strong logical security surrounding all customer data, in transit, during processing, and in storage	L	Acceptable

Figure 31: Filled in business risk model

Based on this business risk model, the breach of sensitive client data by software flaw is related to the business driver: "data protection legislation". The inaccurate calculation of premium due to software bugs is related to the business driver: "customer experience impacts competitive advantage or disadvantage".

SABSA describes 19 Operational Risk Categories along with some possible mappings to related information security and ICT concerns [17, pp. 437-438]. Enterprise Risk Management (ERM) is defined as management of the entire set of risks facing the enterprise: reputation risks, strategic risks, financial risk and operational risk [17, pp. 456].

5.5 No excuses: A business process approach to managing Operational Risk

Risk is defined as: an unknown event undertaken with the expectation of reward. Credit risk can be exemplified with the following description: we buy a stock and hope that the price will rise. Operational risk is different: the only “reward” from successfully managing or mitigating an operational risk is the reduction of a potential loss [11, pp. 21].
Operational risk is (Basel II): The risk of loss resulting from inadequate or failed internal process, people and systems, or from external events [11, pp. 21]. It is typically associated with the following types of potential loss: People (e.g. mistakes), Process (e.g. deficiencies in procedure), Systems (e.g. systems breakdown), and External (e.g. third party actions) [11, pp. 22].
Operational risk includes legal risk, but excludes risks such as market or credit risk, strategic business risks, and reputational risks. Enterprise Risk Management (ERM) is broader than Operational Risk Management (ORM). It consists of [11, pp. 29]:

1. Strategic. High-level goals and objectives for the organization
2. Operations. Effective and efficient use of an organization’s resources
3. Reporting. Reliability of reporting by the organization
4. Compliance with applicable laws and regulations

Common types of operational risk are [11, pp. 33-38]: Internal fraud, External fraud, Legal and Liability losses, Noncompliance with regulations, Processing errors, Physical security breaches, Information security breaches, System failures, Disaster recovery and business continuity, and Inappropriate business practice.

Understanding ORM begins with an understanding of Business Process Management (BPM). BPM refers to aligning processes with organization strategic goals, designing and implementing process architecture, establishing process measurement systems that align with the organizational goals, and educating and organizing managers so that they will manage processes effectively [11, pp. 52].
BPM is an approach to manage the life cycle of a process by [11, pp. 65-70]:

- *Design/ Redesign Process:* After recognition of a gap in the current process model with the current or new business goals or performance expectations.
- *Model/ Simulate Process* that produces some value-adding outputs, or information signals from inputs. From high-level model into its sub-processes.
- *Deploy/ Execute/ Monitor Process* involves the development of detailed standard operating procedures for processes that are largely manual in nature.
- Software may need to be developed, tested and installed to support automated processes.

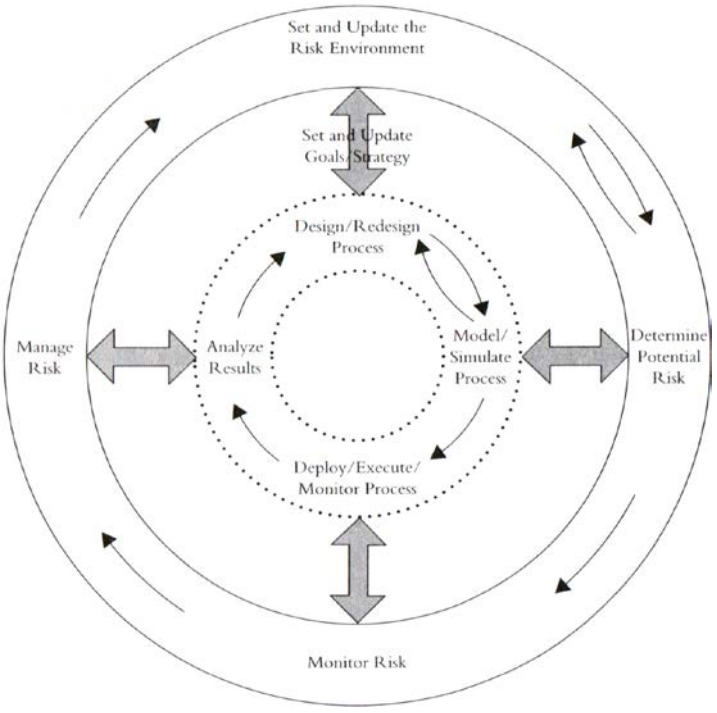


Figure 32: [11, pp. 78] No Excuses, Exhibit 4.2:ORM and BPM Integrated Framework

- *Analyze results* by comparing the actual results gathered to the expected results based on the business goals that were established as targets for the process. These goals are often set as service-level agreements (SLAs).

ORM and BPM frameworks should be combined and integrated. Reporting of process and control deficiencies and of operational risk are not separate components, but rather are activities embedded in every component [11, pp. 78].

The board or executive committee should add its own risk appetite and risk objectives to its list of business goals and objectives [11, pp. 81]. Simulations of the business process should discover possible control defects and operational risks, and identify realized losses or risks that have been experienced for similar processes in other parts of the organization or in other organizations. Typical risks to be discovered are system, application or process failure [11, pp. 101].

This part of the method can be used to relate the breach of sensitive client data by software flaw and the inaccurate calculation of premium due to software bugs to the business process that is impacted.

Integrating ORM and BPM involves the following four components [11, pp. 161]:

1. Setting and revising the risk environment linked to the setting of business goals and designing/redesigning the business process.
2. Modelling and simulating the business process and determining potential risks.
3. Executing and ongoing monitoring of the business process linked to observing and checking of controls for deficiencies, failures, and risk issues.
4. Active risk management linked to the analysis of the business process.

Technology component of the problem:
To the extent that developers of technology solutions, either in your organization or in your clients' and suppliers' organizations, have access to systems that drive your automated processes and have access to data concerning your financial accounts, employees and customers, the potential for your fraud both internal and external, or disruption to your business and the execution of its processes, is one that needs to be considered as seriously as IT service life cycle risk [11, pp. 185].

5.6 Sources of risk from different environments

The review of the IRAM, SABSA and No Excuses material discloses that operational risk does not include the reputational risks nor part of the regulatory risk. Part of the regulatory risks will be related to the business process, while others stem more from the broader enterprise perspective *(e.g. FEC reviews to determine whether bank account numbers are associated with terrorist organizations will relate to the business process. However, the measures to be taken to protect privacy data adequately are not related to the business process but to the broader environment).* Thus, for software vulnerabilities, risk stems from the broader enterprise environment. In addition to these risks, an organization has also to secure their entire IT environment to provide a secure environment for all of their IT assets. We call this the IT generic risk. This is the risk that frameworks like COBIT5 and ISO/IEC 27005 seek to mitigate. From high-level business goals and objectives, goals for the IT environment are set. Ensuring that these goals are not at risk will guarantee that the IT environment can support the needed level for the business environment.

Four sources of risks or impacts that are relevant for evaluating software can be determined:

1. Operational risk related to business processes
2. Regulatory risk effects of non-compliance with the applicable laws and regulations
3. Reputational risk related to the brand of the organization
4. IT generic risk stemming from the non-compliance of software with technical standards that leads to reduction of the overall information security

Examples of vulnerabilities that are related to these risks are discussed in Chapter 6. Figure 33 provides an overview of the different contexts that will affect the evaluation of software security.

