SEVENTH FRAMEWORK PROGRAMME

RASEN

Compositional Risk
Assessment and Security
Testing of Networked Systems

Deliverable D5.1.1

Baseline Methodologies for Legal, Compositional, and Continuous Risk Assessment and Security Testing

Project title:	RASEN
Project number:	316853
Call identifier:	FP7-ICT-2011-8
Objective:	ICT-8-1.4 Trustworthy ICT
Funding scheme:	STREP – Small or medium scale focused research project

Work package:	WP 5
Deliverable number:	D5.1.1
Nature of deliverable:	Report
Dissemination level:	PU
Internal version number:	1.0
Contractual delivery date:	2013-01-31
Actual delivery date:	2013-01-31
Responsible partner:	UiO

Contributors

Editor(s)	Tobias Mahler (UiO)
Contributor(s)	Jürgen Groβmann (Fraunhofer), Tobias Mahler (UiO), Fredrik Seehusen (SINTEF), Bjørnar Solhaug (SINTEF)
Quality assuror(s)	Bruno Legeard (Smartesting), Albert Zenkoff (SAG)

Version history

Version	Date	Description
0.1	12-11-12	Table of contents with partner roles
0.2	11-12-12	Initial drafts of sections 2-6
0.3	20-12-12	Updated drafts of sections 2-6
0.4	10-01-13	Final texts for sections 1-3 & 5-6, draft section 7
0.5	11-01-13	Final texts for sections 1-7
0.6	16-01-13	Project-internal review
0.7	27-01-13	Final quality check
1.0	29-01-13	Final version

Abstract

This report assesses the state of the art of methodologies for, respectively, legal, compositional, and continuous risk assessment and security testing. The existing state of the art is examined from the perspective of the RASEN project, based on the research questions specified by the project. The report concludes with respective baseline methodologies, which will form the basis for the development work to be completed during the project.

Keywords

Methodology; risk management; information security; risk-based security testing; legal risk management; state of the art.

Executive Summary

WP5 of the RASEN project aims to develop methodology for compositional and continuous security risk assessment, methodology for compositional security testing, and methodology for legal risk assessment. The project will moreover develop support and guidelines for how to combine these methodologies into an overall framework where the different parts can leverage each other.

The Project RASEN addresses risk-based security testing for large software systems and software-based large scale networked systems. The central ideas of the RASEN project are the combination of security risk assessment with security testing, the application of compositional and continuous approaches to risk assessment and security testing and the integration of legal risk assessment and security testing. While the work packages 3 and 4 are dedicated to develop tools and technologies, the work package 5 has to ensure the overall integration of the results by means of methodologies and tool integration.

This report assesses the state of the art of methodologies for, respectively, compositional and continuous security risk assessment, security testing and legal risk assessment. The existing state of the art is examined from the perspective of the RASEN project, based on the research questions specified by the project. The report concludes with respective baseline methodologies, which will form the basis for the development work to be completed during the project.

A methodology in the sense of this deliverable describes a collection of systematically aligned activities with associated rules, methods and best practices to achieve a certain goal or result. This document collects the state of the art with respect to the methodology tasks in WP5. These tasks comprise the development of methodologies to cover the combination of risk assessment and security testing, continuous and compositional security assessment and security testing (focusing in particular on compositional assessment and reuse of assessment results), and legal risk assessment and security testing (whose main focus is to support compliance with legal norms). Moreover we opt for an integration of the methodologies so that we can make use of the overall synergies.

So far, we are not aware of any approach or any methodology that covers these areas in a sufficient way. However, we are sure that we can build upon existing methodologies by adopting their principles and providing extensions that especially serve our ideas. As a basis for selecting appropriate approaches we refer to the research areas and research questions defined in the proposal. This report addresses these questions within the following structure:

Section 2 gives an overview of the state of the art on security risk assessment methodologies. As explained, none of the existing methodologies provide appropriate solutions to the research questions raised by the project. However, the project will carefully build on international standards and best practices when targeting the R&D objectives. In particular, the methodologies that we develop in WP5 should as much as possible be compliant with established standards such as ISO 31000 and ISO/IEC 27005.

Section 3 provides an overview on methodologies and best practices for testing and security testing. While the outlined testing methodologies provide a good guidance on how testing activities can be generally aligned with software development processes, the presented security testing methodologies involve additional activities that are necessary for security testing and model-based security testing.

Section 4 discusses the state of the art of methodologies that combine risk assessment and testing. RASEN distinguishes between risk-based testing (risk assessment to improve the testing) and test-based risk assessment (using testing to improve the risk assessment). Section 4 presents one generic process for each alternative. Furthermore the relevant approaches from the literature are classified according to which steps of the processes they considered. Finally, an approach is identified that will be used as a starting point for the research within RASEN.

As discussed in Section 5, the RASEN project envisages a methodology for continuous security risk assessment and testing that is much more iterative than traditional and established approaches. Continuous assessment should be enabled by rapid reassessments in order to maintain the validity of the risk assessment results while the target system or its environment changes and evolves. Security testing is considered to provide the necessary experimental feedback to improve the risk analysis and its assumptions. Considering the existing methodological support described in Section 5, there are some means for risk monitoring, security testing and traceability that can provide leverage and facilitate continuous assessment. However, the state of the art is very sparse regarding the RASEN

research questions on composition and decomposition of risk assessment artifacts, and on reuse of risk assessment and testing artifacts.

Moreover there is currently no approach that integrates technical risk assessment and legal risk assessment, as discussed in Section 6. While there are existing approaches for aligning legal risk management with other risk management approaches based on ISO standard 31000, these do not put sufficient emphasis on compliance management. The latter has a crucial role for the integration with technical risk management, which needs to be carried out in compliance with legal requirements.

The RASEN baseline presented in Section 7 explains how the project will adopt existing methodologies if possible and useful. This applies especially to the area of testing, traditional security testing and risk-based security testing. In all other cases RASEN aims to make strong use of its own visions and will develop methodologies that integrate compositional security risk assessment and model-based security testing, that integrate technical risk assessment and legal risk assessment, and that support an efficient reuse and reassessment of existing artifacts.

Table of contents

TABLE OF CONTENTS 6

1 INTRODUCTION 7

2 SECURITY RISK ASSESSMENT METHODOLOGIES 8

2.1 RISK MANAGEMENT 8
2.2 SECURITY RISK MANAGEMENT 9
2.3 SECURITY RISK ASSESSMENT METHODS 9
2.4 CHALLENGES AND OUTLOOK 11

3 SECURITY TESTING METHODOLOGIES 12

3.1 OVERVIEW ON SECURITY TESTING 12
3.2 SECURITY TESTING GUIDELINES AND METHODOLOGIES 13
3.2.1 Software Testing in the Software Development Life Cycle 13
3.2.2 Classical Security Testing Methodologies and Guidelines 13
3.2.2.1 The NIST Technical Guide to Information Security Testing and Assessment 14
3.2.2.2 OSSTMM 15
3.2.2.3 OWASP Testing Guide 16
3.2.3 Model-based Security Testing Methodologies 17
3.2.4 Summary and Outlook 19

4 RISK-BASED TESTING AND TEST-BASED RISK ASSESSMENT 20

4.1 OVERVIEW OF METHODOLOGIES FOR RISK-BASED TESTING AND TEST-BASED RISK ASSESSMENT 20
4.2 A CLASSIFICATION OF APPROACHES TO RISK-BASED TESTING AND TEST-BASED RISK ASSESSMENT FROM A METHODOLOGY PERSPECTIVE 21

5 CONTINUOUS SECURITY RISK ASSESSMENT METHODOLOGIES 24

6 LEGAL RISK ASSESSMENT METHODOLOGIES 27

6.1 LEGAL RISK MANAGEMENT 27
6.2 INFORMAL APPROACHES TO LEGAL RISK MANAGEMENT 27
6.2.1 Preventive Law 27
6.2.2 Best Practices: Managing Legal Risk Effectively 28
6.2.3 Australian Standards Handbook on Legal Risk Management 28
6.2.4 McCormick 29
6.2.5 Wahlgren 29
6.3 METHODOLOGIES 29
6.3.1 Reid 29
6.3.2 Extending ISO 31000 29
6.4 COMPLIANCE MANAGEMENT 30

7 RASEN BASELINE 32

7.1 INTRODUCTION 32
7.2 SECURITY RISK ASSESSMENT METHODOLOGIES 32
7.3 SECURITY TESTING METHODOLOGIES 32
7.4 RISK-BASED TESTING AND TEST-BASED RISK ASSESSMENT METHODOLOGIES 33
7.5 CONTINUOUS SECURITY RISK ASSESSMENT 34
7.6 LEGAL RISK ASSESSMENT 34

8 CONCLUSION 36

REFERENCES 37

1 Introduction

The Project RASEN addresses risk-based security testing for large software systems and software-based large scale networked systems. The central ideas of the RASEN project are the combination of security risk assessment with security testing, the application of compositional and continuous approaches to risk assessment and security testing and the integration of legal risk assessment and security testing. While the work packages 3 and 4 are dedicated to develop tools and technologies, the work package 5 has to ensure the overall integration of the results by means of methodologies and tool integration. A methodology in the sense of this deliverable describes a collection of systematically aligned activities with associated rules, methods and best practices to achieve a certain goal or result. This document collects the state of the art with respect to the methodology tasks in WP5. These tasks comprise the development of methodologies to cover the combination of risk assessment and security testing, continuous and compositional security assessment and security testing (focusing in particular on compositional assessment and reuse of assessment results), and legal risk assessment and security testing (whose main focus is to support compliance with legal norms). Moreover we opt for an integration of the methodologies so that we can make use of the overall synergies.

So far, we are not aware of any approach or any methodology that covers these areas in a sufficient way. However, we are sure that we can build upon existing methodologies by adopting their principles and providing extensions that especially serve our ideas. As a basis for selecting appropriate approaches we refer to the research areas and research questions defined in the proposal. These question primarily address

- the composition and decomposition of risk assessment and testing artifacts (RQ1, RQ2),
- the reuse of risk assessment and testing artifacts (RQ3),
- the interplay between risk assessment and testing (RQ5), and
- the role of legal risk assessment for security risk assessment and testing (RQ8, RQ9).

This document outlines the state of the art by addressing different kinds of methodologies, which are considered to be relevant for the methodologies to be developed in RASEN. Section 2 describes methodologies that address security risk assessment in general, Section 3 describes security testing methodologies and model-based security testing methodologies, Section 4 describes methodologies for combining risk-based testing and test-based risk assessment, Section 5 addresses continuous security risk assessment and Section 6 legal risk assessment methodologies. Section 7 provides the baseline for the project. The baseline section provides an initial assessment of the state of the art and specifies the starting point for the further development in the RASEN project. The conclusions are then presented in Section 8.

2 Security Risk Assessment Methodologies

In this section we give an overview of established standards, guidelines and methodologies on risk management and security risk assessment. Because security risk assessment can be understood as a specialization of risk assessment, we give an introduction to the former before introducing the security considerations.

2.1 Risk Management

As defined by the ISO 31000 standard [33] [34], risk assessment is a part of the overall risk management, which is "coordinated activities to direct and control an organization with regard to risk". A risk is the combination of the consequence and likelihood of an event (unwanted incident). The consequence is in terms of degree of harm to an asset, and can be measured qualitatively or quantitatively. An asset is something of value to a stakeholder and therefore requires protection from risk. Likelihood is the chance for something to happen, and can be given in terms of frequency or probability, and measured qualitatively or quantitatively. A risk source is something that has the potential to give rise to risk by exploiting vulnerabilities.

The activities of the overall risk management process are depicted in Figure 1. The five activities in the middle are commonly conducted from time to time on a regular basis to identify risks and strategies for risk mitigation. The two remaining activities should be conducted continuously.

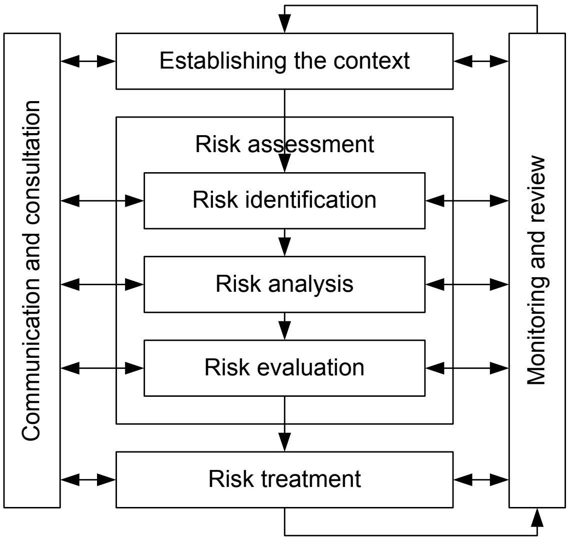

Figure 1 - Risk management process (ISO 31 000 / 2009 [33])

Communication and consultation are continual and iterative processes that organizations conduct in order to provide, share and obtain risk relevant information, and to interact with stakeholders regarding the risk management. Monitoring is the continual checking and supervising to determine the current status and identify possible changes to the risk picture; review is the activity to determine the suitability, adequacy and effectiveness of the current risk management strategies.

