SINTEF A6129

REPORT

Evaluation of the 1st DIGIT field trial

Mass Lund Soldal
Emese Lujza Bogya

SINTEF ICT
Cooperative and trusted systems

April 2008

SINTEF

SINTEF ICT

Address: NO-7465 Trondheim,
NORWAY
Location: Forskningsveien 1
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

Evaluation of the 1st DIGIT field trial

AUTHOR(S)

Mass Soldal Lund, Emese Lujza Bogya

CLIENT(S)

NFR

REPORT NO. A6129	CLASSIFICATION Open	CLIENTS REF.	
CLASS. THIS PAGE Open	ISBN 9788214043891	PROJECT NO. 90B24500	NO. OF PAGES/APPENDICES 27/2
ELECTRONIC FILE CODE 080417.evaluation-report-santander-case.doc		PROJECT MANAGER (NAME, SIGN.) Ketil Stølen	CHECKED BY (NAME, SIGN.) Heidi Dahl
FILE CODE	DATE 2008-04-17	APPROVED BY (NAME, POSITION, SIGN.) Bjørn Skjellaug, Research Director	

ABSTRACT

This report evaluates the 1st DIGIT field trial, a security risk analysis conducted for Santander in the autumn of 2007. The evaluation includes lessons learned from the analysis and an empirical investigation into the use of the CORAS language in the analysis.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	ICT	IKT
GROUP 2	Risk analysis	Risikoanalyse
SELECTED BY AUTHOR	Evaluation	Evaluering

Table of contents

List of figures

List of tables

1 Introduction

This report evaluates the 1st DIGIT field trial. The field trial was a security risk analysis of the publication of content in the Santander loan application framework, specifically of the publication of offline calculators. The analysis was conducted over the course of six meetings from 10 September to 3 December 2007 as part of the DIGIT project.

DIGIT is a project financed by the Norwegian Research Council and includes several industrial partners including Santander. The security risk analysis evaluated in this report is part of Santander's contribution to DIGIT.

The evaluation consists of two parts. The first part is an informal summary of the lessons learned from the security risk analysis. The second part is an in-depth empirical evaluation of the use of the CORAS language during the workshops of the security risk analysis, based on video analysis.

1.1 Structure of the report

Section 2 provides detailed information about the security risk analysis. In Section 3 we summarise the lessons learned, and in Section 4 we present our empirical evaluation of the CORAS language. Section 5 provides concluding remarks.

2 The security risk analysis

The security risk analysis was conducted in accordance with the CORAS method. Part of this method is a process describing which meetings and workshops that should be arranged as part of the analysis, and guidelines for the organisation and arrangement of these meetings and workshops. In the following sections we first give a general introduction to the CORAS process, and then provide the details of how and when the meetings and workshops were organised in this particular analysis.

2.1 The CORAS process

The CORAS process is based on the general, high-level process for risk management standardised by [1], but is a refinement of this general process into a process tailored for the security risk analysis methods we apply, and with practical guidelines for the execution of the security analysis. This refined process can be referred to as "the seven steps of the CORAS method" and is defined in [3]. These seven steps are summarised as follows:

- **Step 1:** The first step involves an introductory meeting. The main item on the agenda for this meeting is to get the representatives of the client to present their overall goals of the analysis and the target they wish to have analysed. Hence, during this initial step the analysts will gather information based on the client's presentations and discussions.
- **Step 2:** The second step also involves a separate meeting with representatives of the client. However, this time the analysts will present *their* understanding of what they learned at the first meeting and from studying documentation that has been made available to them by the client. The second step also involves a rough, high-level security analysis. During this analysis the first threats, vulnerabilities, threat scenarios and unwanted incidents are identified. They will be used to help directing and scoping the more detailed analysis still to come.
- **Step 3:** The third step involves a more refined description of the target to be analysed, and also all assumptions and other preconditions being made. Step three is terminated once all this documentation has been approved by the client.
- **Step 4:** This step is organised as a workshop gathering people with expertise on the target of analysis. The goal is to identify as many potential unwanted incidents as possible, as well as threats, vulnerabilities and threat scenarios.
- **Step 5:** The fifth step is also organised as a workshop. This time with focus on estimating consequences and likelihood values for each of the identified unwanted incidents.
- **Step 6:** This step involves giving the client the first overall risk picture. This will typically trigger some adjustments and corrections.
- **Step 7:** The last step is devoted to treatment identification. This step is best organised as a workshop.