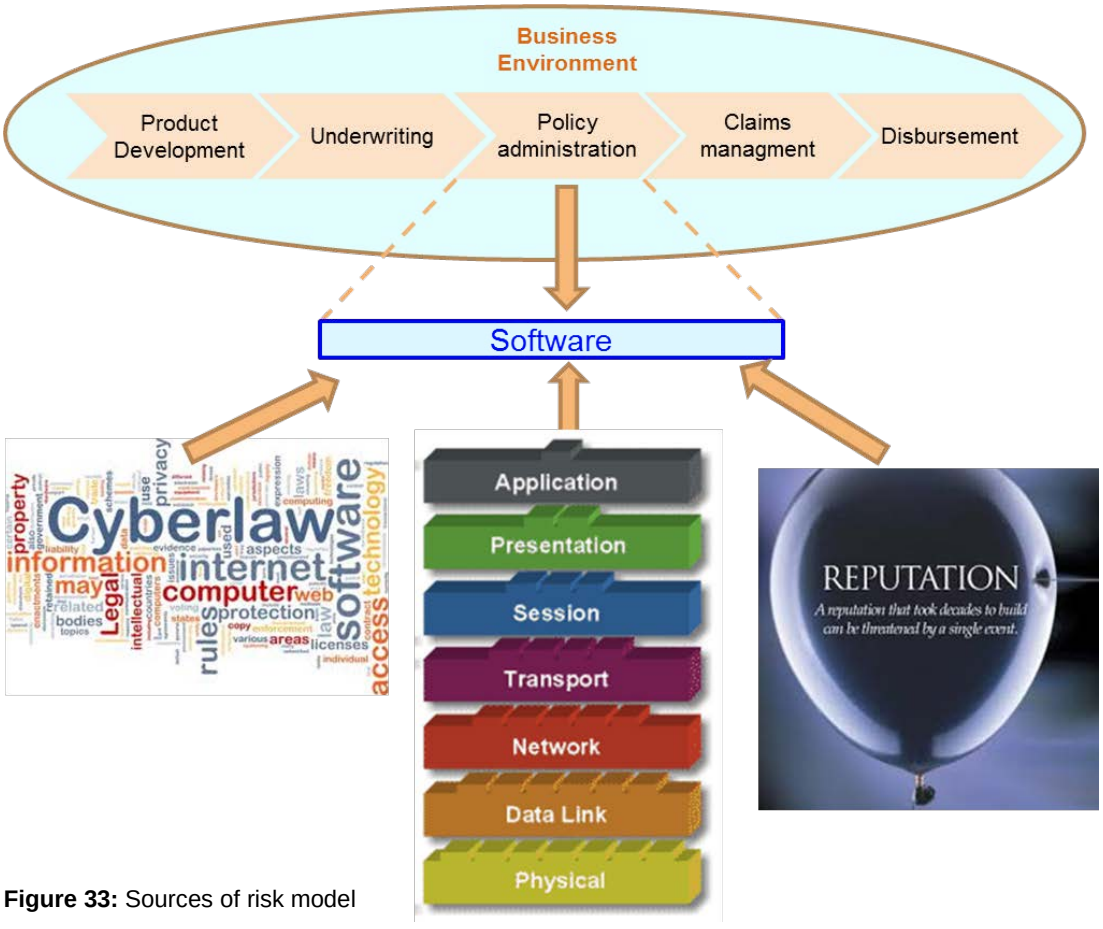


Figure 33: Sources of risk model

5.7 Critical comparison of the IT Security and Risk methods

The purpose of this chapter was to investigate whether methodologies from the field of IT Risk/ Security Governance will contribute to a method of classification of vulnerabilities that is understood by business managers who establish priorities. Below, the conclusion of this exercise is presented.

COBIT5 is a goal-oriented framework in which stakeholder needs cascade in enterprise goals, which cascade in IT-related goals, cascading in enabler goals. For each of these enabler goals, a description is provided for how to identify, analyze and respond to risk.
COBIT5 sets goals for certain IT processes directly from enterprise goals. It does not relate enterprise goals via business process goals, to goals for an application. This makes the framework only relevant within this project for IT generic risk stemming from the non-compliance of software with technical standards that leads to reduction of the overall information security.

IRAM, however, is application-driven. In the BIA phase, a system profile is described, and the business impact in the four areas of the Balance Scorecard is assessed for confidentiality, integrity and availability of information. IRAM assesses three sources of impacts that are relevant for evaluating software:

- Operational risk related to the business processes

C1	Delayed deliveries to customers or clients (eg failure to meet product delivery deadlines)	Deliveries delayed by 6 months	Deliveries delayed by 3 months	Deliveries delayed by one month	Deliveries delayed by one week	Deliveries delayed by one day

Figure 34: [16] IRAM, C1 part of BIA sheet.

- Regulatory risk effect of non-compliance to the applicable laws and regulations

C4	Damage to reputation (eg confidential financial information published in media)	World-wide negative publicity	Continent-wide negative publicity	Nation-wide negative publicity	Local negative publicity	Minor negative publicity

Figure 35: [16] IRAM, O4 part of BIA sheet.

- Reputational risk related to the brand of the organization

O4	Breach of operating standards (eg contravention of regulatory standards)	Closure of building or operation	Serious sanctions imposed	Significant sanctions imposed	Moderate sanctions imposed	Minor sanctions imposed

Figure 36: [16] IRAM, C4 part of BIA sheet.

The vulnerabilities in software can easily be related to the rating of the application. IRAM provides useful input for the aim of this project.

However, in daily practice, the assessment of these different levels in one session can be a disadvantage. Assessing the business process levels in the BIA phase should be accomplished by managers of that process. Assessing the reputational level should, however, be performed with high-level management for the organization as a whole and not for each business process separately. Reputational risk must be assessed on an organizational level and, for process-specific regulations, on a business process level.

The ISO/IEC 27005 established impact criteria in the context phase. In this phase, there is no connection mentioned with the goals of the business process. Actually, the impact of most of the events mentioned that can cause damage should be driven by the goals of the business.
The risk evaluation considerations should, amongst others, include the importance of the business process. Stated in this resource is the principle that if the process is of low importance, the associated risk should be given lower consideration. This is a statement that appears logical, but is not necessarily true. A process of low importance that uses sensitive privacy data can effect a high risk if this privacy data is discovered.
ISO/IEC 27005 acknowledges the need to relate vulnerabilities to business process risk. Since the goals of the business process are not the starting point for the framework, it cannot be used to relate software vulnerabilities with business (process) risks.

The “what, why and how” of the contextual and conceptual phases of SABSA describe the business assets to be protected and their requirements for information security. This will lead to a list of business attributes defined for the organization. Control and enablement objectives are derived from an analysis of business operational risks and provide the motivation for security. In the “how” parts, the business processes that require security are defined and how this protection is achieved is described.
The business attributes are selected from a list of 85 attributes that is part of the SABSA framework. One would expect that the analysis begins with the business process and security, and that other attributes would be introduced at the business process level.
However, SABSA claims that it can be used as justification of control in, for example, the component security architecture back to the contextual security layer. This method should also be useable to relate security vulnerability in software to the contextual layer, where the impact and threat on business assets can be found.
The aim of this project is to relate software vulnerabilities to business risk. The business risk model of SABSA links a vulnerability with high-level enterprise risk. It lacks the step from vulnerabilities via business process risk to higher-level business risks, to be useful to explain the business process risk of a software vulnerability to a business manager.

The book “No Excuses” makes clear that Enterprise Risk Management (ERM) is broader than Operational Risk Management (ORM). ORM includes legal risk, but excludes risks such as market or credit risk, strategic business risks, and reputational risks [11, pp. 22]. This implies that we do not address the entire range of risk when we align our IT risk management with only ORM. We miss the reputation risk, which is an important risk component.
In “No Excuses”, Business Process Management (BPM) is combined with ORM. This theory is used to make the distinction between the different sources of risks or impacts that are relevant for evaluating software.
Simulations of the business process can discover possible control defects and operational risks, and identify realized losses or risks that have been experienced for similar processes. This can be used to link discovered software vulnerabilities to process risk, if they can be ranked as operational risks.