The five remaining activities include the risk assessment, and most established risk assessment methods follow this process. Establishing the context is to articulate the objectives of the risk management, set the scope of the risk assessment, and define the risk criteria. A risk assessment is always conducted with respect to a specific target of analysis, and the context establishment includes creating a target description to understand the scope and focus of the risk assessment, including the assets and stakeholders. The risk criteria are a specification of the risk tolerance with respect to the identified assets, and the identified risks will be evaluated against these in order to determine whether or not they are acceptable. The results of the context establishment serve as a basis for the subsequent risk assessment.

The risk assessment involves the three activities of risk identification, risk analysis and risk evaluation. Risk identification is the process of finding, recognizing and describing risk. This includes the identification of risk sources, vulnerabilities, as well as events and their causes. Techniques for risk identification are described in RASEN deliverable D3.1.1, and may involve the use of historical data, theoretical analysis, expert judgments, and so forth. Risk analysis is to comprehend the nature of risk and to determine the level of risk. The analysis involves estimating the risks by estimating the likelihood and consequence of the identified events (unwanted incidents) that constitute the risks. The resulting risk estimates are used as a basis for the risk evaluation and for the decisions about risk treatment. Risk evaluation is to compare the results of the risk analysis with the predefined risk criteria to determine whether the risks and their magnitude are acceptable or tolerable. Finally, following the risk assessment is the risk treatment. This activity is to identify options for risk mitigation in order to modify the unacceptable risks, for example by identifying means for reducing the likelihood or consequence of risks.

2.2 Security Risk Management

Security risk management can be understood as a specialization of the more general risk management process defined by ISO 31000, and there exist several methods that support security risk assessment. Within the ICT domain, the ISO/IEC 27000 series constitute a number of established standards targeting information security matters. In particular, the ISO/IEC 27005 [35] standard provides guidelines for information security risk management that are compliant with the ISO risk management guide [34].

The ISO/IEC 27005 standard is designed to support the implementation of information security as specified by ISO/IEC 27001 [31]. According the latter, information security should be ensured by establishing, implementing, operating, reviewing, maintaining and improving an Information Security Management System (ISMS). The ISMS is designed to ensure the selection of adequate security controls that protect information assets, which are knowledge or data that has value to the organization [30]. Assets to be considered may also include software, hardware, services and people, as well as intangibles such as reputation and image.

While following the risk management process of the ISO 31000 standard, the security risk assessment is focusing on the identification of information security events and their causes and consequences, and on the identification of information security risks. An information security event may result in a breach of information security. Information security, in turn, is defined as the preservation of confidentiality, integrity and availability of information. Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes; integrity is the property of protecting the accuracy and completeness of information; availability is the property of being accessible and usable upon demand by an authorized entity.

Other security properties that are often considered are authentication, non-repudiation and authorization [30][25]. The preservation of these properties may contribute to the preservation of confidentiality, integrity and confidentiality. Authentication is the provision of assurance that a claimed characteristic of an entity is correct, for example to confirm the identity of an actor. Non-repudiation is the ability to prove the occurrence of a claimed event or action and its originating entities, in order to resolve disputes about the occurrence or non-occurrence of the event or action and involvement of entities in the event. Authorization is to explicitly allow or deny users access to information or resources.

2.3 Security Risk Assessment Methods

In the following, we will review relevant security risk assessment methods.

The purpose of the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) method [2] is to enable organizations to understand and address their information security risks. Similar to ISO 31000, OCTAVE is a risk assessment process that is conducted on a regular basis as part of the overall and continuous security risk management of an organization. The security risk evaluation of OCTAVE considers both organizational and technological issues. In other words, it addresses both the ICT infrastructure and how the infrastructure is used by the organization in conducting its business processes.

The OCTAVE process is asset-driven, and is targeting the three main components that must be in place for risks to arise, namely asset, threat and vulnerability. Loosely speaking, the process follows the same steps as the ISO process depicted in Figure 1, although it is conducted over three phases. Phase 1, Build Asset-Based Threat Profiles, corresponds to the context establishment and includes identification of information-related assets. For the most important assets, the relevant security requirements are described, and threats are identified to understand the threat profile with respect to the assets. Phase 2, Identify Infrastructure Vulnerabilities, is to relate each asset to key information technology systems and components, and to identify vulnerabilities of the infrastructure. Phase 3, Develop Security Strategy and Plans, is to identify risks and risk mitigation strategies.

CCTA Risk Analysis and Management Method (CRAMM) [74] is compliant with ISO 27000. The overall process is divided into a risk analysis part and a risk management part. Risk analysis is the identification of risks by the identification of assets, threats and vulnerabilities. Risk management includes the identification and implementation of countermeasures, as well as auditing the status of such security controls.

CORAS [48] is a model-driven approach to risk assessment that consists of three parts, namely a language, a tool and a method. The method is asset-driven, and is defined by a process closely corresponding to the risk management process of ISO 31000. Step 1 through Step 4 are dedicated to the context establishment, and include the identification and modeling of assets and their dependencies. Step 5 is the risk identification, and includes the identification and modeling of threats, threat scenarios, vulnerabilities and unwanted incidents using CORAS threat diagrams. Risks are identified by systematically going through the target description that is created during the context establishment. Step 6 is the risk estimation, and includes the estimation of likelihoods and consequences of unwanted incidents. The risk estimation is conducted by using the threat diagrams as basis, and it is supported by techniques and rules for formal reasoning about likelihoods in the diagrams. Step 7 is the risk evaluation, in which the identified risks and their risk levels are compared with the risk evaluation criteria defined during the context establishment. The evaluation is the basis for deciding which risks should be considered for treatment. Finally, Step 8 is the risk treatment, which is to identify cost-efficient treatments for unacceptable risks. The results are documented by means of CORAS treatment diagrams.

Peltier [59] defines a risk management process of six steps that is much similar to the ISO 31000 standard. To support management of information security related risks, the Facilitated Risk Analysis and Assessment Process (FRAAP) is introduced, where the objective is to protect the confidentiality, integrity and availability of information. The FRAAP process involves analyzing one system, application, platform, business process, or segment of business operation at a time. The security risk identification and assessment is conducted by a team of experts, and the overall FRAAP process is divided into three phases. The pre-FRAAP phase is for deciding the target and the scope of the analysis, creating the system description, and putting together the FRAAP team. The second phase is the FRAAP session, which is divided into two stages. Stage 1 includes threat and risk identification, probability and impact estimation, as well as identification of controls to reduce information security risks to an acceptable level. Stage 2 includes identification of existing controls, as well as allocation of responsibilities for implementing selected controls. Finally, the post-FRAAP phase is conducted to generate the FRAAP reports and decide a risk assessment action plan for how to follow-up the FRAAP results.

The EBIOS (Expression of Needs and Identification of Security Objectives) [1] method is used to assess and treat risks related to information systems security, and is developed to support compliance with ISO 31000, ISO/IEC 27001 and ISO/IEC 27005. EBIOS is conducted using a process of five steps, namely circumstantial study, security requirements, risk study, identification of security goals, and determination of security requirements.

Microsoft has developed the Security Risk Management Guide (SRMG) [53] to assist organizations in security risk management. The guide explains how to conduct the phases of a risk management process in order to measure security risks and keep them at an acceptable level. The process consists of four phases, namely risk assessment, conducting decision support, implementing controls, and measuring program effectiveness. The risk assessment includes risk identification, estimation and prioritization, and serves as a basis for decision making during the following phase. Conducting decision support is to determine how to cost efficiently address the most important risks by identifying controls, estimating their costs, and estimating their effect in terms of risk reduction. Implementing controls is to create and execute plans for risk mitigation based on the identified controls. Finally, measuring program effectiveness is to follow-up by estimating the progress with regard to the overall risk management process.

The U.S. National Institute of Standards and Technology (NIST) has developed the Risk Management Guide for Information Technology Systems [83]. The guidelines define a risk assessment process that consists of nine steps. Step 1 is the system characterization; Step 2 is the threat identification; Step 3 is the vulnerability identification; Step 4 is the control analysis; Step 5 is the likelihood determination; Step 6 is the impact analysis; Step 7 is the risk determination; Step 8 is the control recommendation; Step 9 is the results documentation. The guide moreover provides support for how to do the risk mitigation, including identification of risk mitigation options and a cost-benefit analysis. Generally speaking, the NIST process follows the process defined by ISO 31000, although it particularly targets security in ICT systems.

RASEN deliverable D3.1 gives an overview of the ARIS GRC tool-set. Governance, risk management and compliance (GRC) typically covers an organization's activities regarding corporate governance, enterprise risk management, and compliance with governmental laws and regulations [27]. The ARIS GRC tool-set is an enterprise level framework [67] that supports an operational risk management [42] process of six phases. The phases include activities of risk identification, risk evaluation and assessment and risk mitigation, as well as risk monitoring and reporting. A core feature of the ARIS risk management approach is that all information is retained in a central repository to allow a holistic and transparent approach with a unified view on risk, risk controls, enterprise models and business processes.

2.4 Challenges and Outlook

Considering the research questions of RASEN, none of the established methods for risk management and security risk assessment presented in this section provide specific support for the particular methodological challenges addressed by the project. These challenges include methods for compositional and continuous risk assessment, as well as assessment of security risk related to legal norms. The RASEN project will nevertheless build closely on established standards such as ISO 31000 and ISO/IEC 27005. On the one hand, this is to ensure that the project develops specialized methods that are still compliant with well-known and established methods and processes. On the other hand, any risk management process should cover all of the standard risk management tasks as depicted in Figure 1. In the RASEN project, the challenge will be how to specialize this process to support compositional risk assessment and analysis of legal issues.

While building on standard methods for security risk management and assessment, RASEN will also leverage existing methods on risk-based testing, test-based risk assessment, continuous risk assessment, as well as legal risk assessment. Such methods are presented in the following sections.

The baseline for the security risk assessment aspects of the RASEN methodology will be presented in Section 7.2.

3 Security Testing Methodologies

A Security Testing Methodology describes the phases, tasks and methods that are necessary to successfully conduct security testing. While the RASEN deliverable D4.1.1 [61] provides an overview over security testing techniques, this section outlines the current state of the art for security testing methodologies and guidelines. We emphasize on methodologies for model based security testing (MBST). However, since there are currently no exhaustive standards, methodologies or comprehensive guidelines/best practices for MBST, we start with an overview of security testing and related methodologies. Section 3.1 provides a summary of security testing and the established security testing techniques. Section 3.2 describes security testing methodologies and methodologies for model-based security testing and Section 3.3 provides an outlook and defines the research challenges that need to be addressed when defining a security testing methodology especially in the field of MBST.

3.1 Overview on Security Testing

The term software security testing characterizes activities to experimentally check the security features and resistance to attacks of software implementations. While a number of approaches have long been around to target specific attacks on systems (e.g. vulnerability scanners) more systematic security testing approaches with respect to specified policies or security properties like confidentiality, integrity or availability are a relatively new concern that has started to be addressed in the last few years. A first research workshop on this topic was actually held on the International Conference on Software Testing (ICST) 2008 in Lillehammer [85]. Meanwhile, this workshop is a permanent part of the ICST and is organized by the research projects and SpaCIoS [72] and DIAMONDS [18].

In general the software security testing activities can be divided into functional security testing and security vulnerability testing [92]. Security functional testing ensures whether software security functions are implemented correctly and consistent with the security requirements in the security requirements specification. It is used to check the functionality, efficiency and availability of the carefully planned security functionalities and systems (e.g. firewalls, authentication and authorization subsystems, access control). Security vulnerability or penetration testing directly addresses the identification and discovery of actually undiscovered system vulnerabilities. Security vulnerability and penetration testing in general analyze systems for potential vulnerabilities that may result from a poor or improper system configuration, from known and/or unknown hardware or software flaws, or from operational weaknesses in process-related or technical countermeasures.

In the same way security breaches exploit weaknesses from different sources with different technologies and techniques, security testing methodologies and guidelines have to consider this kind of diversity and adopt and integrate testing techniques from different areas. For example, Takanen et al. [84] claim that more than 70% of modern security vulnerabilities are programming flaws, with only less than 10% being configuration issues, and about 20% being design issues. Programming errors and related vulnerabilities (e.g. buffer overflows, SQL injection, etc.) currently represent one the biggest security problem and can often be detected before a system is deployed. In contrast configuration errors are most often only detectable after a system has been deployed. Besides this, especially denial of service (DOS) attacks and distributed denial of service (DDOS) attacks are considered as a severe security problem with respect to the availability of systems and their services. Thus, testing non-functional characteristics such as scalability and robustness become a critical part of a security testing strategy. Moreover, the compilation of techniques may vary when the subject of test changes. Application Security Testing will have a stronger relation to testing techniques that can be used for the detection of programming errors, while network security testing requires approaches that simulate network related attacks (e.g. DOS attacks) and are dedicated to reveal vulnerabilities in dedicated security components like firewalls and security protocols. Active security testing techniques (e.g. penetration testing and fuzzing) are often used in the system development or maintenance phases. It may intervene in the systems operation and can ultimately disrupt critical operations. Passive techniques (e.g. network scanner, monitoring) are often used in the operational phase to detect configuration related vulnerabilities, anomalies in network traffic, and continuously monitor the health of the system.

Unfortunately security testing, especially security vulnerability or penetration testing, lacks systematic approaches, which enable the efficient and goal oriented identification, selection and execution of test cases and a proper integration in the software development process. In addition, dedicated strategies for the application of security tests during the operation of the software, especially for large network-based infrastructures, are still missing.