2.2 Meetings and participants

The security risk analysis conducted for Santander consisted of six meetings between the analysis team and representatives of Santander.

Table 1 summarises the meetings of the security risk analysis. The table gives the date of each meeting, a summary of the activities for each meeting, and a reference to which steps in the CORAS process they represent.

Meetings	Date	Activities	Steps
Meeting 0	5 Sept 2007	• Planning	
Meeting 1	10 Sept 2007	• Presentation of target	Step 1
Meeting 2	25 Sept 2007	• Presentation of target description • Asset identification • High-level analysis	Step 2
Meeting 3	16 Oct 2007	• Approval of target description • Definition of risk evaluation criteria	Step 3
Meeting 4	2 Nov 2007	Workshop 1: • Identification of threats, vulnerabilities and unwanted incidents	Step 4
Meeting 5	21 Nov 2007	Workshop 2: • Likelihood and consequence estimation	Step 5
Meeting 6	3 Dec 2007	Workshop 3: • Risk evaluation • Treatment identification	Steps 6 & 7

Table 1: Meetings of the security risk analysis

SINTEF was responsible for the administration of the analysis. Representatives from Santander contributed as domain experts. The SINTEF analysis team and the Santander participants are presented in the tables below.

			Meeting					
Role	**Name**	**Company**	**1**	**2**	**3**	**4**	**5**	**6**
Project leader	Ketil Stølen	SINTEF	X	X	X			
Analysis leader	Heidi Dahl	SINTEF	X	X	X	X	X	X
Analysis secretary	Mass Soldal Lund	SINTEF	X	X	X	X	X	X
Analyst	Aida Omerovic	SINTEF	X	X	X	X	X	X
Analyst	Emese Lujza Bogya	SINTEF				X	X	X

Table 2: Security risk analysis team

			Meeting					
Role	**Name**	**Company**	**1**	**2**	**3**	**4**	**5**	**6**
Coordinator	Tor Gaute Indstøy	Santander	X	X	X	X	X	X
Target owner	Anne Karine Walfjord	Santander	X		X	X	X	X
System operator	Øystein Andersen	Santander	X	X	X	X		
Security expert	Tarje Milde	Santander	X		X			
IT architect	Geir Berglind	Santander	X		X			

Table 3: Santander team

2.3 Deliverable

The deliverable of the security risk analysis was a security risk analysis report documenting risks, their risk levels and recommended treatments. The report had a total of 69 pages and two appendices. The quality assurance of the report was conducted in accordance to standard SINTEF practice. The report was delivered to Santander on 6 February 2008 pr mail by the analysis leader.

3 Lessons learned

This section presents lessons learned from the analysis. The analysis was carried out following the CORAS method as documented in [3] (see Section 2). This presentation follows basically the same structure. In Section 3.1 we look at the system and target description, in Section 3.2 the asset identification, in Section 3.3 the likelihood and consequence scales, and risk matrices, in Section 3.4 the identification of threats, vulnerabilities and unwanted incidents, in Section 3.5 the likelihood and consequence estimation, in Section 3.6 the risk evaluation and treatment identification, and finally in Section 3.7 writing of the report and other supplementary work.

3.1 System and target description

Santander provided a system description in the form of rich pictures and use cases. This system description held high quality and provided a good starting point for our formalisation of the target description. It was, however, sometimes difficult distinguish the "as is" system description from the "to be" system description, and this led to some misunderstandings and need for clarification during the discussions about scope and target. In addition to Santander's presentation of the system description, the analysis team was invited to participate in a demonstration, outside of the standard meetings and workshops of the analysis, of part of the system to be analysed. This demonstration proved very valuable, as it gave the analysis team a deeper understanding of the target of analysis.