6 Relating software risks to business (process) risks

6.1 Introduction

As mentioned in the introduction of this thesis, the ultimate aim of this project is to establish a methodology that will bridge the gap between the technical vulnerabilities and risks understood by the business. In this chapter, a related method developed during this project will be presented.
Based on a hypothetical environment and a couple of hypothetical software vulnerabilities, a method established from the material in previous chapters will be used to link software vulnerabilities with business (process) risk.

6.2 Developed model for software security vulnerabilities

The model consists of four basic parts. First, the environment where the software vulnerability is discovered should be thoroughly understood. Before continuing with the method, the necessary components of risk are discussed. Second, the probability of a threat exploiting the software vulnerability must be assessed. Third, the impact of a probability has to be assessed. Finally, the resulting risk for the (broad) business environment should be established.

6.2.1 The environment

Before we are able to judge how severe a vulnerability is, we must closely inspect the environment in which the vulnerability is found. This is required to be able to evaluate the discovered vulnerabilities of a (sub)system, taking into account the broad environment with its controls and requirements (see Section 4.2).

Where the system belongs to

An application is not an environment on its own, but is part of a broader context that will provide requirements for an application. The ISO/IEC 27034 describes that applications have Business, Regulatory and Technological contexts [5, pp. xxvi].

Each system will have a business context: what does the business do, what products does it make, or services does it provide, what type of firm is it, what is the culture, what is the organizational structure. This influences how software risks are perceived. A system supports a business process (step) that is part of the total business process of an organization. The business processes should be designed in line with the business goals for the processes that are derived from the mission of the organization. The system should support the business requirements of the process.

The policy administration system supports the process step policy administration within the whole insurance process and has its own goals. To achieve the goals, controls will be defined to manage the process for the specific steps, e.g.:

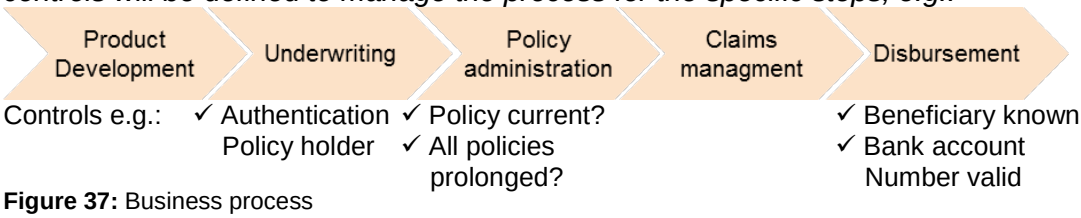


Figure 37: Business process

Controls can be automated or manual. In the above example, the authentication check, determining whether a person is who he says he is, will be a manual check before the policyholder is entered into the system. Checking whether a policy is current might be programmed in the policy system.
It is important to oversee the entire control structure when performing a review of software. Controls that might be lacking in the object under review can be covered in other (sub)systems (not covered by the review) or manually performed.

However, there is also a business context that is broader than the business process environment with, for example, reputational risk and strategic business risk. The latter may be less important for secure software, but reputational risk is heavily related to the security quality characteristic of confidentiality.

The impact on reputational risk of a breach that leads to disclosure of policy data to unauthorized persons is a risk that must be taken into account when securing software. However, how severe a breach will be is to be decided by management.

Next to the business process environment, there is the wider environment that includes the regulatory context and technical context.

The policy administration system can, for example, include privacy data about the policy holders. The storage and processing of these data must conform to the Data Privacy Act. The regulators can also require that data of the policy administration system comply with certain quality requirements.

The technological context can include that the developed software must comply with some technical standards to guarantee the information security for the entire company. For example, the internal network is not encrypted, and messages need to be encrypted by the policy administration system before they are sent.

What the system does

The assessor should acquire an overview of the system elements, including their functions and relationships. Most information systems today are highly complex systems containing many subsystems that are linked together, e.g. an ERP system. SABSA describes a system approach to decompose highly complex business information systems. This provides a method to mask complexity by presenting a relatively simplified view on each level of decomposition. System components are also seen as a black box, simply showing that the box has inputs and outputs when viewing a (sub)system. At other times, the inner of a black box can be examined if more granularity is desired [17, pp. 56].

In the example environment, the policy administration system is decomposed into five subsystems. This provides the possibility to focus, for example, on the calculation subsystem and to view the other four subsystems as inputs and outputs of this subsystem.

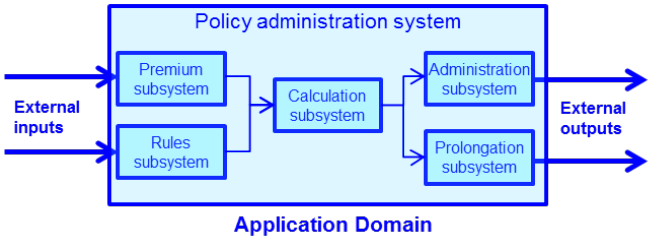


Figure 38: Overview of a system

How the system acts

At this point, the data and control flow of the (sub)system are analysed. How does the inner system process the transactions, and which components of the process have to be performed in parallel or simultaneously?

Related to the policy administration system, for example, how does the calculation subsystem designed (e.g. which parameters are important for the calculation, how is the calculation be processed, does it need manual input)? Also, what are the relations to other processes; for example, the calculation cannot be initiated before the premium subsystem has ended its process, because the latest premium must be known for the calculation.

Provided with knowledge about the environment, we can assess how severe a vulnerability is, knowing the environment of the software.
We can use this information to assess the threat, e.g. when we know that a fraudulent change of a beneficiary in the policy system will be discovered in the payment system. The threat inhibitor of fear to be discovered will be set to high.
We also can use the environment information to assess the impact. We now know what the vulnerability is related to and can, for example, link the impact with the process the software supports.

6.2.2 Risk components

According to the NIST definition:
Risk is a measure of the extent to which an entity is threatened by a potential circumstance or event, and is typically a function of [21, pp.6]:

(i) the adverse impacts that would arise if the circumstance or event occurs; and
(ii) the likelihood of occurrence.

SABSA states that the *likelihood* of occurrence is a product of two separate probabilities [17, pp. 454]:

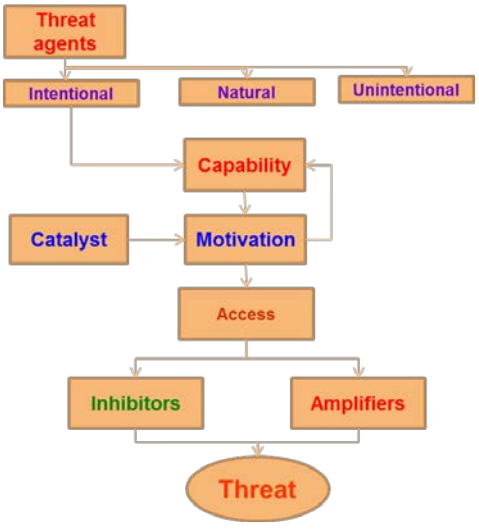
- The probability that the event will occur (the level of threat)
- The probability that when the event occurs, the control will fail (the level of vulnerability)

Since we start with a vulnerability that is discovered during a software penetration test, we will only assess the threat. The probability that the control will fail is another issue that is not part of the method developed within this project.

From the definition of risk, we know that before we can explain the risk to a manager, we need to know the vulnerability that is likely to occur and need to know what the impact of this occurrence is, using the risk language that he will understand.