3.2 Security Testing Guidelines and Methodologies

This section provides an overview over existing security test methodologies and guidelines. Currently we know three documents that provide explicit guidance for a security testing. These guidelines are dedicated to security testing practitioners and represent the current knowledge when security testing is applied in industry and business. Moreover there are a number of academic papers that describe methodologies that are dedicated to certain security aspects or to special kinds of systems. However, the basis for security testing is its integration in the overall testing process and its relation to the system development and deployment. Section 3.2.1 provides an overview on testing methodologies in the software development life cycle. Section 3.2.2 describes the currently existing security testing guidelines for classical security testing and Section 3.2.3 provides an overview on security testing methodologies in the area of MBST.

3.2.1 Software Testing in the Software Development Life Cycle

A classical software testing process consists of phases for test planning, test Software Testing in general can be implemented at any time in the development process, however the main part of the testing activities occur after the requirements have been defined and the coding process has been completed. Testing activities (e.g. test planning test, specification, test implementation, test execution, test evaluation) have to be well planned and seamlessly aligned with the system development activities.

Existing process model like the V-Model [87] and the Rational Unified Process [86] do cover already the main test activities and their alignment with the overall software of system development process. An enhancement of the traditional V-Model, the so-called W-Model [78], explicitly models the relationship between development activities and testing activities. The W-model is a process model for software development processes. It is based on the far spread V-model. Besides the main development activities (requirement definition, functional and technical system model, and component specification) the W-model especially focuses test activities. The test activities start early in the development process and are directly tied to the development activities. Developers with specialized knowledge in the field of quality assurance and particularly testing, are directly involved in the individual development activities. Thus, they are able to influence the system specification with respect to testability and maintainability and can align the test activities with their related system development activities. The Standard for Software and System Test Documentation (IEEE 829-2008) [28] is an IEEE standard that specifies a set of artifacts that are used in eight defined phases of software testing process. Each of the phases is considered to produce its own separate type of artifacts, i.e. the Test Plan, the Test Design Specification, the Test Case Specification, the Test Procedure Specification, the Test Item Report, the Test Log, the Test Incident Report, and the Test Summary Report. The Test Management Approach (TMAP) [75] additionally provides a structured approach to the processes and activities of testing. It is developed on basis of practical testing experiences and defines the following corner stones that are considered to be essential for testing. A business-driven test management allows the control of costs, risks and results, a process structure defines the well structured phases and activities for testing and a testing toolbox provides a set of dedicated testing techniques, infrastructures and organizations to properly conduct the testing. A fourth essential claims the necessity of being flexible and adaptive. TMAP is widely accepted in the industry.

However, while normal testing methodologies emphasizes testing in the software development phase, security testing methodologies must additionally emphasize on testing activities in the deployment and maintenance phases.

3.2.2 Classical Security Testing Methodologies and Guidelines

This section summarizes dedicated security testing methodologies and guidelines. We are currently aware of three methodologies or guidelines that are published for the purpose of supporting security testing of actual products and systems.

- The Technical Guide to Information Security Testing and Assessment from the National Institute of Standards and Technology (NIST) [70] has been published in 2008 and provides a collection of recommendations for technical testing and examination methods and techniques and a methodology on how to apply these methods and techniques.
- The Open Source Security Testing Methodology Manual (OSSTMM) published by the Institute for Security and open Methodologies (ISECOM) [26] is currently available in the version 3.02 and belongs to a series of evolving security testing standards that have been published since 2001.
- The OWASP Testing Guide published by the Open Web Application Security Project (OWASP) Foundation in 2008 [58] and describes best practices for security testing for web applications.

3.2.2.1 The NIST Technical Guide to Information Security Testing and Assessment

The Technical Guide to Information Security Testing and Assessment from the National Institute of Standards and Technology (NIST) [70] provides recommendations and best practices for security testing and a security testing methodology. The NIST guide distinguishes three typical phases of a security assessment or testing process. The phases are (Test-) Planning, (Test-) Execution and Post-(Test-) Execution.

The **Planning** phase is used to gather all relevant information that is used during a security assessment. The main tasks of this phase are developing a security assessment policy, prioritizing and scheduling assessments, selecting and customizing technical testing and examination techniques, developing the assessment plan and addressing any legal considerations with respect to the security assessment. According to the NIST guide “a security assessment should be treated as any other project, with a project management plan to address goals and objectives, scope, requirements, team roles and responsibilities, limitations, success factors, assumptions, resources, timeline, and deliverables” [70].

The **Execution** phase is primary dedicated to identify and validate vulnerabilities in the target system For this purpose the NIST guide relates to different techniques that can help to achieve this goal. The NIST guide explicitly distinguishes between

- the analysis of the target (Target Identification and Analysis Techniques) by means of network discovery techniques, network port and service identification, vulnerability scanning techniques etc., and
- the validation of vulnerabilities (Target Vulnerability Validation Techniques) e.g. by means of password cracking, penetration testing, and social engineering.

In the document, the techniques are roughly presented characterized and evaluated.

The **Post-Execution** phase treats the findings from the previous phase. The NIST guide proposes a set of possible actions that range from simple reporting that is used to inform about the findings to developing mitigation actions and recommendations that help to mitigate the findings or prevent their exploitation. The activities in the post-execution phase are not exclusively part of the testing process but are also part of a risk assessment process that make use of the testing results.

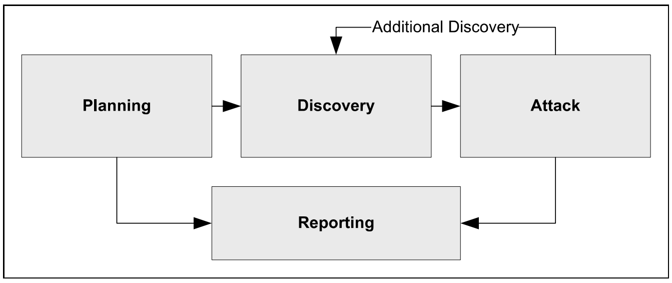

Figure 2 – Four-stage penetration testing methodology [70]

All testing activities are described as iterative processes that use the results of subsequent stages to initiate a new iteration with updated results (see Figure 2).

3.2.2.2 OSSTMM

The Open Source Security Testing Methodology Manual (OSSTMM) [26] is a methodology that describes how to obtain operational security of physical locations, human interactions, and all forms of communications such as wireless, wired, analog, and digital by means of security testing. Security testing in the context of the OSSTMM means dynamic security testing as well as static security testing.

The OSSTMM has been first published in 2001 and since then is updated regularly. The OSSTMM is published by ISECOM, an organization that additionally provide a certification scheme, the ISECOM Licensed Auditor (ILA) program, for testers to state security testing, penetration testing, or security analysis expertise with respect to the OSSTMM.

The OSSTMM defines so-called classes (like Physical Security Class, Spectrum Security Class, Communication Security Class) and channels (like Human, Physical, Wireless, Telecommunications, and Data Networks) that define classes and subclasses of testing areas. For each of the classes the OSSTMM defines testing approaches that are all mapped on a common security testing workflow. The workflow consists of 10 different activities that can be carried out depending on the requirements of the classes and channels.

1. Collect data of normal operations to comprehend the target.
2. Actively test operations by agitating operations beyond the normal baseline.
3. Analyze data received directly from the operations tested.
4. Analyze indirect data from resources and operators (i.e. workers, programs).
5. Correlate and reconcile intelligence from direct (step 3) and indirect (step 4) data test results to determine operational security processes.
6. Determine and reconcile errors.
7. Derive metrics from both normal and agitated operations.
8. Correlate and reconcile intelligence between normal and agitated (steps 1 and 2) operations to determine the optimal level of protection and control that would best be implemented.
9. Map the optimal state of operations (step 8) to processes (step 5).
10. Create a gap analysis to determine what enhancements are needed for processes governing necessary protection and controls (step 5) to achieve the optimal operational state (step 8) from the current one.

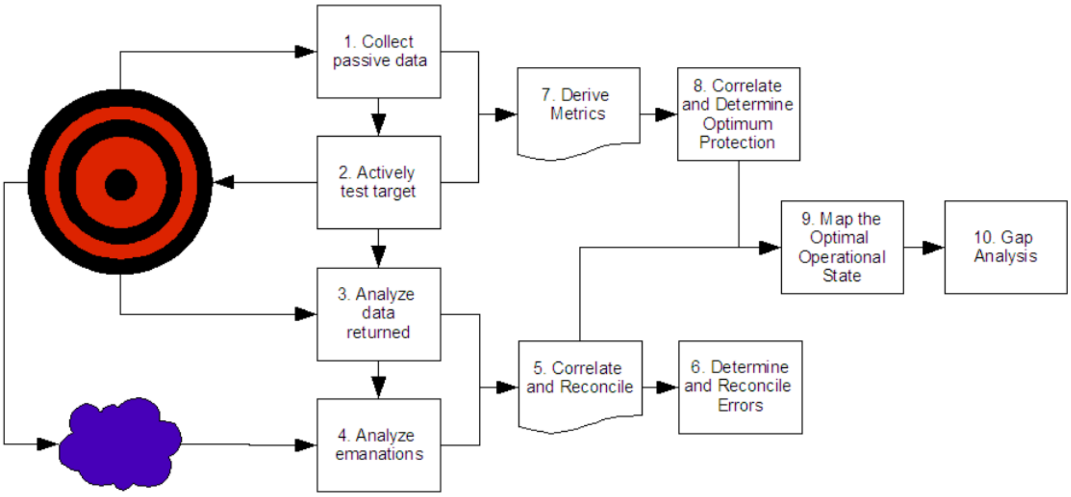
1. Collect passive data
2. Actively test target
3. Analyze data returned
4. Analyze emanations
5. Correlate and Reconcile
6. Determine and Reconcile Errors
7. Derive Metrics
8. Correlate and Determine Optimum Protection
9. Map the Optimal Operational State
10. Gap Analysis

Figure 3 – The OSSTMM workflow [26]

3.2.2.3 OWASP Testing Guide

The OWASP security testing methodology [58] is developed by a community of security experts and evolves regularly to include new types of attack. The methodology is explicitly dedicated to testing the security of web applications that are based on web services and provides a set of "best practices" for web application penetration testing. This methodology works equally well for black box, grey box and white box testing and contains 350 pages divided in 10 categories that provide tests for over 60 security controls. The OWASP testing guide distinguishes phases or tasks that are directly motivated by the different technologies that are used in a web application and by different kind of attacks that are known for web applications. The testing itself is framed by phases of information gathering in the beginning and test result analysis at the end.

1. **Information Gathering** is the first phase and focused on collecting as much information as possible about the target application to be able to concisely plan the actual test phases. To that end, data from the system itself and from the security risk analysis are required.
2. The **testing phases** are aligned with the typical structure of a web application**:**

   2.1. **Configuration Management Testing** analyzes the infrastructure and topology of a web application. Information such as code source, HTTP methods permitted, administrative functionality, authentication methods and infrastructural configurations can be obtained.

   2.2. **Testing dedicated Security Mechanism** concentrate on testing authentication, session management, and authorization. While Authentication is the act of establishing or confirming something (or someone) as authentic, e.g. in the logon process, Session Management broadly covers all controls of a user from authentication to leaving the application and thus provides a central security function. Authorization is the concept of allowing access to resources only to those permitted to use them. Testing for Authorization means understanding how the authorization process works, and using that information to circumvent the authorization mechanism.

   2.3. **Business Logic Testing and Data Validation Testing** focuses on the input validation and the business logic. The OWASP guide considers the failures during input as one of the most common web application security weakness. This kind of weakness leads to almost all of the major vulnerabilities in web applications, such as cross site scripting, SQL injection, interpreter injection, locale/Unicode attacks, file system attacks, and buffer overflows. Moreover vulnerability in the business logic cannot be detected by a vulnerability scanner and relies upon the skills and creativity of the penetration tester. In this sense, this type of vulnerability is usually one of the hardest to detect and at the same time, one of the most dangerous to the application, if exploited.

2.4. **Denial of Service Testing** provides tests that simulate DoS and DDoS attacks. The fundamental concept of a network DoS attack is a malicious user flooding enough traffic to a target machine that it renders the target incapable of keeping up with the volume of requests it is receiving.

2.5. **Infrastructure and Technology Testing** i.e. Web Services Testing, SOA Testing, and Ajax Testing are applied with the goal to fin vulnerabilities in supplemental technologies like AJAX and in the Web Service infrastructure.

3. **Collecting Results**: The final phase is used to break down the security findings and evaluate them with respect to their security risks. Thus, the phase is strongly connected to risk management. Risk management results are used to prioritize the findings and the actions that are necessary to mitigate the findings and thus possibly associated risks. For this purpose the OWASP testing guide refers to the OWASP Risk Rating Methodology, a basic framework for risk rating and management that allow a customizable risk rating for an application or an organization.