The analysis team formalised the system description in UML using a UML template for MS Visio. The target description used the same diagrams with a colour coding (black for outside target and red for inside target). The final target document had the form of a MS PowerPoint presentation. While the use of MS Visio and MS PowerPoint works excellently for presentations of system and target descriptions, it is unfortunate if we view the system description as an artefact with a value of itself. For the customer the possibility of using the system description for other purposes, and hence its value, would increase if it had been formalised in a specialised UML tool.

The formalisation of the system and target descriptions made to some degree non-standard use of UML diagrams, such as using package diagrams for describing the architecture of the system. This was unfortunate as it led to some confusion with respect to the meaning of the diagrams. The use of colour coding in order to define the target gives a feeling of being somewhat *ad hoc*. In the light of this it could be useful if the guidelines for making system and target descriptions were elaborated and made more detailed.

Summary

- We should in future security risk analyses state explicitly at the start of the analysis whether it is an analysis of the system "as is" or the system "to be".
- Demonstrations of the target system are useful for the analysis team and should be recommended.
- We should consider using a specialised UML tool for formalising the system and target descriptions.
- Elaboration of the guidelines for making system and target descriptions would be useful.

3.2 Asset identification

During the asset identification there was some discussion related to the ranking of assets. The reason for this was that while all the participants from Santander viewed themselves as stakeholders, they belonged to different parts of the enterprise – with different responsibilities, concerns and interests. This was resolved when we were able to establish a common interpretation of how we understood the stakeholder/party "Santander".

The distinction between direct and indirect assets was of great use during the asset identification; it helped us focus the analysis on the more "concrete" assets. This also helped us make a kind of compromise between the concerns and interests of Santander as an enterprise and the concerns and interests of the participants as Santander employees.

Summary

- When defining the stakeholder/party of the analysis it is important to establish a common interpretation of who the stakeholder/party is and his/hers/its/their concerns and interests.
- The distinction between direct and indirect assets provides a meaningful way of classifying assets, and may help focusing the analysis on more concrete aspects without compromising concerns at enterprise level.

3.3 Likelihood and consequence scales and risk matrices

The definition of a likelihood scale and most consequence scales was relatively smooth. However, in the definition of one of the consequence scales we had trouble defining how the consequences should be expressed, and how the different severities should be distinguished from each other. We were not able to resolve this on the meeting where likelihood and consequence scales were to be defined, with the result that we had to use time on the next meeting to make the final definitions.

Definition of the most of the risk evaluation matrices was unproblematic. However, one of the matrices ended up as entirely red, i.e. with all combinations of consequence and likelihood defined as unacceptable. It is a question whether this is problem or not, but it may indicate that we were not able to define a consequence scale that distinguished sufficiently between the severities of different events. This also raises the question whether we should always use the same likelihood scale for all assets, or if we should allow for more fine grained likelihood scales when we have problems defining consequence scales that can distinguish between acceptable and unacceptable events.

Summary

- A more extensive library of "reusable" consequence scales could be useful.
- The practice of using the same likelihood scale for all assets should be reassessed.

3.4 Identification of threats, vulnerabilities and unwanted incidents

The identification of threats, vulnerabilities and unwanted incidents was carried out according to the guidelines, and with on-the-fly modelling of threat diagrams projected on a whiteboard. This approach to identification of threats, vulnerabilities and unwanted incidents worked well and was to a large degree successful. The only criticism of these sessions is that the SINTEF analysis team could have been more challenging towards the other participants when discussion was slow.

The CORAS editor was used by the analysis secretary for the on-the-fly modelling. The editor works satisfactory in the sense that it is possible to use it for this purpose without major problems or loss of data. However, the editor still has many bugs and known issues which makes it more difficult to use than should have been necessary. Further, it is difficult to hide the problems of the editor from the customer during on-the-fly modelling. This may give an impression of lack of professionalism, or lack of maturity of the CORAS method, event though this is not necessarily the case (i.e. it can be somewhat embarrassing to use the editor in front of people).

Summary

- On-the-fly modelling during security risk analysis workshops seems to be a good means for identifying and structuring vulnerabilities, threat scenarios and unwanted incidents.