6.2.3 Probability of the threat occurring

To assess whether a threat exists for the vulnerability, the following model will be used:



Natural threat agent	Fire, flood, power failures, and other.
Unintentional threat agents	Parties that cause damage or loss of service without direct intent.
Intentional threat agents	Those parties that would knowingly seek to make a threat manifest
Capability	Capability to conduct and sustain an attack or totally destroy the system and any replacement
Threat catalyst	Those factors or actions that cause an attack to be initiated at the time and on the target selected.
Threat agent motivators	Factors and influence s that motivate a threat agent.
Threat inhibitors	Any factor that decreases either the likelihood of an attack taking place or being successful
Threat amplifiers	Any factor that increases either the likelihood of an attack taking place or being successful

Figure 39: [9, pp. 34-38 Information Assurance: Security in the Information Environment, Resume paragraph 3.3 Threat components applying to malicious threats.

In the current business environment, most reports will only describe some of the threat agents that exist (if any). Some reports present threats to the level of access, but the level of inhibitors and amplifiers are rarely mentioned. To be able to include these factors, one must have knowledge about the environment of the vulnerability – threat combination as discussed in Section 6.2.1.

Figure 40 presents a model to score threats. It looks like a mathematical model, but is not intended for that purpose. The model aims to serve two major objectives:

- A tool to provide insight about which parameters entail that a threat is viewed at a certain level (e.g. high threat).
- In the bottom part of the model, the score for threat inhibitors must be set, and the business can assist with this. The information that is input for these decisions can be retrieved from the environment information (see Section 4.2).

Threat score model			
	Possible rating	Vulrability with environment information	Rating
Natural threat agent	0-4		
Unintentional threat agents	0-4		
Intentional threat agents	0-4		
A = highest of three agents			
Capability	0-4		
B = score agent + capability			0
Threat catalyst	0-4		
C = B + Catalyst			0
Threat agent motivators	0-4		
Access: D = C + motivators			0
Threat inhibitors	0-16		
E = D - inhibitors			0
Threat amplifiers	0-16		
Overall rating threat level			0

Figure 40: Threat Score Model

6.2.4 Impact of the vulnerability

In Section 5.6 four contexts that influence the impact of software vulnerabilities are mentioned that are relevant for evaluating software:

1. Operational risk related to the business processes
2. Regulatory risk caused by non-compliance with the applicable laws and regulations
3. Reputational risk related to the brand of the organization
4. IT generic risk stemming from the non-compliance of software with technical standards that leads to reduction of the overall information security

Figure 33 in Section 5.6, provides an overview of the different contexts that will affect the evaluation of software security.

The method, developed by this project, uses the IRAM method introduced in Section 5.2 for the assessment of the impact of a software vulnerability for the first three sources of impact. COBIT5, described in Section 5.1, will be used to evaluate the impact of IT generic risk stemming from the non-compliance of software with technical standards that leads to the reduction of the overall information security.

6.2.5 Risk evaluation

The risk of a software vulnerability will be evaluated based on the assessed likelihood of the threat and the impact on the business when the vulnerability occurs.

The table in Figure 41 will be used to evaluate the business risk. The name of the software vulnerability will be added in column A. Threat level in column B will be copied from the Threat Score Model, and the Impact will follow from the IRAM method or COBIT5 Impact assessment. The Risk value in Column D will need to be looked up in the 5 x 5 matrix in Figure 42.

	A: Vulnerability	B: Threat level	C: Impact	D: Risk
1				
2				
3				
4				

Figure 41: Risk evaluation

The 5x5 matrix is a matrix of an existing risk model that was adjusted for the present purpose.
The five levels of probability threats are not based on a mathematical calculation, but rather are based on estimates the author made and can be changed when necessary.

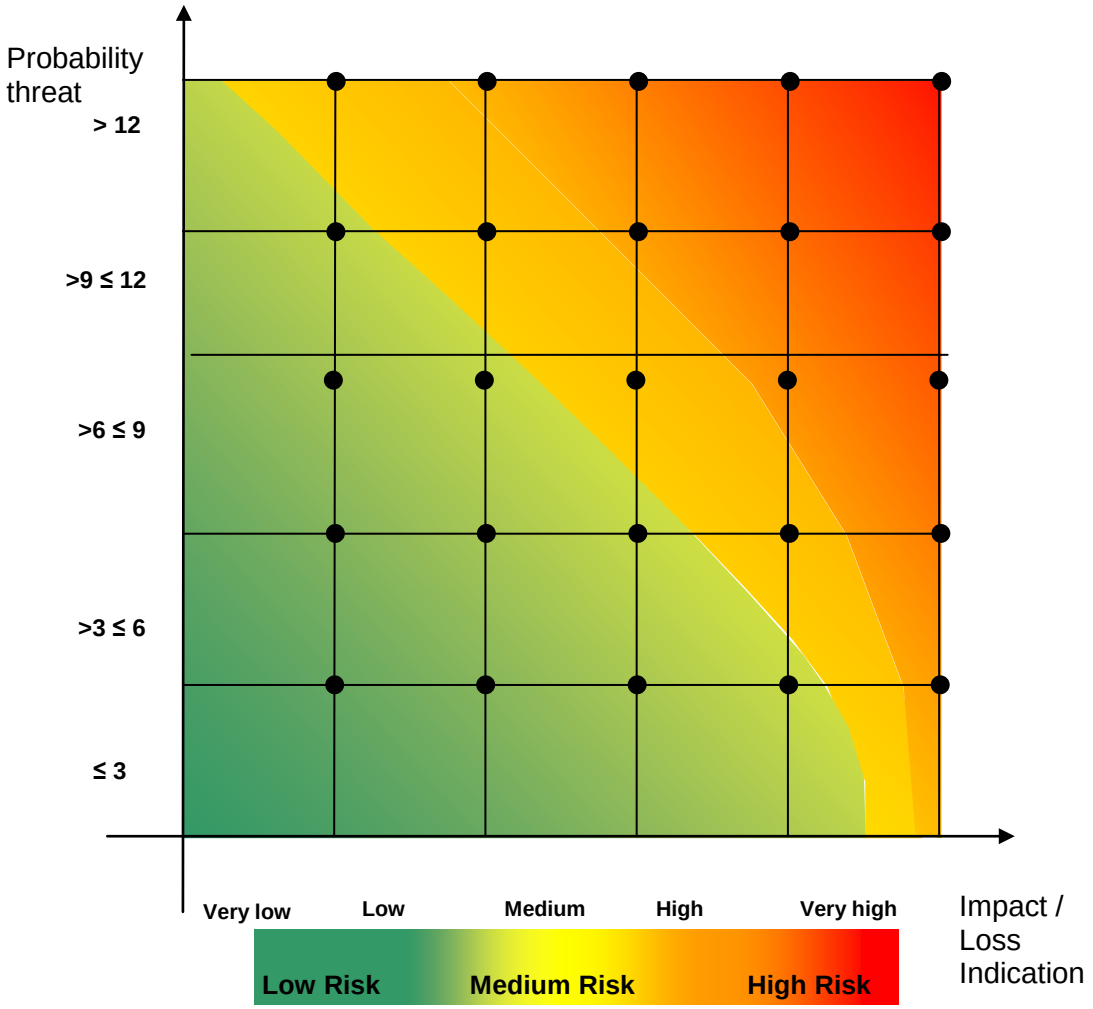


Figure 42: 5x5 matrix

Based on the 5x5 matrix presented above, we show the business manager what the effect of the software vulnerabilities will be on the risk profile of the organization. The estimates that are the basis for the assessed risk can be shown to the manager to initiate a discussion about the level of risk in terms the business manager understands.

7 Case study of software vulnerabilities

In this chapter the application of the model developed by the project will be validated based on four cases studies. The case studies present a hypothetical software vulnerability within a hypothetical environment and evaluate if the model can be used to relate software vulnerabilities with business risks.
The case studies will produce material that makes the estimates of the assessor visible. During the discussion with the business manager this material can be used to create mutual understanding about the risk of a software security vulnerability.