3.2.3 Model-based Security Testing Methodologies

Model-based Testing (MBT) constitutes a number of technologies, methods, and approaches with the aim, to improve the quality, the efficiency, and the effectiveness of test processes, tasks and artifacts. Model-based Security Testing (MBST) is a special form of MBT that is focused on the testing of security properties of a system. A model in the sense of MBT or MBST is usually an abstract, partial representation of the system under test specifying the desired behavior, the architecture, or the interaction with the environment. The test cases derived from this model are functional tests on the same level of abstraction as the model. For a SUT, various system models might exist, such as Requirements models, behavioral models, risk models and usage models. In contrast to MBT, MBST emphasizes on test behavior that additionally requires addressing interactions that simulate attack scenarios or that provokes unwanted system behavior (i.e. bringing a system in unstable or insecure states). Model-based security testing involves in accordance to model based testing the following major activities: defining the system and security requirements, building the model, defining test selection criteria and transforming them into operational test case specifications, generating tests, conceiving and setting up adaptor components and the test environment and then executing the tests on the SUT. Figure 4 shows a generic model-based testing process as defined in [21]

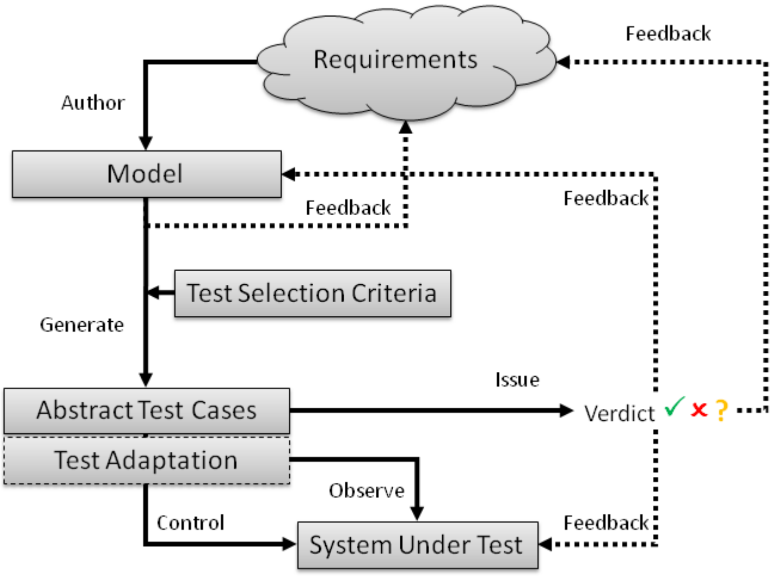

Figure 4 – Artifacts and activities for model-based testing (ETSI [21])

Test selection criteria are used to control the generation of test cases by being formalized into test case specifications. The chosen selection criterion assigns priorities to the test cases that define the order in which they will be executed. The criterion also indirectly defines properties of the generated test suites, like their fault detection power, cardinality, complexity, etc.

For MBST the overall workflow is quite similar. However, going into details there are slight differences especially when we have closer look at the artifacts that are addressed in the activities.

- The requirements, that are referred to in an MBST methodology are security requirements that define either functional security controls or non-functional security properties.
- The models in an MBST methodology are either models that explicitly specify security controls (e.g. RBAC models, models for authentication protocols) or models that describe vulnerabilities or vulnerable systems. Models of vulnerable systems are often obtained by applying techniques that slightly modify or mutate the functional model of a system.
- Some approaches use additional security test purposes that complement the functional models and are directly derived from either security requirements or security test objectives. These test purposes relate to the functional model and specify a certain unwanted state or situation. Test purposes and the functional model form the input of a test generation process that generates tests, which try to force the system in the state that has been specified by the test purposes [51].
- Test selection is either based on different kinds of model coverage or by means of risk ratings that are taken from a risk assessment. The latter is referred to as Risk-based Security Testing (RBST).

Felderer et Al. [22] provide a classification scheme for MBST approaches that distinguishes the approaches by the kind of artifacts that are involved. The classification provides a two dimensional view. One dimension considers whether risk ratings are used for the test selection or not. The other dimension defines the grade of automated test generation. Regarding the latter the authors distinguish between no generation, semi-automated generation, and fully automated generation. Respecting all combination the authors define 6 classes, 3 classes that have a distinct relation to security risk assessment and 3 classes that have no distinct relation to security risk assessment.

The challenges in creating a common MBST methodology are the diversity of approaches. In the following we introduce research areas that we consider to be relevant and that pose requirements to a security testing methodology that is to be developed in RASEN.

- Mutation-based Model-based security Testing (MBMBST) approaches generate tests that simulate malicious behavior. For this purpose the test generation process is not applied to valid system specifications but to system specification that had been slightly modified or mutated, so that they can be used as a kind of threat or vulnerability model. Relevant approaches have been presented by Jürjens [38] and in the DIAMONDS project [18]. Moreover, in the SPaCIoS [72] project a methodology for a security testing based on model checking techniques has been proposed. The approach consists of multiple subsequent steps that start with the definition of a formal model, the test generation on basis of a model checker, the refinement of the generated attack vectors and the test execution. The models used for test generation are security-enhanced models that have been modified by the application of mutation operators.
- Model-based security testing from behavioral models and test purposes is an extension of functional MBT to serve security testing. The technique and methodology has been developed by Smartesting in the POSE project [51] and has been extended in the DIAMONDS [18] project. In addition to functional MBT the model for test generation captures the expected behavior of the system under test (SUT). It formalizes the security functions of the SUT but also the possible stimuli of an attacker as well as the expected answer of the SUT. The test purposes are test selection criteria that formalize security test patterns and define the way to generate tests from the test generation model.
- Model-based statistical testing (MBSTT) is a black-box testing technique that enables the generation of representative tests from the tester's or user's perspective. MBSTT has been extended for risk-based testing [97] and applied to safety-critical embedded systems as well

as to security relevant systems. MBSTT has been used to construct the generic test models and to automatically generate test cases from the concrete test models.

- Zech [96] proposes new model-based based approach for risk-based security testing of cloud environments is proposed. Negative security requirements are derived from risk analysis in order to cover those security aspects not covered by existing positive requirements. Misuse cases are then defined based on the negative requirements. The approach of Zech is further outlined in Section 4

3.2.4 Summary and Outlook

Security testing like any other testing follows a series of activities and uses artifacts that aim to systematically plan, design, specify, realize, and execute tests, and to evaluate the test results and, if needed, to readjust the planning etc. Facing the increasingly complex difficulties according software development and the validation and verification of their functional and non-functional properties, which require teams and clear structures for work, in the last years several software development process models have emerged. These process models coordinate the activities and define the clear structure needed for modern software development. Besides models like the V-Modell XT, and the Rational Unified Process, there exists a so-called W-Model [78]. This model integrates inter alia the phases of requirements analysis, design, and implementation with the phases of test specification, execution, and evaluation. Moreover, dedicated testing methodologies like TMAP [75] provide a good guidance for structuring the test phases. Security testing and model based security testing may be adopted using the principal ideas of this process model. Still, there exists only little experience in development processes that integrate security or model based security testing as a whole.

Additionally, security engineering and thus security testing is increasingly challenged by the openness, dynamics, and distribution of networked systems. Most verification and validation techniques for security have been developed in the framework of static or known configurations, with full or well-defined control of each component of the system. This is not sufficient in networked systems, where control and observation of remote (sub) systems are dynamically invoked over the network and may change over time. Last but not least MBST and the integration with risk management techniques pose additional requirements. In general we see that the following challenges need to be overcome.

- Whereas for functional software testing, several standards exist, for security testing or model-based security testing no such standards are available that could be used as clear reference. The standards and best practices that are available are on a level of abstraction, that they do not provide sufficient guidance.
- MBST is a relatively new field and especially dedicated to the systematic and efficient specification and documentation of security test objectives, security test cases and test suites. A lack of specific experience on particular techniques, their applicability and constraints for model based security testing is missing.
- Techniques for compositional MBT and MBST are only rarely known. Standards and best practices that may provide guidance do not exist.
- The relation of security properties and their implementation in applications and, as a consequence, possible variations or variation points of the testing methodologies to be defined, are not sufficiently understood. However, this issue is partially addressed by existing security testing standards (see Section 3.2.2).

The baseline for security testing aspects of the RASEN methodologies is presented below in Section 7.3

4 Risk-Based Testing and Test-Based Risk Assessment

4.1 Overview of Methodologies for Risk-Based Testing and Test-Based Risk Assessment

From a process perspective, there are essentially two ways to combine the risk assessment process with the testing process: Either enhance the risk assessment process with testing activities, or enhance the test process with risk assessment activities. We refer to the former approach as test-based risk assessment, while the latter approach is referred to as risk-based testing. In this section, we focus on areas in the test process (or risk process) where risk assessment (or testing) can be used as input.

In Figure 5, we have illustrated a standard testing process, and identified two areas where risk assessment can be used. The first area is the transition from step 1 to step 3, where the idea is to use risk assessment in order to identify and prioritize tests. The second area is the transition from step 4 to step 6, where the risk assessment can be used to prioritize the test execution and to structure/classify the test results.

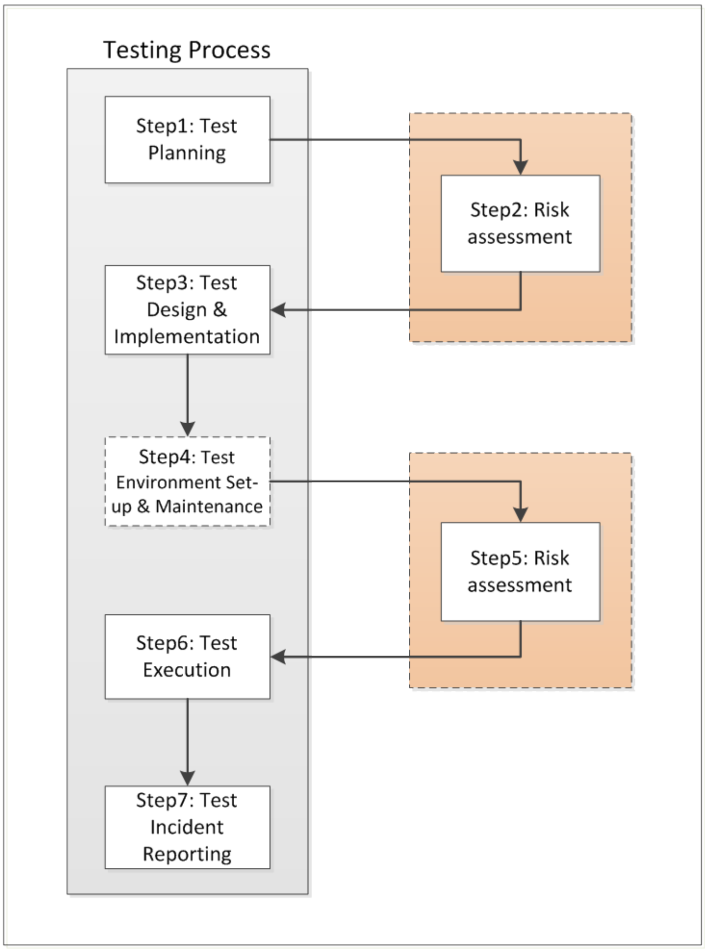
Testing Process
Step1: Test Planning
Step2: Risk assessment
Step3: Test Design & Implementation
Step4: Test Environment Set-up & Maintenance
Step5: Risk assessment
Step6: Test Execution
Step7: Test Incident Reporting

Figure 5 – Risk-based testing process

In Figure 6, we have illustrated a standard risk assessment process and identified two areas in the process where testing can be used. The first area is the transitions from step 1 to step 3. The idea here is to use test results as input to the risk identification. The other area is the transition from step 5 to step 7, where the idea is to use testing to validate the risk assessment results that have been produced in step 5, and to use the results of the test validation as input to step 7.

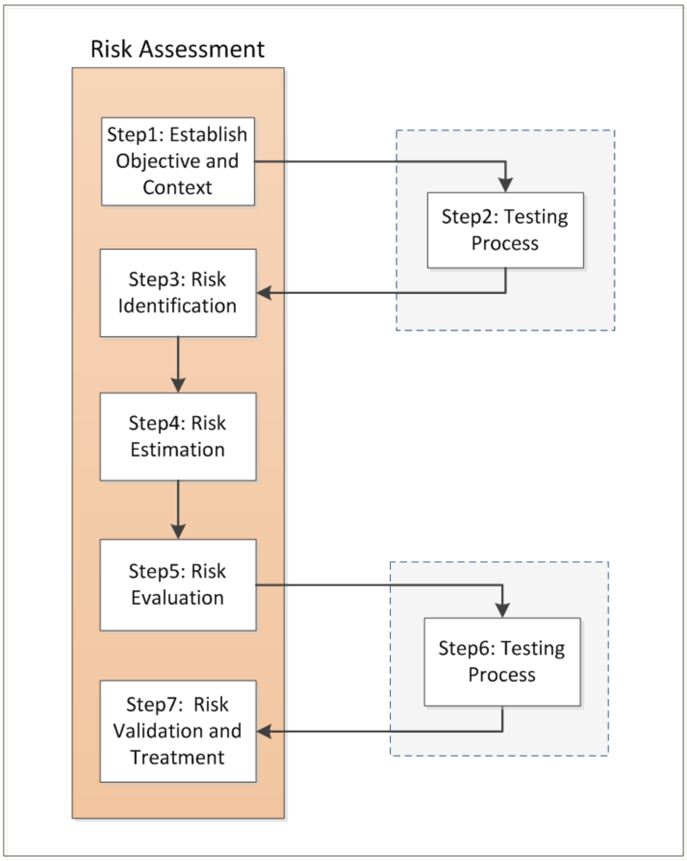

Figure 6 – Test-based risk assessment process

4.2 A Classification of Approaches to Risk-Based Testing and Test-Based Risk Assessment from a Methodology Perspective

In this section, we classify approaches to risk-based testing and test-based risk assessment according to how the “added” steps in the processes are covered (see previous section).