- There are still substantial problems with the CORAS diagram editor. Effort should be invested to improve the editor, especially with respect to compliance with the textual syntax.

3.5 Likelihood and consequence estimation

Likelihood and consequence estimation was carried out by walk-through of the threat diagrams. This was mainly unproblematic, but the problem of the undecided consequence scale re-emerged, and made consequence estimation for the relevant asset less efficient than it could have been.

The likelihood and consequence estimates were documented by annotating the threat diagrams using the CORAS diagram editor. This works ok, but the CORAS diagram editor lacks explicit support for annotating threat scenarios and unwanted incidents with likelihood estimates. This means likelihood estimates must be entered as text together with the description of the threat scenarios or unwanted incidents, which makes the use of the editor unnecessarily tedious.

Summary

- It is important to have the likelihood and consequence scales thoroughly established before starting the likelihood and consequence estimation.
- Explicit support for annotating threat diagrams with likelihood estimates would improve the CORAS language editor.

3.6 Risk evaluation and treatment identification

Risk evaluation was carried out off-line by the analysis leader based on the likelihood and consequence estimates. The risk evaluation was documented in risk diagrams and risk matrices. Treatments were identified in a treatment workshop and documented in treatment overview diagram. Unfortunately, a bug in the CORAS diagram editor prevents us from making risk and treatment overview diagrams with the editor, so these diagrams had to be prepared using MS Visio. Further, we have no support for automatically making risk evaluation matrices, something that could have been useful.

Summary

- The CORAS language editor should be fixed so it can be used for making risk diagrams.
- Support in the CORAS language editor for generating risk evaluation matrices could be useful.

3.7 Finishing of the report and other supplementary work

We decided not to use the report generation functionality of the CORAS tool, but to prepare the final report with MS Word. With the exception of risk and treatment overview diagrams (see Section 3.6), the report uses diagrams exported from the CORAS language editor. This saves the work of redrawing the diagrams, e.g. with MS Visio, but making changes to the diagrams of the report becomes more tedious.

As a result of the work with the report, a template for security risk analysis reports written in MS Word was made. Each meeting of the security risk analysis stared with a short introductory presentation, and also these MS PowerPoint presentations have been converted into templates for future use. Both the report template and the presentation templates have been applied in a later analysis, and proved to save efforts.

Summary

- Combined use of the CORAS language editor and MS Word for producing security risk analysis reports works fine, but including diagrams for the editor in the written report is tedious work.

- Templates can reduce the work of future analyses, and further work on improving the templates should be considered.

4 Evaluation of the CORAS language

In connection with the security risk analysis trial we made an evaluation of some aspects of the CORAS method, namely the CORAS language. The evaluation is based on a methodology for case studies.

A common conception is that graphical modelling languages can increase the understanding and advance the communication between stakeholders in technical matters related to computerized systems. In this evaluation we aim to investigate whether the CORAS language – a graphical language for modelling of security risk information – may have this effect in a security risk analysis.

We make a first formulatiōn of our overall hypothesis for the evaluation as follows:

> "The CORAS language contributes to the understanding of risks in a security risk analysis, and contributes to communication between participants of a security risk analysis."

In the following, this formulation is refined into research questions that we aimed at answering in the case study that the trial constituted. Section 4.1 gives background on the CORAS language, Section 4.2 presents our case study design and Section 4.3 provides the results of the evaluation.

4.1 Background on the CORAS language

The CORAS language is a graphical modelling language to be used in security risk analyses. The language is used as part of the CORAS method, which provides a methodology – including a process and a number of guidelines, as well as the language – for conducting security risk analysis of computerized systems. The most comprehensive treatment of the CORAS method is found in [2], which includes material from other publications on the method, such as [3,5]. The CORAS method is based on the use of structured brainstorming. In these brainstorming sessions, the CORAS language is applied for making models of threat scenarios and risks on the fly. The way in which the models are constructed, as specified in the guidelines, structures the brainstorming sessions. These models are also important contributions to the documentation of the security risk analysis results, i.e. the security risk analysis report.