7.1 Environment

The environment of a hypothetical insurance organization is illustrated in the picture below. The following components are shown:

- A wider environment that, for example, will impose legislation that will force the organization to change the retirement age from 65 to 67.
- A business environment that influences how software risks are perceived. For example, within an insurance organization, people tend to be driven to ensure that there is no risk. Also, cultural goals such as “we do what we promise”.
- A process with goals that will execute a certain activity for the organization to contribute to the organizational goals. For example, the insurance process will process an inquiry of a customer for a pension policy to ultimately yield the payment of pension.
- An application system that will support a (sub)process to attain its goals. In the example below, the policy administration system has the same scope as the process. The goals for the system will be equal to the process.
- Application subsystems that will perform a particular task within the whole application domain. For example, the goal of a calculation subsystem might be to calculate the premium of a customer accurately and on time.

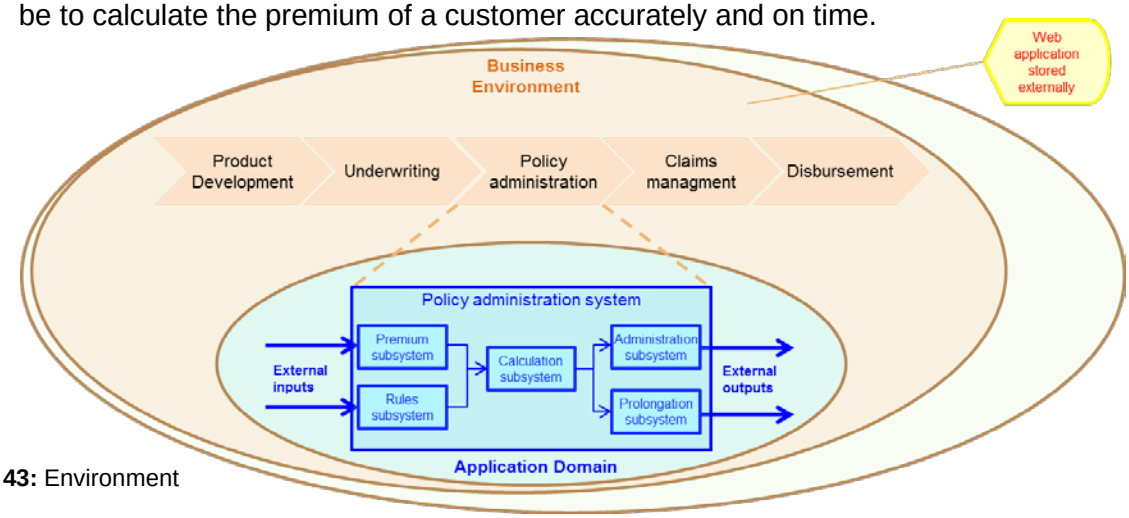


Figure 43: Environment

7.2 Case study software vulnerabilities

Based on the contexts mentioned in Section 6.2.4, we present four kinds of software vulnerabilities, discovered during a hypothetical software penetration test:

1. *The software of the premium subsystem includes a bug that opens a back door to enable an attacker to change the beneficiary of the payment of a pension.*
2. *An externally-hosted website includes a software flaw and makes it vulnerable for injection of pictures that compromise the organization.*
3. *Due to a software bug, the SQL statement to retrieve customer privacy data can be misused to get a dump of all of the customer data within the other system.*
4. *The software is developed in such a way that it requires a particular port within the firewall to be opened in order to function properly.*

7.3 Case study probability of the threat occurring

For the four examples stated in Section 7.2 the proposed model below is filled in. With the results of the model the assessor and the business manager can discuss whether they agree with the estimates made and thus with the overall threat level.

Threat score model		**Vulnerability 1**		**Vulnerability 2**		**Vulnerability 3**		**Vulnerability 4**	
	Possible rating	The software of the premium subsystem includes a bug that opens a backdoor to enable an attacker to change the beneficiary of the payment of a pension	Rating	Externally hosted website is includes a software flaw and makes it vulnerable for injection of pictures that compromise the organization	Rating	Due to a software bug the SQL statement to retrieve customer privacy data can be misused to get a dump of all the customer data within the other system.	Rating	The software is developed in such a way that it needs some port within the firewall to be opened to function properly.	Rating
Natural threat agent	0-4	N/A		N/A		N/A		N/A	
Unintentional threat agents	0-4	N/A		N/A		N/A		N/A	
Intentional threat agents	0-4	Disaffected employee	4	Hackers or pressure groups	3	Hackers or unauthorised staff	3	Hackers	4
A = highest of three agents			4		3		3		
Capability	0-4	Staff with no development knowledge are not able to understand what need to be performed. Developer have no access to the production environement	0	Software, technology, facilities, education and training, methods and books and manuals are easy to retrieve	4	Hacker have the knowlegde to perform the attack. Normal users will be less knowledgable.	2	Hackers have the technology and the knowledge to perform port scans to search for vulnerabilities.	4
B = score agent +			4		7		5		4
Threat catalyst	0-4	Developer get access right for production environment to solve a production issue.	4	Dispute about a policy hold with the insurance company owning the website	2	N/A	0	Technology change can effect that a port that was e.g. open for an application is now widely open	2
C = B + Catalyst			8		9		5		6
Threat agent motivators	0-4	Personal gain.	4	Try to get even.	3	Power for the hacker Curiosity for the normal user	3	Power to be able to place the hack.	3
Access: D = C +			12		12		8		9
Threat inhibitors	0-16	Deveoper will loose his job Change will be discovered in payment system	10	Fear to get captured by police force.	4	Viewing of information is hardly monitored nowadays	0	The use of the ports can be monitored and the hacker can be noticed.	2
E = D - inhibitors			2		8		8		7
Threat amplifiers	0-16	N/A	0	Target vulnerabilities, script available to execute it.	2	Access to information is a strong amplifier.	2	A server within a big Insurance firm could be a target to be proud of	2
Overall rating threat			2		10		10		9

Figure 44: Case study: Threat Score Model

Based on outcomes of the examples in the model, we can draw some conclusions. Changes in the circumstances can have a huge impact, such as in the first example. The vulnerability could not be exploited, because the developer was not able to access production systems. The threat catalyst (i.e. that the developer gains access to the production environment to solve a production problem) changed that situation, and now he was able to exploit the vulnerability.
Although, this vulnerability appears very severe, the threat inhibitor that the change of the beneficiary will be discovered in the payment system, makes it useless to exploit the vulnerability.
This, however, requires that one has knowledge of the broad environment of the software to be able to assess the threat on its real merits.

7.4 Case study impact of the vulnerability

Based on the results of the threat assessment, only the impact of the threats of examples 2, 3 and 4 needs to be assessed. However, for illustration purposes, the project will assume that Threat 1 will also be likely to occur.
As mentioned, there are four contexts that influence the impact of software vulnerabilities: operational risk related to the business processes, regulatory risk, reputational risk, and technology risk. For each of these categories, an example is included in this Section to assess the impact based on the IRAM method (first three examples) and the COBIT5 framework for the last example.

7.4.1 Bug in software of the premium subsystem

Definition of the example vulnerability:
The software of the premium subsystem includes a bug that opens a back door to enable an attacker to change the beneficiary of the payment of a pension.