Almost all the approaches that combine testing and risk assessment fit best into the category of risk-based testing, i.e. risk assessment is primarily used to aid the testing. An overview of all approaches to risk-based testing that we are aware of, together with a description of how they address step 2 and step 5 in the process in Figure 5, is given in Table 1. As can be seen here, all of the approaches support step 2, and only two of them support step 5. For step 2 in the table, we have attempted to categorize the manner in which the approaches support step 2 using the following categories:

Test prioritization: The approach uses risk assessment to prioritize test design and/or implementation.

Test identification: The approach uses risk assessment (typically through fault/threat modeling) to identify test scenarios.

Test scenario generation: The approaches used risk assessment results (together with a test model) to automatically generate test scenarios or test cases.

Test model generation: The approaches used risk assessment results to generate the test model from which test cases can be derived.

Approach	Step 2	Step 5
Bach [6]	Test prioritization.	Not considered.
Redmill [61][63][64]	Test prioritization.	Not considered.
Souza et al. [77][75]	Test prioritization.	Not considered.
Bai et al. [7]	Test prioritization.	Not considered.
Felderer et al. [23]	Test prioritization.	Not considered.
Ottevanger [57]	Test prioritization.	Not considered.
Rosenberg et al. [68]	Test prioritization.	Not considered.
Wong et al. [91]	Test prioritization.	Not considered.
Murthy et al. [56]	Test identification.	Not considered.
Zech et al. [95][94][93]	Test identification.	A misuse case model (from the risk assessment) is used as input to a *data fuzzing* activity that generated fuzzed tests to be executed.
Casado et al. [14]	Test identification.	Not considered.
Kumar et al. [44]	Test identification.	Not considered.
Gleirscher [24]	Test identification.	Not considered.
Erdogan et al. [20]	Test prioritization and identification.	Not considered.
Chen et al. [15][16]	Not considered.	Risk assessment is conducted to prioritize and select test cases.
Stallbaum et al. [79]	Test model annotated. Test scenario generation and prioritization.	Not considered.
Kloos et al. [43]	Test model generation.	Not considered.

Table 1 – Approaches to risk based testing

In Table 2, we have listed all approaches related to test-based risk assessment that we are aware of, and indicated how or if they support step 2 and step 6 of the process (shown in Figure 6). As can be seen in the table, none of the approaches support step 2, but all of them in varying degree touch upon step 6.

Approach	Step 2	Step 6
Benet [10]	Not considered.	Impact of failed test cases can be traced to so-called risk requirements whose impact on the risk picture in case they are not satisfied is clear.
Wong et al. [91]	Not considered.	Risk values of code blocks or functions can be automatically reduced when they are covered by successful test cases.
Erdogan et al. [20]	Not considered.	This is step is included as part of their process, but no particular technique for doing it is proposed.
Zech et al. [95][94] [93]	Not considered	In their framework, the test results give feedback to the test model and the risk model. But how this is done is not described.

Table 2 - Approaches to test-based risk assessment

The RASEN project's baseline for risk-based testing and test-based risk assessment is presented below in Section 7.4.

5 Continuous Security Risk Assessment Methodologies

Risk management consists of a set of coordinated activities to direct and control an organization with regard to risk [33]. The risk management process, as defined by ISO 31000 and depicted in Figure 1 (above, page 8), is a continuous process that should be an integral part of the overall management of an organization, and embedded in the organizational culture and practices.

Although the overall risk management process of an organization is continuous, the risk assessment process is not. Instead, risk assessment should be conducted from time to time on a regular basis in order to update the risk picture and determine whether the existing means for risk mitigation are adequate. Hence, the current risk picture focuses on a particular system configuration at a particular point in time and is therefore valid only under the assumptions made when it was established [47]. At the same time, the need for handling change is well recognized my most risk assessment methods. As prescribed by the monitor and review activity of the ISO 31000 process, risk management should detect "changes in the external and internal context, including changes to the risk criteria and the risk itself, which can require revision of risk treatment and priorities"[40]." [33]. However, for large, heterogeneous, compositional and highly dynamic systems that rapidly evolve, the risk picture rapidly evolves too. Most established risk management and risk assessment methods are not well-equipped to handle such rapid changes in a methodic and systematic way.

A straightforward solution is to conduct a new risk assessment from scratch whenever a possibly risk relevant change has occurred. Needless to say, such an approach is not very adequate as it is very time and resource consuming, and it also often implies conducting exactly the same analysis again for the parts that are not affected by the changes. Instead, methods should be equipped with guidelines and procedures to support continuous risk assessment where changes to risks are continuously identified and reassessed. As discussed in RASEN deliverable D3.1.1 there are several techniques, modeling languages and tools that provide some support for different specific tasks and challenges related to continuous assessment. At the level of methodology and process, the state of the art is much sparser, although some approaches have been proposed in the recent years. In the following we present and discuss the most important ones.

The method proposed in [46] takes as starting point the risk management process and guidelines of ISO 31000, extending these with support for continuous assessment of changing and evolving systems. The guiding principle is that only the parts of the target of analysis that are affected by change should be reassessed in each of the iterations. In particular, each of the risk management activities from context establishment, via risk assessment, to risk treatment is extended with guidelines for how to make change explicit, how to document the changes, and how to systematically track changes from the target of analysis to the risk models. The approach relies on models or other documentation of both the target of analysis and the risks that are prepared for change management. Such a preparation is made in the first iteration of the process, which mostly is conducted according to the standard process. The most important novelty of this initial iteration is that the relations between the target description and the risk model are identified and documented. Based on these relations, guidelines are provided for how to conduct reassessments whenever triggered or required by system changes. During context establishment, all such changes are explicitly documented, including system changes, changes in assets, changes in the environment and changes in the risk criteria. Also during risk assessment the changes are documented as part of identification of changes in threats, vulnerabilities, unwanted incidents, and so forth.

As explained in Section 2, many of the established security risk assessment methods can be understood as instantiations of the ISO 31000 process. A core idea of the extension of ISO 31000 in [46] to support change management is that the additional guidelines can be likewise instantiated by other methods. Hence, the extended guidelines should enable continuous risk assessment using approaches such as OCTAVE, CRAMM and CORAS. This is demonstrated with CORAS in [46], thereby providing support for continuous risk assessment using this model-driven approach. This CORAS extension is moreover supported by tool and modeling support as discussed in more details in RASEN D.3.1.1.

The model-driven approach to security risk management presented in [29] uses enterprise architectures as the basis for security management and risk assessment. The idea is to provide an integrated description of an organization's structure, processes and underlying ICT infrastructure, thereby bridging the technical and business oriented views on information security. The method defines a security management process that includes the standard risk assessment process as defined by e.g. ISO 31000. Moreover, in order to handle changes and support reassessment and continuous risk management, the approach comes with support for identifying and modeling dependencies between different views. The system views include organizational view, business process view, informational view, technical view and physical view, and more. By using the dependencies, changes are systematically traced to support structured reassessment of security risks.

Secure software engineering aims to take security considerations into account throughout the whole software development life cycle (SDLC) [40] [54]. The conjecture is that to "be most effective, information security must be integrated into the SDLC from system inception", and that early "integration of security in the SDLC enables agencies to maximize return on investment in their security programs" [40]. In this setting security risk assessment is a crosscutting activity with the potential of supporting all of the (security) engineering activities of the SDLC. Because the SDLC is often iterative and continuously evolves, it should be supported by methods for continuous risk assessment. In [11] the risk assessment process is integrated with the system development process using a modular approach. In particular, the problem addressed is that with a modular and component-based system development process, there is a need to understand both the security risks for each component and also how risks should be composed when components are. The risk assessment process is based on CORAS and makes use of all steps of the CORAS process. The CORAS process is adapted to the component-based development process, providing support for identifying, evaluating and mitigating security risks of software components in the development process. The method is supported by techniques for modular risk assessment to enable risk assessment of composed software components.

Risk monitoring is a means to facilitate continuous risk assessment by continuously updating the risk picture based on monitored indicators. The approach proposed in [65] defines a method for risk monitoring that extends the standard risk assessment process. The method is divided into three steps that need to be conducted before the risk monitoring can start. Step 1 is to perform an initial risk assessment of the system. This corresponds to the three risk assessment steps depicted in Figure 1. The purpose is to provide information about relevant risks, and to do a rough risk estimation to decide which risks need to be monitored. Risk models are built to document how threats may exploit vulnerabilities to initiate threat scenarios that may lead to unwanted incidents. Step 2 is to identify relevant key indicators for the risks to be monitored, and is done based on the results of Step 1. Step 3 is to find functions for likelihood, consequence and risk values to calculate these values based on the monitored indicators. After completing these steps, the risk assessment results as presented in the risk models can be continuously updated while the indicators are monitored and the monitored values are aggregated by using the defined functions.

Systems of systems are collections of systems that are interconnected by the exchange of services. Due to their, possibly complex, dependencies and dynamic nature, traditional methods for risk assessment may be insufficient for adequately managing the involved security risks. In [45] a method is proposed for capturing and monitoring the impact on service dependencies on the security of provided services. The method is divided into four steps as follows. Step 1 is to document the system of systems, roughly corresponding to the context establishment activity of the ISO 31000 process. The step includes modeling the system of systems and capturing the service dependencies. Step 2 is to analyze the impact of service dependencies on risk to security of provided services. This partly covers the risk identification of the ISO process, but also includes asset identification. The main purpose is to understand the potential impact of service dependencies on the identified security assets. Step 3 is to identify indicators for the system of systems. This includes identification of the security risks to be monitored, and identification of relevant indicators to enable the monitoring. Step 4 is to specify the design and deployment of identified indicators for the system of systems. The design specifications are algorithms for how to aggregate indicator values and map them to the likelihood and consequence values that are used in the risk assessment.

The RASEN project aims to develop methodological support for continuous risk assessment, and will make strong use of two principles, namely compositional risk assessment and test-based risk assessment. Compositional methods facilitate continuous risk assessment since they allow large, complex systems to be reassessed by addressing only the parts that have changed. As seen in this section, the state of the art is very sparse on methodological support for compositional assessment. Regarding test-based risk assessment, either by active or passive testing, there is a potential for leveraging methodological guidelines for building and utilizing traceability between the target model and the risk model. Such traceability facilitates the identification of risks that may be affected by monitored system changes, and the identification of risks that should be reassessed based on test results.

6 Legal Risk Assessment Methodologies

In this section we provide an overview of existing methodologies and related approaches for legal risk management.

This section is structured as follows. First, in Section 6.1, we briefly introduce and define the notion of legal risk management. Subsequently, the literature on legal risk management is reviewed, distinguishing between two primary perspectives, namely respectively, approaches with and without a methodology. Much of the literature on legal risk management provides important background knowledge without actually defining an explicit and well-defined methodology. Such literature is briefly included here in Section 6.2, because many aspects described in that literature could be usefully included in such a methodology. This then lays the foundation for Section 6.3, surveying approaches to legal risk management that do include a methodology. Then, Section 6.4 surveys selected approaches in compliance management, which can be seen as a possible complement to existing legal risk management methodologies.

6.1 Legal Risk Management

There is currently no agreement in the literature about what legal risk management is or how it should be applied.

For the purposes of this report we will use the following working definition: When risk management is applied with a focus on legal issues, we call it legal risk management. This definition is rather wide and also includes the assessment of legal issues in risk management that primarily focuses on other perspectives (e.g., in enterprise risk management or in ICT security risk management).

Approaches to legal risk management are examined in light of the following RASEN research questions:

- RQ8: What is a good methodology for helping organizations understand and assess security related legal norms, and what conceptual framework is needed for this?
- RQ9: How can we use security testing to provide evidence of compliance to security related legal norms?

These research questions reflect the focus of the RASEN project on ICT security issues and legal compliance.

6.2 Informal Approaches to Legal Risk Management

A significant part of the literature does not propose an explicit and well-defined methodology for legal risk management, but rather provides complementary perspectives to such a methodology. While such approaches cannot be discussed in detail within the scope of this report, it is useful to at least provide a brief overview of such perspectives, because they may be combined with, or integrated in, a methodology for legal risk management.

6.2.1 Preventive Law

According to Louis M. Brown [12] and others, preventive law is comprised of legal and practical principles for anticipating and minimizing legal risks. The goal of preventive law is to provide for the 'legal health' of individuals and business entities. Preventive law is based on the assumption that the most successful medical treatment is prevention, and seeks to transfer this assumption from the medical to the legal context. Preventive law focuses, for example, on how to prevent liability or how to perform a legal (compliance) audit. Reid [66] has observed that '…preventive law does not appear to involve as systematic and consistent an approach to identifying and managing legal risk as legal risk management.' One key difference between traditional preventive law approaches and risk management is the degree to which values are assigned to risks. Risk management uses values to prioritize risks in order to facilitate rational decision-making about different treatments. I have not found this aspect addressed within preventive law. Hence, although preventive law, to a certain degree, does include elements of methodologies including, e.g., periodic legal check-ups, legal autopsies, preventive analyses, etc., it is not based on a structured evaluation of risks as typically provided for in risk management.