The first version of the language appeared in 2002 as a UML profile, but has since then been developed into a specialized language (domain specific language) through several iterations with feedback from trials, teaching and experiments [10,11]. The latest version of the language, which was the version that is applied in the 1st DIGIT field trial, is documented in [5].

4.1.1 Claims about the CORAS language

In the following we have a look at what should be considered "intended use" of the CORAS language, and at the results of earlier evaluations of the language.

Early publications on the language (e.g. [12]) usually state the objectives of the language as:

- To describe the target of evaluation at the right level of abstraction.
- To facilitate communication and interaction between different groups of stakeholders involved in a security risk analysis.
- To document security risk analysis results and the assumptions on which these results depend to support reuse and maintenance.

In [6], the success criteria of the language are stated as:

"[A] language for describing security risks, that:

a. is suitable for use in structured brainstorming sessions
b. is easily understandable for the participants in the brainstorming, including those who receive the analysis results afterwards
c. has a precise syntax, meaning its design should be based on:
  i. best practice within information visualization
  ii. experiences with realistic security risk scenarios
  iii. users' preferences
  iv. existing risk modeling techniques
d. has a structured semantics that translates arbitrary diagrams into English
e. supports and documents the different steps in the security analysis process"

In [5], which focuses on the semantics of the language, three claims regarding "our graphical approach to security risk modelling" are made. The claims are that the CORAS language contributes to solving the issues of:

- How to facilitate communication in a group consisting of people with different backgrounds and competences.
- How to estimate the likelihoods and consequences of identified risks.
- How to document the security analysis in a comprehensible manner.

In the setting of this evaluation, where the use of the language is in focus, the first bullet is the most relevant. In the document, this claim is further elaborated:

> "Our aim has been to provide the participants with a means of communication that covers both technical and more high-level information, without being too complicated to understand. Offering a common basis for communication will hopefully reduce misunderstandings and thereby give a more correct risk picture."

These formulations form the basis when we formulate our research questions with respect to the intended use of the CORAS language.

4.1.2 Earlier evaluations of the CORAS language

Evaluations of earlier versions of the language include an empirical study of the terminology on which the language is based [9] and an empirical study of the use of graphical icons in the language [8]. The former concluded that the terminology of the language to a large degree is intuitively understandable, while the latter concluded that using graphical icons increased the readability (measured in reading speed) of diagrams made with the language. A qualitative evaluation of the language [7] concluded that in most cases it has the necessary expressiveness for modelling security risk analysis results, but that parts of the language was superfluous and that some situations could not be modelled satisfactory. All these studies have led to changes in the language where the identified shortcomings have been mitigated [10].

4.2 Case study design

We base our case study design for evaluation of the CORAS language on a methodology for case studies describe by Robert K. Yin [14].

According to Yin a case study design should include:

1. research questions
2. proposition

3. characterization of the unit(s) of the evaluation
4. the logic linking the data to the proposition
5. the criteria for interpreting the findings

In the following we characterize the design of the case study. While we do not follow the points of Yin directly, our case study design does cover all of the points.

4.2.1 Setup of the case

The 1st DIGIT field trial is a real security risk analysis, conducted for the Norwegian branch of Santander. Santander in Norway has responsibility for the online banking systems of Santander in the Nordic countries. The target of analysis was a system for car loan applications in Finland. The system is used by car dealers to apply for car loans on behalf of their constomers. The front-end of the system consists of a framework, running locally at the car dealers' computers, and a loan calculator running in the framework. The framework communicates with Santander's servers, and part of the target is also the method used for publishing updates of the loan calculator on the server from which they are downloaded by the framework.

A security risk analysis consists of several meetings. In this security risk analysis, the meetings followed the structure recommended by the CORAS guidelines, documented in [3]. The security risk analysis is lead by an analysis team. The analysis team always consists of an analysis leader and an analysis secretary. In this analysis, the team also had an additional member who took part in the discussions at the meetings, and an evaluator who observed the meetings.