The impact of this vulnerability is related to the business process, so we need to relate from the discovered vulnerability back to the impact on the business process. At this point, we need to use information about how the software acts, what the system does, and understand the environment of the system. We will use the following environment:

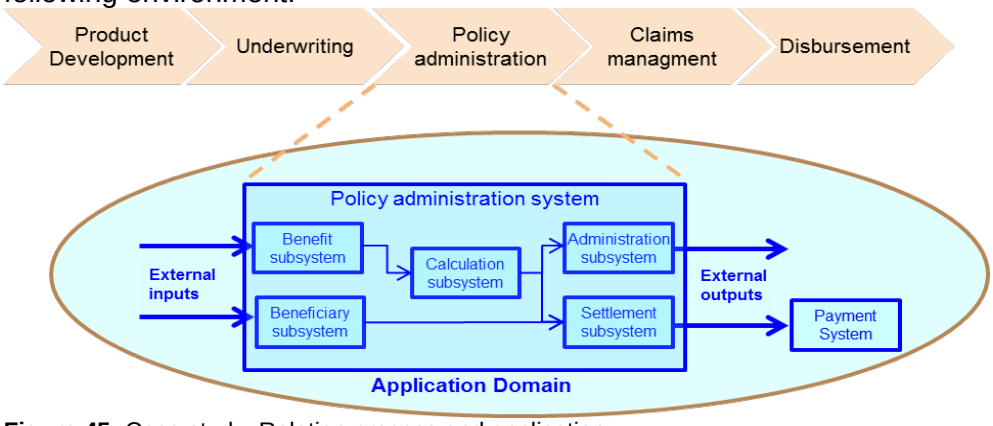


Figure 45: Case study: Relation process and application

Information about how the software acts should normally be collected during the software security assessment phase. During the Risk Analysis stage of the Microsoft SDL method, a Data Flow Diagram (DFD) is often drafted in which the flow of information is drawn. The DFD can provide knowledge about the impact of the vulnerability on how the software acts.

Changes in beneficiaries are entered by staff of the policy administration department via an input screen of the beneficiary subsystem. The access to this functionality is restricted to several employees. The changes will only be validated if the changes are approved by another staff member (4-eyes check).
However by including a backdoor in the software of the beneficiary a fraudulent user is able to retrieve a shell and enter beneficiary data without the proper authorization and without the 4–eyes check.

It is necessary to know what the system does in order to know why it is important that the piece of software that contains the vulnerability works in the expected manner. What are the relations between the subsystems? What are the input and output relations?

In our system, the Settlement subsystem uses the information of the Calculation subsystem and the Beneficiary subsystem as input. When the input from the Beneficiary subsystem is incorrect, the settlement can then be drafted for the wrong beneficiary.

"To which environment does the system belong?" is the latest important question at system level. The system can receive input from other systems and provide output to other systems. It is also possible that the system will be supervised by a monitoring system. This can involve monitoring of the process flow through the system or monitoring that is more security-related (e.g. if there are no direct changes at the database level).

The Policy Administration system has a user interface for e.g. input of Beneficiary information. Settlements will be transferred to the Payment system, which will distribute the pensions to the customers.

The Business context relevant for the bug discovered is important to know, in order to gain insight into the business impact of the vulnerability. It is important to know which process (part) the system supports. The process will have control objectives that need to be supported by the software if portions of the process are automated.

As drawn in the hypothetical environment, the Policy Administration system supports the Policy Administration process. One of the controls defined for the process is that a Beneficiary may only be changed by an authorized person and must be approved by a second authorized person.

A business manager of the process to which the control objective is related is able to assess the impact of an event that occurs. This assessment should be performed without taking into account the measures that have already been completed in his process (step) or in another process (step). Ultimately, the question would be what the impact is when the vulnerability materializes.

When a Business Impact Assessment is performed for the application of the process, a link with this assessment can be made to rate the impact of a vulnerability.

The impact when a fraudulent person can change the beneficiary should be assessed. This can, for example, be related to the maximum amount of a settlement or a settlement run (e.g. the business-related fraudulent payments as a high impact rating (Figure 46: relevant part of BIA)).

User guide

Business Impact Rating

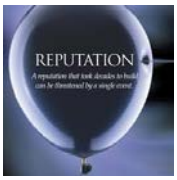
Integrity

Ref.	Business impact type *Business impact of errors in information or of deliberate manipulation of information to perpetrate or conceal fraud (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	*B*	*C*	*D*	*E*	
Financial							
F1	Loss of sales, orders or contracts	20% +	11% to 20%	6% to 10%	1% to 5%	Less than 1%	
F2	Loss of tangible assets (eg fraud, theft of money, lost interest)	$20m+	$1m to $20m	$100K to $1m	$10K to $100K	Less than $10K	If someone is able to change all the beneficiaries of a pension settlement run to him as being the beneficiary.
			X				

Figure 46: Case study: [16] IRAM, F1 and F2 part of BIA sheet.

The business manager can only perform the assessment if it is first explained to him how the vulnerability in the software relates to his process (step) or application.

7.4.2 Software flaw in externally-hosted website



Definition of the example vulnerability:
Externally hosted website includes a software flaw and makes it vulnerable for injection of images that compromise the organization

All applications that an organization uses, internally or externally hosted, should be rated for their impact on the company's objectives. The relevant risk that the organization can face in this situation is reputation risk. This type of risk does not have a direct link to the business processes and should be assessed on an organizational level.

The board members of the hypothetical organization stated in the company's mission that they are against war and do not want to be involved with parties selling arms. Pressure groups who oppose war associate the company with insuring companies that sell arms. They hacked the website and use it as their publishing platform. The damage to the company's reputation is rated as in Figure 47 (relevant part of BIA).

User guide

Business Impact Rating

Integrity

Ref.	Business impact type *Business impact of errors in information or of deliberate manipulation of information to perpetrate or conceal fraud (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	*B*	*C*	*D*	*E*	
Customer-related							
C4	Damage to reputation (eg confidential financial information published in media)	World-wide negative publicity	Continent-wide negative publicity X	Nation-wide negative publicity	Local negative publicity	Minor negative publicity	When web page will present adverse information this will cause broad negative publicity

Figure 47: Case study: [16] IRAM, C4 part of BIA sheet.

7.4.3 Software bug in SQL statement

Definition of the example vulnerability:
Due to a software bug, the SQL statement to retrieve customer privacy data within a policy administration system can be misused to accomplish a dump of all the customer data within the customer relation system where the request is send to.

This vulnerability can be related to the business process, but can also be viewed as a separate regulatory risk category. Besides, the organization can face reputation risk. This type of risk does not have a direct link to the business processes and should be assessed on an organizational level.
Regulatory bodies have established regulations for the use and storage of privacy information and (rather vague) requirements for automated systems. Legislation states that a business should take appropriate technical and organizational measures against unauthorized or unlawful processing of personal data, and against accidental loss or destruction of, or damage to, personal data.

The board members of the hypothetical organization stated in the company's mission that they will support their customers, regardless of the situation. This company-wide philosophy also has an effect on the way the company will handle the data of its customers.
The board members have rated the impact of a breach of compliance with Data Privacy Legislation and Reputation Damage as high.

User guide

Business Impact Rating

Confidentiality

Ref.	Business impact type *Business impact of unintended or unauthorised disclosure of information (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	*B*	*C*	*D*	*E*	
Operational							
O4	Breach of operating standards (eg contravention of regulatory standards)	Closure of building or operation	Serious sanctions imposed	Significant sanctions imposed	Moderate sanctions imposed	Minor sanctions imposed	A breach that will disclose large number of customer data will result in serious fines of the regulator.
			X				
Customer-related							
C4	Damage to reputation (eg confidential financial information published in media)	World-wide negative publicity	Continent-wide negative publicity	Nation-wide negative publicity	Local negative publicity	Minor negative publicity	A breach that will disclose large number of customer data will result in broadly negative publicity.
			X				

Figure 48: Case study: [16] IRAM, O4 an C4 part of Excel sheet.

However, before an impact can be assessed, it is first necessary to have a clear understanding about precisely which privacy data will be disclosed. It is necessary to establish which information, when combined, constitute privacy-related data. This may require a description of the system environment as presented in Section 6.2.1.

7.4.4 How software flaw effect network exposure

Definition of the vulnerability:
The software is developed in such a way that it requires some port within the firewall to be opened in order to function properly.