6.2.2 Best Practices: Managing Legal Risk Effectively

Best practices in managing legal risk are not always easily accessible for research purposes. However, a recent study by RSG Consulting [69], commissioned by the law firm Berwin Leighton Paiser LLP, shows how law departments of companies (in-house counsel) say they can effectively manage legal risk. The report is based on interviews with 12 leading general counsels from a variety of businesses to understand their different approaches to legal risk management and to collate key themes. This report identifies a marked change over the last three years in the way businesses manage legal risk. Earlier, most institutions managed legal risk on an ad-hoc, event-driven basis, relying principally on the commercial awareness and judgment of their in-house or external legal advisers. The research shows a trend towards introducing process to support lawyers in making risk decisions, escalating issues and integrating with the risk management frameworks and risk culture of the broader business. For example, strategically prioritizing the most significant legal risks is today viewed as essential when considering the most effective allocation of internal and external resources. According to this report, good legal risk management is characterized by a specific legal risk management process, culture, knowledge and communication, structure (responsibility) and individuals. Amongst these, the legal risk management process itself is arguably the element that is closest to the interest of this RASEN report, as it most closely resembles what is here understood as a methodology. Regarding the process, the report notes the following ([69], page 7):

“The majority of General Counsel said that the most important indicator of good legal risk management was having robust and clearly defined processes to evaluate risk on a continuous basis. In particular, assessment processes must be specific to legal risk management, not borrowed from accountancy frameworks or imposed by an audit function. For these frameworks to be effective, they need to be adapted to the legal context. The result of the process must be good reporting, ensuring critical risks are made visible to the right people as early as possible.”

Thus, while the RSG Consulting report emphasizes the need for a clear methodology (a process), it provides few details about the details of said process.

6.2.3 Australian Standards Handbook on Legal Risk Management

The standardization body Australian Standards has published a dedicated handbook on legal risk management [81]. The legal risk management handbook seeks to explain how the Australian Standard on risk management, a predecessor to ISO 31000, can be applied to the delivery of services by the legal profession. According to this handbook, legal risk management has a range of applications related to (page 13):

“[P]lanning and making decisions about significant issues: for example, when considering changes in policy, introducing new strategies and procedures, managing major projects, […] managing internal organizational changes or managing potentially sensitive issues.”

The use of legal risk management is in this handbook is explained through a small case study, which is here summarized to illustrate the kernel ideas. In a medium-sized company, the managing director notices certain misconduct by the Chief Information Officer (CIO) and needs to consider terminating the CIO. This case is explained in two scenarios, in order to exemplify the difference between a thorough risk assessment and a quick and unsatisfactory decision based on an insufficient basis.

In scenario 1, the CIO is terminated, but this termination fails, because it causes a number of negative consequences to the company, largely due to the company’s non-compliance with workplace surveillance laws.

In scenario 2, the involved actors conduct a legal risk management workshop, through which they identify the non-compliance with workplace surveillance laws. They succeed in managing this issue, and the CIO resigns voluntarily (for no apparent reason).

This is obviously an artificial case created to illustrate the use of diligently managing risk in the day-to-day practice. However, both the quite general advice given in the handbook and the example case leave a number of questions unanswered. These questions relate to how exactly risk management works in a legal context, and how the interplay between facts and law should be adequately modeled and assessed in legal risk management.

6.2.4 McCormick

McCormick [52] approaches legal risk management from the perspective of financial law, because *legal risk* has received some attention in the banking sector. McCormick's approach thereto consists of identifying, assessing, monitoring and mitigating legal risks, but he does not describe these steps in any detail. It is, however, noteworthy that McCormick emphasizes that 'it is impossible to ascribe rigid mathematical measurement formulae to operational and legal risks' ([52] at Section 10.36.). This remark points to the mathematical formulae utilized in financial risk management, which indeed appear much too formalized in our context. Nevertheless, McCormick seems to believe that legal risk assessment is possible, provided that other, less formal approaches are used. Although McCormick stresses the need for a clear methodology for legal risk management, he provides no single detailed methodology.

6.2.5 Wahlgren

Wahlgren's [90] study on *legal risk analysis* compares typical risk-related legal work tasks, on one hand, to risk analysis methods, on the other hand. Wahlgren concludes that, e.g., fault tree analysis, matrices, checklists, etc., can indeed support many legal tasks as complementary methods. Wahlgren focuses on a review of particular analytical methods, rather than on drafting a singular and consistent methodology. Wahlgren's conclusion (that legal tasks can be supported by risk analysis methods) preliminarily indicates that it is possible to design one or more specialized methodologies based on a suitable combination of such methods. According to Wahlgren, however, a number of challenges and limiting factors are related not only to the traditions and education of lawyers, but also to the nature of law, where discretionary expert legal judgments play a significant role. This nature may somewhat limit the possibility of formalizing the reasoning about risks in a legal context.

6.3 Methodologies

The previous section has reviewed approaches to legal risk management that show the need for having a methodology, but which do not themselves specify such a methodology. This section reviews some approaches to legal risk management that include explicit and well-defined methodologies.

6.3.1 Reid

Reid's [66] framework for legal risk management is an effort to apply to a legal analysis an earlier version of the Australian/New Zealand Standard for Risk Management, a predecessor to ISO 31000. In Reid's methodology, each proposed step corresponds to one of the steps in the Australian Standard. In essence, Reid proposes to (1) identify *objectives* which are somehow related to legal compliance and the protection of legal interests, (2) identify and (3) analyze *legal risks*, (4) evaluate and select *legal risk management strategies*, which are subsequently (5) implemented and (6) monitored.

Reid uses her framework primarily to discuss legal risks related to eCommerce. However, it is not clear whether the aim of and context for Reid's framework should be considered limited in any way. Essentially, her proposed methodology applies to any business context in which it may be relevant to reason about the existence and treatment of legal risks. Reid proposes a sufficiently detailed methodology, which clearly focuses on legal issues. However, Reid's work predates ISO 31000 and is not conceptually in line with recent developments in risk management.

6.3.2 Extending ISO 31000

Mahler [49] has shown that legal risk management can be organized based on the above-mentioned general risk management process described in ISO standard 31000 (see 38 and above, Section 2). His methodology consists of five discrete process steps, and two continuous practices, as displayed below in Figure 7.

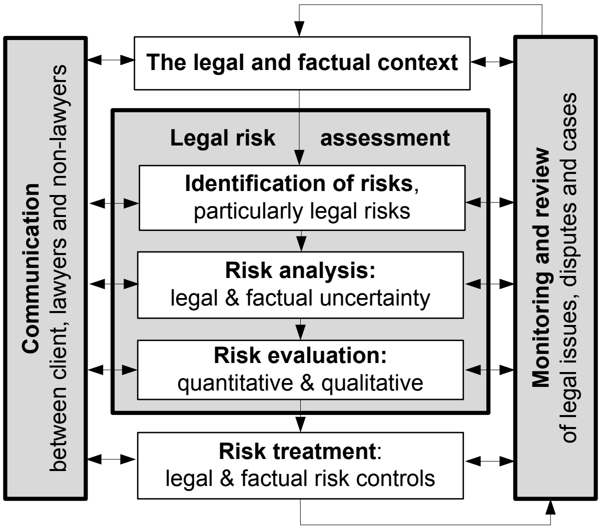
The legal and factual context
Communication between client, lawyers and non-lawyers
Legal risk assessment
Identification of risks, particularly legal risks
Risk analysis: legal & factual uncertainty
Risk evaluation: quantitative & qualitative
Risk treatment: legal & factual risk controls
Monitoring and review of legal issues, disputes and cases

Figure 7 – Legal risk management process

The risk management process starts with establishing the context. This includes, from the legal perspective, both the legal context and what may be called the factual context. The latter is the project, system or organization under analysis. The former centers on understanding the body of law (and eventually contracts) applicable to whatever is the target of analysis. After having identified the context, the process continues with the legal risk assessment.

First, risk is identified – and this includes, particularly, legal risk. For example, this can be envisaged as a brainstorming activity, in which qualified experts contribute with their knowledge and experience. In essence, the brainstorming should concentrate on identifying risk events (what may happen in this project?) and assessing these events under the law.

In the second step, the identified risk is analyzed, in order to understand the risk and to estimate risk levels. For every identified risk one would thus seek to establish whether it is a high risk or a low risk. This risk level depends in practice on the likelihood of occurrence and the severity of the consequences of the risk.

Once the identified risks have been analyzed, we can proceed to the third step, the risk evaluation, which commences the decision-making upon risk. At this stage, the results of the risk analysis are evaluated with reference to the applicant's risk criteria (risk aversive or risk seeking?). Based on these criteria, some risks may be acceptable, while others may need to be treated.

The process concludes with the risk treatment, which identifies risk controls and selects those controls that should be implemented. Both legal and other risk controls can be considered.

This work provides a clear legal risk management process. By following the risk management process of ISO 31000 and the related risk management vocabulary, this approach can easily be integrated with other approaches in fields like information security. Yet, in light of the research questions emphasized above in Section 6.1, this work lacks several aspects. First, the methodology is generic, and not directly related to information security. Thus, it is possible that the risk management concepts are not fully aligned with relevant conceptual frameworks in information security. Second, this methodology was primarily developed with a focus on contracts and lacks an explicit focus on compliance, which is of key importance for the RASEN project. Therefore, this approach should be extended with an explicit focus on compliance management.

6.4 Compliance Management

A number of authors link legal risk management to compliance management. For example, Iversen [36] presents legal risk management as an integral part of corporate governance and focuses particularly on compliance. While compliance management is sometimes addressed as a separate issue, it is increasingly combined with risk management and with corporate governance, rendering the combination governance, risk management and compliance (GRC, [82][27]).

While there is literature on compliance in the information security context, such compliance is not necessarily put into a (legal) risk management context (e.g., [89]). When compliance is put into a risk management context (e.g., [50]), it is often discussed within an enterprise risk management (ERM) context, but not condensed to a methodology that is distinct from an ERM standard such as COSO [17]. Moreover, compliance also plays a role in other context, such as compliance with standards [9].

Occasionally, compliance is also mentioned in patents, such as in the Budde et al. system and method for compliance management [13]. This essentially applies a simple process, consisting in

- Identifying business processes;
- Determining compliance requirements;
- Determining compliance ownership;
- Identifying compliance/risk issues;
- Creating an action plan;
- Forwarding the action plan for resolution; and
- Reviewing the action plan results.

Notably, key elements of this process can be aligned with a standard risk management approach, but this is not done in the patented process.

The Australian standard on compliance programs [5] describes four sets of principles:

- Commitment
- Implementation
- Monitoring and measuring; and
- Continual improvement.

The first set of principles, focusing on commitment, ensures that there is commitment by the governing body and top management to effective compliance. This includes endorsement by top management, appropriate allocation of resources, and identification of compliance obligations. The second set of principles focuses on implementation, including assigning responsibility for compliant outcomes, training, incentives and controls for compliance. The third set of principles focuses on monitoring and measuring compliance, in order to ensure that the organization is able to demonstrate its compliance program through both documentation and practice. Finally, the standard focuses on continual improvement, based on regular reviews.

7 RASEN baseline

7.1 Introduction

This baseline section provides an initial assessment of the state of the art and specifies the starting point for the further development in the RASEN project. The structure of the section mirrors the preceding sections that have dealt with the respective types of methodologies:

- Section 7.2 describes the baseline for security risk assessment;
- Section 7.3 provides the baseline for security testing;
- Section 7.4 focuses on methods to combine, respectively, risk-based testing and test-based risk assessment
- Section 7.5 addresses continuous security risk assessment; and
- Section 7.6 relates to the baseline for the legal risk assessment.

7.2 Security Risk Assessment Methodologies

While developing methodological guidelines and principles to tackle the research challenges of RASEN, we will build closely on established standards such as the ISO 31000 risk management standard and the ISO/IEC 27005 standard on information security risk management. On the one hand, this is to ensure that the methods and processes that we develop are as much as possible compliant with existing industrial practices and that the RASEN methodologies can be applied while maintaining such compliance. On the other hand, we aim in this project to develop principles and guidelines that are as generally applicable as possible, and that can be instantiated in several of the existing and well-established approaches to security risk assessment. For example, in the same way as OCTAVE, CORAS, CRAMM, EBIOS, etc. instantiates the ISO 31000 process, RASEN should enable the same kind of instantiations of the extended and specialized guidelines that will be developed in the project.

7.3 Security Testing Methodologies

Current methodologies for security testing neither explicitly address the integration with the software development processes, nor do they especially address component-based systems or systems that evolve over time. Designing a security testing methodology that covers the complexity of security testing for evolving, componentized systems has to be carefully aligned with process models as described in Section 3.2.1 and additionally must have a strong emphasis on the operational and maintenance phases of the system. Especially the latter is important to be able to usefully perform targeted security checks even with a changing security situation and the emergence of new vulnerabilities (e.g. emerging zero-day-exploits).

A general guidance on which artifacts are relevant for testing is defined in IEEE 829 [28]. Moreover, a dedicated and practically relevant testing methodology like TMAP provides a good basis for structuring the test phases and test processes. For integration with the development process we will consider process models like RUP and the W-Model (see Section 3.2.1). The current security testing standards from Section 3.2.2 may help to organize the RASEN approach on a high level. The standards provide the general phases for security testing (i.e. Information Gathering, Testing, Collecting/Assessing Results), and with variations that depend on certain kind of technologies and their related potential vulnerabilities.

A main focus in RASEN is on model-based security testing and compositional management of security risks and security tests. Therefore the security testing methodology developed in the project needs to have an explicit relation to risk assessment and risk management methodologies. Moreover, the methodology must be able to handle the decomposition of tests or test suites along the decomposition of risks, or the decomposition of the system to be tested. Glimpses of the latter are already addressed with the integration of testing approaches in compositional system development processes as reported in Section 3.2.1. For MBST methodologies the ETSI MBT standard provides a good initial template that has to be refined with respect to security testing and risk-based security testing.