In addition to the analysis team, the security risk analysis had participants from Santander. These participants are what Yin calls the units of the evaluation. In this particular security risk analysis they were IT professionals – IT architects, IT security experts, system developers and IT support personnel.

At the beginning of each security risk analysis meeting, the participants were given a brief introduction by the risk analysis secretary, presenting the risk analysis concepts and the elements of the CORAS language that were relevant for the discussions on that meeting. These presentations followed the definitions from [2] and the semantics of the language as defined in [5]. An example of the introduction the participants were given at the risk analysis meeting is shown in Appendix A.

4.2.2 Research question

Based on what has been said earlier about the desired properties of the CORAS language, we can reformulate our overall hypothesis to the following:

> "The participants of the security risk analysis trial use and understand the CORAS language as intended (i.e. in accordance to definitions of concepts and the semantics) regardless of background and with a minimum of training"

Yin recommends that the phrasing of the research questions should guide the choice of research strategy. However, he also says that when you have a predisposition towards a specific research strategy, as in our case, you should formulate your research questions to best match the strategy. For case studies Yin recommends "how" and "why" questions as the form of the research questions, which means following the interpretive school. Based on this recommendation and the background information about the CORAS language, we formulate the following overall research question:

> "How is the understanding of the language and the use of its concepts among the participants in the security risk analysis trial?"

This can again be made more specific by the following sub-questions:

- "Do the participants understand the language as intended?"
- "Do the participants use the language as intended?"
- "Do the participants' background influence how they use the concepts and understand the language, and in case how?"
- "Do misunderstandings come to the surface when the diagrams are used (drawn), or when the diagrams are to be understood (read)?"
- "Are there certain concepts that are more easy/faster understood, than others? Do some particular misunderstandings occur more than once?"

By "understand the language" we mean being able to interpret diagrams made in the language in accordance with the structured English semantics defined in [5]. This is of course difficult to investigate directly. We must therefore resort to an investigation of the *use* of the language.

When we continue to formulate propositions, we therefore make propositions that are related to the use of both concepts and the language.

4.2.3 Propositions

The difference between research questions and propositions is that research questions state what we are interested in investigating, while propositions are statements more directly related to what we are able to study.

Based on the overall research question we can formulate an overall proposition for the evaluation:

> "After a short introduction on each meeting, the participants of the trial show a use of the language and its concepts that indicate that they understand the language and its concepts as intended."

Again, we can define a number of sub-propositions that make the statement more specified. In these propositions we try to state what kind of use we are looking for as indications of understanding.

- "The duration of the discussions related to the meaning of the language and its concepts, beyond the ones during the presentation of the concepts, is negligible."
- "There are few misunderstandings in the discussion at the meetings originating from the use of the language and its concepts."
- "The participants of the meetings seldom have to ask the analysis leader about what concepts, language elements or diagrams mean."

4.2.4 Data collection

During the security risk analysis meetings we collected data that we use to investigate the validity of the propositions formulated above. The main method of data collection was observation of the meetings. These observations were made in two ways:

- The evaluator observed the meetings and made notes of the occurrence of specific events during meeting.
- The meetings were recorded on video. The video from the meetings were analysed after the security risk analysis using the video analysis software Observer [13] from Noldus.

The subjects did mainly talk during the analysis, and the data collection therefore focused on their actual speech rather than, for example, their body language. Both the observations by the evaluator and the video analysis focused on the occurrence of specific events during the meeting. These events are defined such as to be indicators of how the language and its concepts are used during the meetings. An event is characterized by its duration, prelude (reason) and content. When similar events occur more than once, we can find its frequency and possibly establish patterns.

The evaluator focused on observing "negative" indicators – i.e. events that indicate that there are misunderstandings or participants have problems understanding the use of concepts or the language. These indicator events were taken down on a form by the evaluator during the meetings and categorized into event classes. The evaluator form and the event classes are shown in Appendix B. These forms formed the basis for development of a coding scheme used in the video analysis (see Section 4.2.6).