The attack surface of the entire security infrastructure can be Negatively impacted by a software design flaw. Software can be designed in such a way that it requires certain ports on servers to be opened in order to function properly. This can introduce vulnerabilities on the generic infrastructure level that will also affect other components (including the "crown jewel" applications).
This vulnerability is related to the technical context of the software. The developed software has to comply with the technological standards of the organization. The information technology infrastructure is designed as a layered environment with certain controls on different layers that, as a whole, ensure the security of the environment.
COBIT5 may be used for these processes. For example:

- APO01.08: Maintain compliance with policies and procedures. Establish procedures to maintain compliance with and performance measurement of policies and other enablers of the control framework, and enforce the consequences of noncompliance or inadequate performance. Track trends and performance, and consider these in the future design and improvement of the control framework. One of the activities is: Analyze noncompliance and take appropriate action [4, pp. 56].
- DSS05.02: Manage network and connectivity security. Use security measures and related management procedures to protect information over all methods of connectivity [4, pp. 193].

The application is tested in the test environment using dummy interfaces. During the installation on the production environment, the implementing team recognizes that certain required input does not reach the application. The developers inspect the production system and tell the implementers that a certain port on the firewall needs to be open.
The people who implemented the infrastructure for the application are requested to open the port. They, however, inform the implementers that the installed firewall is operating according to the Operational Security Guidelines and do not want to open the port unless someone accepts the risk. The application owner is the only party willing to accept this risk. He has already paid millions of Euros for the application and wants it to work as soon as possible.

The conclusion is that the application owner should not be able to accept this risk, even when he wants to accept it. He has a strong motivation to not follow the standard and lacks knowledge about the overall security environment. When he accepts this risk, he will only be attentive to the component that he understands, which is his business application environment.

The impact of this kind of vulnerability should not be assessed by someone from the business, but by someone with the technical security knowledge of the comprehensive environment. Information for this kind of assessment cannot be gathered from business impact assessment. This may require a technique such as Fault Tree Analysis to deduce what the "impact" is on the overall security architecture.

For the example we assume that the impact of the vulnerability is rated as medium.

7.5 Case study of the resulting risk

Using the model, the probability of the threat occurring, and the impact of the vulnerability, the risk of the example software vulnerabilities can be assessed.

	Vulnerability	Threat level	Impact	Risk
1	Bug in software of the premium subsystem	2	High	Low
2	Software flaw in externally hosted website	10	High	High
3	Software bug in SQL statement	10	High	High
4	Software flaw effect exposure of network	9	Medium	Low

Figure 49: Case study: Risk evaluation

Based on the risk model, the assessed risk has been completed in the risk column of the table.

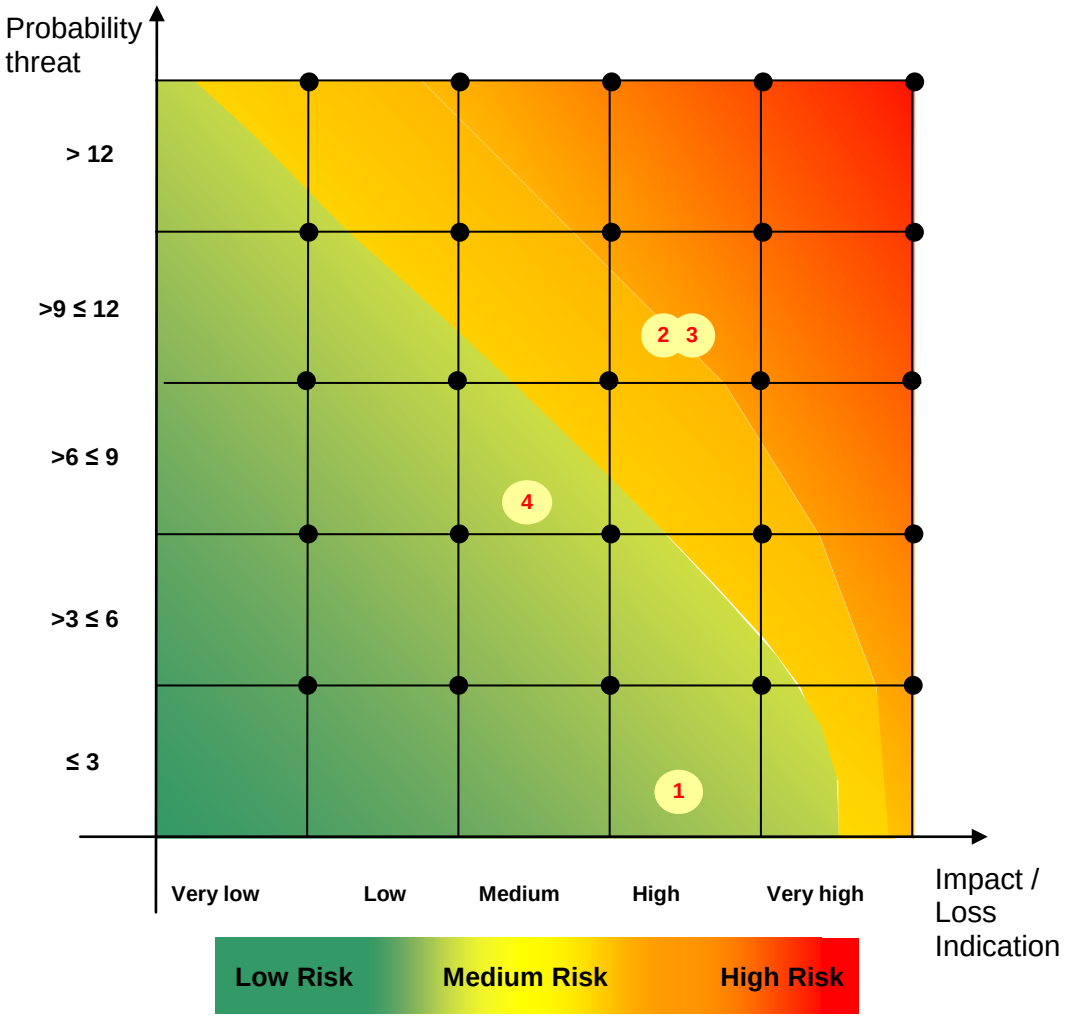


Figure 50: Case study: 5x5 matrix

Based on this model, we show the business manager the effect of the software vulnerabilities on the risk profile of the organization.

8 Conclusion

8.1 Summary of the observations

This chapter presents the main conclusion of the research related to secure software development, computer-related risks, and IT risk/ security governance frameworks.

8.1.1 Conclusion concerning secure software development methods

Software security methodology presents and ranks the priority of vulnerabilities in a rather technical way. Below, the basic conclusions for each method are presented.

The Microsoft Security Development Lifecycle (SDL) [14] is a method that is provides a useful method for evaluating the threats based on Data Flow Diagrams of the system. Risks of vulnerabilities that are discovered following this method were evaluated at the start of the method using the DREAD method, and are currently evaluated using the SBSRS. Both methods emphasize the circumstances or conditions that are required to be present.
The method does not provide useful input to link software vulnerabilities with business risks.

The Software Security: Building Security In [18] resource is a method that recognizes the need to relate software risk to business risks. It states that “Only by practicing risk management and factoring in critical business information about impacts will software security escape the realm of the geeks and have an impact on business [18, pp. 79]”. This method presents examples of architectural risk analysis and abuse cases to identify technical software risks. A Risk Management Framework (RMF) relates these technical software risks to business goals.
The method presents the concept of linking software risk with business risks. However, the RMF connects risk on such a high level that it will not be of practical use to achieve the aim of the present project.