7.4 Risk-Based Testing and Test-Based Risk Assessment Methodologies

In the state of the art in Section 4, we described two generic processes: one risk-based testing (RBT) process and one test-based risk assessment (TBR) process. We also gave an overview approaches to RBT and TBR and classified them according to which steps of the processes they covered. Based on this overview, the approaches of Zech et al. [95],[94],[93] and Erdogan et al. [20] are the most comprehensive.

Zech et al. propose a framework (shown in Figure 8) in which both Step 2 and Step 5 of the RBT process, as well as Step 6 of the TBR process, are considered. However, the framework is intended to target cloud-based systems. It is also heavily dependent on model-transformations. For instance, the risk model is intended to be automatically derived from the model of the system under test. However, how this is achieved is not explained, and it is unclear to us how this may be achieved for the kind of risk models that will be considered in the RASEN project.

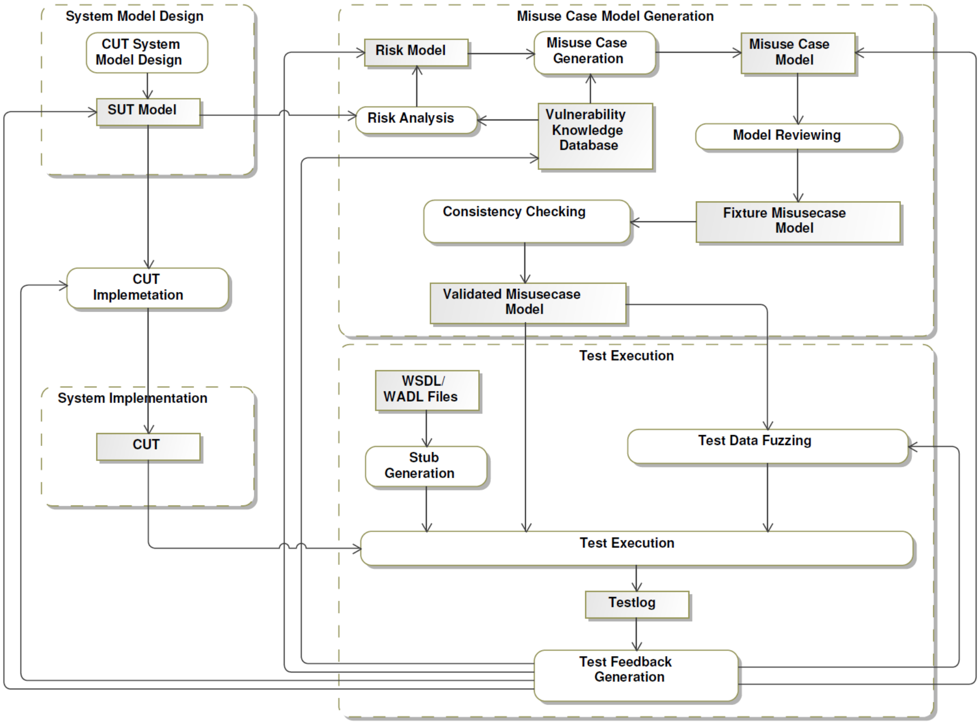

Figure 8 – Framework of Zech et al. [93]

The process used by Erdogan et al. [20] (shown in Figure 9) supports Step 2 of the TBR process and Step 6 of the RBT process. The process is furthermore applicable to many kinds of systems (not just cloud systems as is the case with Zech et al.) and it is based on precise risk models that enable tool-support in many steps of the process. We therefore believe that the approach of Erdogan et al. [20] can be a good starting point for the RASEN project.

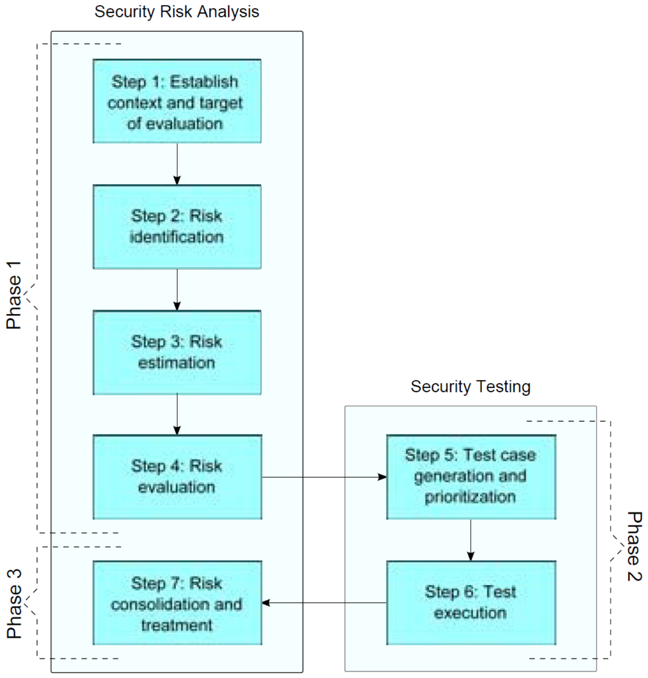
Security Risk Analysis
Step 1: Establish context and target of evaluation
Step 2: Risk identification
Step 3: Risk estimation
Step 4: Risk evaluation
Phase 1
Security Testing
Step 5: Test case generation and prioritization
Step 6: Test execution
Phase 2
Step 7: Risk consolidation and treatment
Phase 3

Figure 9 – Process used by Erdogan et al. [20]

7.5 Continuous Security Risk Assessment

As discussed in Section 5, several modeling and assessment techniques have been proposed during the last few years on how to handle change and facilitate continuous security risk assessment. However, much less work has been done at the level of methodological process and guidelines. While building on the general principles of international standards, such as ISO 31000, RASEN will explore the potential of using as baseline the more highly iterative processes as proposed in e.g. [46]. Continuous security risk assessment will generally be much more iterative than traditional methods. This is because continuous assessment makes frequent or continuous updates of the risk models and risk estimates based on reassessments, monitoring, testing results, and so on. The appropriate and adequate method and process for continuous risk assessment depend much on the assessment and modeling techniques that support such a method. We will therefore make sure that our methodological support for continuous risk assessment as much as possible makes use of our tools and techniques. One important principle that RASEN will leverage is that of compositional risk assessment. Compositionality in the setting of security risk management is virtually not supported at all by state of the art approaches; an important objective of RASEN is therefore to develop techniques to support composition, and next to utilize these techniques when developing methodological guidelines. This method should facilitate continuous risk assessment by allowing separate parts of the target system to be analyzed independently and in isolation. We envisage an iterative risk assessment process where a more lightweight reassessment can be conducted of smaller parts of the target to ensure a risk picture that is kept continuously valid while systems change.

7.6 Legal Risk Assessment

Generally speaking, the aim of all legal risk management is the adequate management of risk in the legal context. A useful point of departure for legal risk management is to base it on ISO 31000. This has the advantage of providing a common set of concepts and a process that can be integrated with other risk management approaches, including those related to information security. Thus, the legal risk management methodology described above in Section 6.3.2 can be taken as a point of departure for the RASEN baseline. Yet, this methodology should be extended with elements from compliance management, for at least two reasons. First, the RASEN research project focuses on the ICT security perspective, and legal issues (including risks) primarily play a role in the context of compliance with

ICT security obligations. Second, this combination of compliance management with legal risk management will also facilitate the integration of compliance audits and security testing and in order to assess the degree to which an organization complies with relevant security-related norms.

8 Conclusion

WP5 of the RASEN project aims to develop methodology for compositional and continuous security risk assessment, methodology for compositional security testing, and methodology for legal risk assessment. The project will moreover develop support and guidelines for how to combine these methodologies into an overall framework where the different parts can leverage each other.

Section 2 gives an overview of the state of the art on security risk assessment methodologies. As explained, none of the existing methodologies provide appropriate solutions to the research questions raised by the project. However, the project will carefully build on international standards and best practices when targeting the R&D objectives. In particular, the methodologies that we develop in WP5 should as much as possible be compliant with established standards such as ISO 31000 and ISO/IEC 27005.

Section 3 provides an overview on methodologies and best practices for testing and security testing. While the outlined testing methodologies provide a good guidance on how testing activities can be generally aligned with software development processes, the presented security testing methodologies present additional activities that are necessary for security testing and model-based security testing.

Section 4 discusses the state of the art of methodologies that combine risk assessment and testing. RASEN distinguishes between risk-based testing (risk assessment to improve the testing) and test-based risk assessment (using testing to improve the risk assessment). Section 4 presents one generic process for each alternative. Furthermore the relevant approaches from the literature are classified according to which steps of the processes they considered. Finally, an approach is identified that will be used as a starting point for the research within RASEN.

RASEN envisages a methodology for continuous security risk assessment and testing that is much more iterative than traditional and established approaches. Continuous assessment should be enabled by rapid reassessments in order to maintain the validity of the risk assessment results while the target system or its environment changes and evolves. Security testing is considered to provide the necessary experimental feedback to improve the risk analysis and its assumptions. Considering the existing methodological support described in Section 5, there are some means for risk monitoring, security testing and traceability that can be leverage to facilitate continuous assessment. However, the state of the art is very sparse regarding the RASEN research questions on composition and decomposition of risk assessment artifacts, and on reuse of risk assessment and testing artifacts.

Moreover there is currently no approach that integrates technical risk assessment and legal risk assessment. As discussed in Section 6, while there are existing approaches for aligning legal risk management with other risk management approaches based on ISO standard 31000, these do not put sufficient emphasis on compliance management. The latter has a crucial role for the integration with technical risk management, which needs to be carried out in compliance with legal requirements.

In summary, RASEN will adopt existing methodologies if possible and useful. This applies especially to the area of testing, traditional security testing and risk-based security testing. In all other cases RASEN aims to make strong use of its own visions and will develop methodologies that integrate compositional security risk assessment and model-based security testing, that integrate technical risk assessment and legal risk assessment, and that support an efficient reuse and reassessment of existing artifacts.

References

[1] Agence nationale de la sécurité des systèmes d'information: EBIOS 2010 – Expression of Needs and Identification of Security Objectives (2010)

[2] Alberts C. J., A. J. Dorofee: OCTAVE Criteria. Technical Report CMU/SEI-2001-TR-016, CERT (2001)

[3] Amland S.: Risk based testing and metrics. In 5th International Conference EuroSTAR, volume 99, pp 8–12 (1999)

[4] Amland S.: Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study. Journal of Systems and Software, 53(3), 287–295 (2000)

[5] Australian Standard AS 3806-2006 Compliance programs.

[6] Bach J.: Heuristic risk-based testing. Software Testing and Quality Engineering Magazine, 11:99 (1999)

[7] Bai X. and R.S. Kenett: Risk-based adaptive group testing of semantic web services. In Computer Software and Applications Conference (COMPSAC'09). 33rd Annual IEEE International, volume 2, pp. 485–490. IEEE (2009)

[8] Bauer T., F. Böhr, D. Landmann, T. Beletski, R. Eschbach, J. H. Poore: From Requirements to Statistical Testing of Embedded Systems. Software Engineering for Automotive Systems (SEAS'07), ICSE Workshops (2007)

[9] Beck K. et al.: Manifesto for Agile Software Development (2009). [ONLINE] Available at: http://agilemanifesto.org/ [Accessed 5 December 2012]

[10] Benet A.F.: A risk driven approach to testing medical device software. Advances in Systems Safety, pp. 157–168 (2011)

[11] Brændeland G., K. Stølen: Using model-driven risk analysis in component-based development. In: Dependability and Computer Engineering: Concepts for Software-Intensive Systems, pp. 330-380, IGI Global (2011)

[12] Brown, L. M.: Preventive Law. Westport, Conn., Greenwood Press (1970)

[13] Buddle J.J., B.S. Burke, R.A. Perkins, L.E. Roday, R. Tartaglia, and I.A. Vermiglio: System and Method for Compliance Management, US Patent (2005)

[14] Casado R., J. Tuya, and M. Younas: Testing long-lived web services transactions using a risk-based approach. In Quality Software (QSIC'10), 10th International Conference on, pp. 337–340. IEEE (2010)

[15] Chen Y. and R.L. Probert: A risk-based regression test selection strategy. In Proceeding of the 14th IEEE International Symposium on Software Reliability Engineering (ISSRE'03), Fast Abstract, pp. 305–306 (2003)

[16] Chen Y., R.L. Probert, and D.P. Sims: Specification-based regression test selection with risk analysis. In Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research, pp. 1–14. IBM Press (2002)

[17] COSO: Enterprise Risk Management: An Integrated Framework. Committee of Sponsoring Organizations of the Treadway Commission (2004)

[18] DIAMONDS: Development and Industrial Application of Multi-Domain Security Testing Technologies [ONLINE] Available at http://www.itea2-diamonds.org/index.html [Accessed 5 December 2013]

[19] Emmerich, W., A.; Finkelstein, C.; Montangero, S.; Antonelli, S.; Armitage, and R. Stevens. Managing Standards Compliance. IEEE Transactions on Software Engineering 25(6), 836-851 (1999)

[20] Erdogan G., F. Seehusen, K. Stølen, J. Aagedal: Assessing the usefulness of testing for validating the correctness of security risk models based on an industrial case study. To appear in Proc. International Workshop on Quantitative Aspects in Security Assurance (QASA'12)

[21] ETSI (European Telecommunication Standards Institute): Methods for Testing & Specification (MTS); Model-Based Testing (MBT); Requirements for Modelling Notations, ES 202 951 v 1.1.1 (2011)

[22] Felderer M., B. Agreiter, P. Zech, R. Breu: A Classification for Model-Based Security Testing. The Third International Conference on Advances in System Testing and Validation Lifecycle (VALID'11) ISBN: 978-1-61208-168-7 (2011)

[23] Felderer M., C. Haisjackl, R. Breu, and J. Motz: Integrating manual and automatic risk assessment for risk-based testing. Software Quality. Process Automation in Software Development, pp. 159–180 (2012)

[24] Gleirscher M.: Hazard-based selection of test cases. In Proceeding of the 6th international workshop on Automation of software test, pp. 64–70. ACM (2011)

[25] Hernan S., S. Lambert, T. Ostwald, A. Shostack: Threat modeling – uncover security design flaws using the STRIDE approach (2006) [ONLINE] Available at: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx [Accessed: December3, 2012]

[26] Herzog P.: OSSTMM 3.0. -- Open-Source Security Testing Methodology Manual; Institute for Security and Open Methodologies (2010)

[27] Hoffmann M.: Governance, Risk, and Compliance (GRC), An integrated approach (2010)

[28] IEEE 829-2008: IEEE Standard for Software and System Test Documentation, ISBN 978-0-7381-5747-4, 2008.

[29] Innerhofer-Oberperfler F., R. Breu: Using an enterprise architecture for ICT risk management. In: Information Security South Africa Conference (ISSA'06), (2006)

[30] ISO 27000. International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27000 Information technology – Security techniques – Information security management systems – Overview and vocabulary (2009)

[31] ISO 27001: International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27001 Information technology – Security techniques – Information security management systems – Requirements (2005)

[32] ISO 29119: International Standards Organization. ISO 29119 Software and system engineering - Software Testing-Part 2 : Test process (draft), (2012)

[33] ISO 31000: International Standards Organization. ISO 31000:2009(E), Risk management Principles and guidelines, (2009)

[34] ISO Guide 73: International Organization for Standardization: ISO Guide 73 Risk management – Vocabulary (2009)

[35] ISO/ 27005: International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27005 Information technology – Security techniques – Information security risk management (2011)

[36] Iversen, J.: Legal Risk Management I Private Virksomheder. 2. ed. København: Thomson, (2007)

[37] Jürjens J.: Secure Systems Development with UML. Springer (2005)

[38] Jürjens J.: Model–based security testing using UMLsec, Electronic Notes on Theoretic Computer Science, 220(1) (2008)

[39] Karolak D.W.: Software Engineering Risk Management (Practitioners). Wiley-IEEE Computer Society (1995)

[40] Kissel R., K. Stine, M. Scholl, H. Rossman, J. Fahlsing, J. Gulick: Security Considerations in the System Development Lifecycle. National Institute of Standards and Technology. NIST Special

Publication 800-64 Revision 2 (2008)

[41] Kitchenham B. and S. Charters: Guidelines for performing systematic literature reviews in software engineering. EBSE 2007-001, vol. 2, (2007)

[42] Kling M.: Primer on SAG's ARIS Solution Governance, Risk & Compliance, Business Development GRC, Software AG (2011)

[43] Kloos J., T. Hussain, and R. Eschbach: Risk-based testing of safety-critical embedded systems driven by fault tree analysis. In Software Testing, Verication and Validation Workshops (ICSTW'11), IEEE Fourth International Conference on, pp. 26–33. IEEE (2011)

[44] Kumar N., D. Sosale, S.N. Konuganti, and A. Rathi: Enabling the adoption of aspects-testing aspects: A risk model, fault model and patterns. In Proceedings of the 8th ACM international conference on Aspect-oriented software development, pp. 197–206. ACM (2009)

[45] Ligaarden O. S., A. Refsdal, K. Stølen: Using indicators to monitor security risk in systems of systems: How to capture and measure the impact of service dependencies on the security of provided services. In: IT Security Governance Innovations: Theory and Research, pp. 256-292. IGI Global (2012)

[46] Lund M. S, B. Solhaug, K. Stølen: Risk analysis of changing and evolving systems using CORAS. In: Foundations of Security Analysis and Design VI (FOSAD VI). LNCS, vol. 6858, pp. 231–274. Springer (2011)

[47] Lund M. S., B. Solhaug, K. Stølen: Evolution in relation to risk and trust management. Computer 43(5), 49–55 (2010)

[48] Lund M. S., B. Solhaug, K. Stølen: Model-Driven Risk Analysis – The CORAS Approach. Springer (2011)

[49] Mahler, T.: Legal Risk Management: Developing and Evaluating Elements of a Method for Proactive Legal Analyses, with a Particular Focus on Contracts. PhD thesis, University of Oslo, (2010)

[50] Marchetti, Anne M.: Enterprise Risk Management Best Practices. From Assessment to Ongoing Compliance. Hoboken, N.J.: Wiley, (2012)

[51] Masson A., M.-L. Potet, J.Julliand, R.Tissot, G.Debois, B.Legeard, B. Chetali, F. Bouquet, E. Jaffuel, L. Van Aertrick, J. Andronick, A. Haddad: An access control model based testing approach for smart card applications: Results of the POSE project, JIAS, Journal of Information Assurance and Security, 5(1), 335–351 (2010)

[52] McCormick, R.: Legal Risk in the Financial Markets. Oxford: Oxford University Press, (2006)

[53] Microsoft Solutions for Security and Compliance and Microsoft Security Center of Excellence: The Security Risk Management Guide (2006)

[54] Microsoft: Security Development Lifecycle. [ONLINE] Available at: http://www.microsoft.com/security/sdl/ [Accessed 20 December 2012]

[55] Murthy K. K., K. R. Thakkar, S: Laxminarayan: Leveraging Risk Based Testing in Enterprise Systems Security Validation. In: Proc. First Int Emerging Network Intelligence Conf, pp. 111-116 (2009)

[56] Murthy K.K., K.R. Thakkar, and S. Laxminarayan: Leveraging Risk Based Testing in Enterprise Systems Security Validation. In 2009 First International Conference on Emerging Network Intelligence, pp. 111–116. IEEE (2009)

[57] Ottevanger I.: A risk-based test strategy. IQUIP Informatica B. V, pp. 1–13, (1999)

[58] OWASP Testing Guide v3.0 (2008) [ONLINE] Available at: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf [Accessed 5 December 2012]

[59] Peltier T. R.: Information Security Risk Analysis, 2nd edn. Auerbach Publications, Boca Raton (2005)

[60] Qin Z., Y. An, and Y. Chen: Research of risk-based testing for web applications. Computer

Engineering and Applications, 39(1),157–159, (2003)

[61] RASEN: Baseline for compositional risk-based security testing. RASEN project deliverable D4.1.1 (2013)

[62] Redmill F.: Exploring risk-based testing and its implications. Software Testing, Verification and Reliability, 14(1):3–15, (2004)

[63] Redmill F.: Risk-based test planning during system development. In Proceedings of the 6th National Software Engineering Conference, (2004)

[64] Redmill F.: Theory and practice of risk-based testing. Software Testing, Verification and Reliability, 15(1):3–20, (2005)

[65] Refsdal A., K. Stølen: Employing key indicators to provide a dynamic risk picture with a notion of confidence. In: Trust Management III. IFIP Advances in Information and Communication Technology, vol. 300, pp. 215–233. Springer (2009)

[66] Reid, K.: Risk-E-Business: A Framework for Legal Risk Management Developed through an Analysis of Selected Legal Risks in Internet Commerce. Ph.D. thesis, University of New South Wales, (2000)

[67] Roeleven S., M. Jorna: How to implement effective Enterprise Risk Management, Building a sustainable Governance Risk & Compliance solution (2011)

[68] Rosenberg L., R. Stapko, and A. Gallo: Risk-based object oriented testing. In Proceedings of the 24 th annual Software Engineering Workshop, NASA, Software Engineering Laboratory. (1999)

[69] RSG Consulting: Managing legal risk effectively - an evolving approach. A collection of insights from General Counsel (2012). [ONLINE] Available at: http://www.blplaw.com/index.cfm/a-framework-for-legal-risk-management/1995 [Accessed 29 January 2013]

[70] Scarfone K., M. Souppaya, A. Cody, A. Orebaugh: Technical Guide to Information Security Testing and Assessment. NIST Special Publication 800-115 (2008)

[71] Schneidewind N.F.: Risk-driven software testing and reliability. International Journal of Reliability Quality and Safety Engineering, 14(2):99–132 (2007)

[72] Secure Provision and Consumption in the Internet of Services (SPaCIoS). Project no. 257876, FP7-ICT-2009-5, ICT-2009.1.4: Trustworthy ICT 01/10/2010 - 30/09/2013: http://www.spacios.eu/ (last visited Dec 5, 2012)

[73] Secure Provision and Consumption in the Internet of Services (SPaCIoS): SPaCIoS Tool mock up, Technology survey, Validation methodology patterns v.1, Deliverable D4.1, http://www.spacios.eu/deliverables/spacios-d4.1.pdf (2011)

[74] SIEMENS: CRAMM – the total information security toolkit. [ONLINE] Available at: http://www.cramm.com/ [Accessed 06 December 2012]

[75] Sogeti: TMAP – the test management approach. [ONLINE] Available at: http://www.tmap.net/ [Accessed 06 December 2012]

[76] Souza E., C. Gusmão, and J. Venâncio: Risk-based testing: A case study. In Information Technology: New Generations (ITNG'10), Seventh International Conference on, pp. 1032–1037. IEEE (2010)

[77] Souza E., C. Gusmão, K. Alves, J. Venâncio, and R. Melo: Measurement and control for risk-based test cases and activities. In Test Workshop (LATW'09), 10th Latin American, pp. 1–6. IEEE (2009)

[78] Spillner A.: W-model - test process parallel to the development process. In: Proceedings Jornada sobre Testeo de Software (JTS'04), ITI Instituto Tecnológico de Infomática, Universidad Politécnica de Valencia, Spain (2004)

[79] Stallbaum H., A. Metzger, and K. Pohl: An automated technique for riskbased test case generation and prioritization. In Proceedings of the 3rd international workshop on Automation of software test, pp. 67–70. ACM (2008)

[80] Stallbaum H., A. Metzger, K. Pohl: An Automated Technique for Risk-Based Test Case Generation and Prioritization. In: Proceedings of the 3rd international workshop on Automation of software test, 67-70 (2008)

[81] Standards Australia, and Standards New Zealand: Legal Risk Management. Handbook 296:2007, Sydney (2007)

[82] Steinberg, Richard M.: Governance, Risk Management, and Compliance. Hoboken: John Wiley & Sons. (2011)

[83] Stoneburner G., A. Goguen, A. Feringa: Risk Management Guide for Information Technology Systems. National Institute of Standards and Technology, Gaithersburg. NIST Special Publication 800-30 (2002)

[84] Takanen A., J. DeMott, and C. Miller: Fuzzing for software security testing and quality assurance, 1st ed. Norwood, MA, USA: Artech House, Inc. (2008)

[85] The 1st International ICST workshop on Security Testing (SECTEST'08) [ONLINE] Available at: http://www.cs.colostate.edu/icst2008/workshops.html#st [Accessed 5 December 2013]

[86] The IBM Rational Unified Process Data Sheet. [ONLINE] Available at: ftp://ftp.software.ibm.com/software/rational/web/datasheets/RUP_DS.pdf [Accessed 5 December 2012]

[87] The V-Modell® XT (2004) [ONLINE] Available at: http://v-modell.iabg.de/v-modell-xt-html-english/index.html [Accessed 5 December 2013]

[88] van Veenendaal E.: Practical Risk-Based Testing{Product RISk MAnagement: the PRISMA method. Improve Quality Services BV (2009)

[89] von Solms S.H.: Information Security Governance–Compliance Management vs Operational Management. Computers & Security 24(6), 443-47 (2005)

[90] Wahlgren, P.: Juridisk Riskanalys: Mot En Säkrare Juridisk Metod. Stockholm: Jure (2003)

[91] Wong W.E., Y. Qi, and K. Cooper: Source code-based software risk assessing. In Proceedings of the 2005 ACM symposium on Applied computing, pp. 1485–1490. ACM (2005)

[92] Yang G. T., S. Y. Sheng, F. Y. Yuan: Research on Software Security Testing. In: World Academy of Science. Engineering and Technology 69 (2010)

[93] Zech P., M. Felderer, and R. Breu: Towards a model based security testing approach of cloud computing environments. In Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference, pages 47–56. IEEE (2012)

[94] Zech P., M. Felderer, R. Breu: Towards Risk–Driven Security Testing of Service Centric Systems. 12th International Conference on Quality Software (2012)

[95] Zech P.: Risk-based security testing in cloud computing environments. In Software Testing, Verification and Validation (ICST'11), IEEE Fourth International Conference on, pp. 411–414. IEEE (2011)

[96] Zech P.: Risk-Based Security Testing in Cloud Computing Environments. PhD Symposium at the Fourth IEEE International Conference on Software Testing, Verification and Validation (ICST), 2011 Trust Management (IFIPTM'09), pp. 215-233, Springer (2009)

[97] Zimmermann F., R. Eschbach, J. Kloos, T. Bauer: Risk-based Statistical Testing: A Refinement-based Approach to the Reliability Analysis of Safety-Critical Systems EWDC 2009. In: Proceedings of 12th European Workshop on Dependable Computing, HAL - CCSD (2009)