The ISO/IEC 27034 describes a process that can be used to develop secure software. This method also includes other phases of the Application Security Lifecycle that can be used to cover the entire application lifecycle.
The method states that activities cause risks, and that the organization will determine security requirements for mitigating those risks. However, it is not described how the context has to be related to risks that can occur in software. Instead, the method defines a Targeted Level of Trust. The Targeted Level of Trust for an application can be used to select the required security activities that need to be performed for the given application.
For the purpose of this project, this method lacks a connection from the security objectives to the risk that will ultimately affect the business processes.

8.1.2 Conclusion concerning computer-related risks

The present project investigated whether methodologies from the field of general computer risks can contribute to a method of classification of vulnerabilities that is understood by business managers who establish priorities.
The material concerning computer-related risk states that a different perception of risk at the business and software engineering levels is to be taken into account [10, pp. 455-469].
This confirmed the opinion of the author even further that technical software risks are not understood at the business level. The software risk has to be communicated to the business managers in a language that they will understand. The developed method helps the assessor of the software security to provide the business with an

improved view of the reasons why a software risk should be a concern for a business manager.

Defining the software object to be assessed and understanding the contexts in which it belongs is of major importance. If one does not understand the business process and broader business environment, it would be very difficult to assess the real business risks of a software vulnerability on its merits.

Threat trees can be useful resources to identify threats in the business environment and drill down to the software threats. When these threats are known, the likelihood of a threat can be evaluated based on the model in which the capability, motivation and catalysts will be assessed with the inhibitors and amplifiers of the attacker.
The threat score model developed during this project can be used to make the estimates made visible. The model is useful to discuss with the business managers to validate the appropriateness of the risk estimates.

8.1.3 Conclusion concerning material to relate IT and business risk

The book “No Excuses” presents an overview of Operational Risk Management (ORM) and combines it with Business Process Management (BPM). ORM includes legal risk, but excludes risks such as market or credit risk, strategic business risks, and reputational risks [11, pp. 22].This theory is used to make the distinction in the different sources of risks or impacts that are relevant for evaluating software.

In the process of performing the literature search, four sources of risks or impacts that are relevant for evaluating software were determined:

1. Operational risk related to business processes
2. Regulatory risk caused by noncompliance to the applicable laws and regulations
3. Reputational risk related to the brand of the organization
4. IT generic risk stemming from the noncompliance of software with technical standards that leads to reduction of the overall information security

This project developed a method that uses these four sources of impact to relate discovered software vulnerabilities to business risks.

Methodologies from the field of IT risk/ security governance can contribute to a method of classification of vulnerabilities that is understood by business managers who establish priorities. Below, the conclusion for the reviewed methods is provided.

COBIT5 is a goal-oriented framework. Starting with the goals of the stakeholders, it cascades to enabler goals for IT processes. It does not relate enterprise goals via business process goals, to goals for an application.
The developed method uses this framework for the fourth source of impact described above when evaluating software vulnerabilities: IT generic risk stemming from the noncompliance of software with technical standards that leads to reduction of the overall information security.

IRAM is an application-driven approach. The developed method uses IRAM for the assessment of three sources of impact that are relevant for evaluating software security vulnerabilities:

1. Operational risk related to the business processes
2. Regulatory risk caused by noncompliance with the applicable laws and regulations
3. Reputational risk related to the brand of the organization

A Business Impact Assessment of an application will include ratings for these different sources of impact. The security vulnerabilities in software can easily be related to this rating.

The ISO/IEC 27005 is an Information Security Management System that acknowledges the need to relate vulnerabilities to business process risk.
Since the goals of the business process are not the starting point for the framework, it cannot be used in the developed method to relate software vulnerabilities to business (process) risks.

The business risk model of SABSA links a vulnerability with high-level enterprise risk. It lacks the step from vulnerabilities via business process risk to higher-level business risks to be useful to explain the business process risk of a software vulnerability to a business manager.
Based on this, SABSA was not used in the developed method.

8.2 Assessment of objectives

The ultimate aim of the project was to establish a methodology that would bridge the gap between the technical vulnerabilities and risks understood by the business.

In this project, a methodology was developed that has four building blocks:

(i) Understand the environment
   a. Understand where the system belongs; what the greater context is.
   b. Understand what the system does; what the elements, functions are.
   c. Understand how the system acts; data and control flow of the system.

(ii) Assess the probability of the threat occurring
   A threat score model is developed that is used to evaluate whether there is a threat agent and to characterise his capability, motivation, catalyst, and the situational inhibitors and amplifiers..

(iii) Evaluate the impact of the vulnerability
   The impact will be evaluated for one or a combination of the context that influence the impact:
   a. Operational risk related to the business processes
   b. Regulatory risk caused by noncompliance with the applicable laws and regulations
   c. Reputational risk related to the brand of the organization
   d. IT generic risk stemming from the noncompliance of software with technical standards that leads to reduction of the overall information security.

(iv) Evaluate the risk
   The risk of a software vulnerability will be evaluated in a 5x5 matrix based on the assessed likelihood of the threat and the impact on the business when the vulnerability occurs.

The business impact of discovered software vulnerabilities can be assessed based on this model. The estimates that have led to the assessed business risk are made visible and can be the starting point of the discussion between the software security assessor and the business manager owning the application.

By successfully developing a model to assess the business risk of software vulnerabilities, the aim of this project has been accomplished. This project has demonstrated the applicability of the developed method based on a hypothetical organization and four hypothetical software security vulnerabilities.

9 Bibliography

[1] Iso/iec 27005:2011 information - security techniques - information security risk management, 2011.

[2] Iso/iec 27034: Information technology – security techniques – application security, part 1: Overview and concepts, 2011.

[3] Cobit 5: A business framework for the governance and management of enterprise it, 2012.

[4] Cobit5, enabling processes, 2012.

[5] Iso/iec 27034: Information technology – security techniques – application security, working draft: Part 2: Organization normative framework, 2013.

[6] Iso/iec 27034: Information technology – security techniques – application security, working draft: Part 3: Application security management process, 2013.

[7] E.G. Amoroso. *Fundamentals of Computer Security Technology*. Prentice-Hall PTR, Upper Saddle River, 1994.

[8] P. Bernard. *COBIT 5: A Management Guide*. Van Haren Publishing, Zaltbommel, 2012.

[9] A. Blyth and G.L. Kovacich. *Information Assurance: Security in the Information Environment*. Springer, London, second edition edition, 2006.

[10] R.N. Charette. *Applications Strategies for Risk Analysis*. McGraw-Hill Book Company, New York, 1990.

[11] D.I. Dickstein and R.H. Flast. *No Excuses: A Business Process Approach to Managing Operational Risk*. John Wiley & Sons, Inc., Hoboken, 2009.

[12] A. Fuchsberger. Lecture presentation: Building security in; software security. 2013.

[13] M. Howard and D. LeBlanc. *Writing Secure Code*. 2003.

[14] M. Howard and S. Lipner. *The Security Development Lifecycle: A process for Developing Demonstrably More Secure Software*. Microsoft Press, Redmond, 2006.

[15] ISACA. Cobit 5 for risk, 2013.

[16] ISF. Information risk analysis methodology (iram), 2010.

[17] A. Clark J. Sherwood and D. Lynas. *Enterprise Security Architecture: A Business-Driven Approach*. CMP Books, San Francisco, 2005.

[18] G. McGraw. *Software Security: Building Security In*. Pearson Education, Inc, Boston, 2006.

[19] Microsoft. Security bulletin severity rating system, 2012.

[20] P.G. Neumann. *Computer Related Risks*. The ACM Press, New York, 1995.

[21] NIST. Special publication 800-30 revision 1: Guide for conducting risk assessments, 2012.

[22] D.B. Parker. *Fighting Computer Crime: A New Framework for Protecting Information*. John Wiley & Sons, Inc,, New York, 1998.

[23] J. Sherwood. Sabsa, white paper, enterprise security architecture, 2009.

[24] F. Swiderski and W. Snyder. *Threat Modeling*. Microsoft Press, 2004.