4.2.5 Interpretation/analysis of the data

When we analyse and interpret the data it is important that we establish a relation between the analysis and the validity of the propositions. If the propositions are true, we expect the occurrence of "negative indicators" to be few and short; and relatively stable throughout the trial (because a large decrease could indicate that the experience gained though participation is more important than the intuitive understanding).

In addition to count the number of occurrence of these indicator events, we have also used the video analysis to find out what situations cause the events. Another part of the analysis was to find out exactly which concepts were misunderstood, and which were understood. If the propositions are true, the misunderstandings should not occur because of the reading of the diagrams. In other words, as long as the reason for the misunderstanding was not about the reading of the diagram, the hypothesis will be valid.

We also tried to see if we could establish patterns by finding resemblances in the events, and maybe specific orders of events. If there are not any patterns to be found in the occurrence of events, (but we are facing different specific events) our hypothesis will be strengthened.

4.2.6 Coding scheme

The basics of using the video analysis tool Observer is to code (or tag) the video with a predefined coding scheme [13]. A thorough analysis of the video would make it possible to code every single action of the observed persons, but this would be an enormous task and not all data would be relevant to our research. The question is therefore which specific events we are looking for? We have to specify the events we should be looking for and want to code.

Because the recorded meetings might show events we had not anticipated before we stared the analysis, we did a repetitive coding. This made it possible to apply any new knowledge gained in the coding process to our research. This means we had to stay open for new ways of analysing the video and to change the coding scheme accordingly.

In the following we present the considerations made in the creation of the coding scheme.

- **Actions/verbal interaction.** Our main focus of coding was on the communication between the participants. Even though body language of course is an important part of the communication process, this is not a part of the coding.

- **Subjects.** The participants of the security risk analysis meetings can be divided in two groups: the analysts and the clients. We focused mainly on the client's actions.
- **Questions.** We have not only coded the use of the CORAS language, but also wherever the subjects have questions about it. Identifying the relevant questions can of course be a delicate matter since the questions may not be directly related to the CORAS language or its semantics.
- **Misunderstandings.** What qualifies as a misunderstanding can be put up for discussion. We differentiate misunderstanding from question by saying that when there is a question, the subject knows that (s)he does not understand, while in a misunderstanding it is the analyst who notices that something in not clear. We have concentrated on the misunderstandings with respect to the semantics of the CORAS language.
- **Patterns.** In order to find out if there is any pattern in the misunderstandings we also coded the circumstances of the events – the situation the subject found themselves in while the misunderstanding occurred. The two obvious situations are reading a diagram and drawing (indirectly) it.

We used the video analysis software Observer to code our videos. As stated earlier, the data collection was conducted in an iterative way, open for improvements of the coding shceme. From what we learned during the actual coding, the following considerations were made.

- **No starting/ending points**. The total time of the videos was 796 minutes (around 13 hours). We soon noticed that the misunderstandings had so short durations that there was no point in coding their starting and ending points.
- **"Remarks".** As mentioned earlier, the topics of the discussions could not always be clearly divided into the two groups "about the CORAS language" and "not about the CORAS language." Some statements were, for instance, indirectly about conducting security risk analysis using CORAS. We therefore found it interesting to code certain remarks, expressed by the clients. The problem here is that they cannot be divided into the two clear groups of positive and negative statements about the CORAS language.
- **Correct use, wrong word.** We noticed that many times the semantics of the CORAS language was understood correctly but the clients did not use the correct, specific expression. We therefore found it interesting to have a coding for these situations, i.e. situations the client has a clear difficulty in coming up with the correct word.

The resulting coding scheme used in the video analysis is presented in Table 4 below.

Question asked by the client about:	The **semantics** of CORAS
	One of the **concepts**
Misunderstanding that occurs while:	**Reading** a diagram
	Drawing a diagram
	Talking about a diagram
Understanding:	Using a **specific word** (expression)
	Using an **unspecific word**
Remark (anything else concerning CORAS)	

Table 4: The final coding scheme

4.3 Results

The videos are recordings of almost the whole security risk analysis, while the CORAS language is not used in all the meetings [3]. It is therefore not surprising that there are no coded events for some parts. The coded events are summarised in Table 5. As we can see, half of the events are from meeting 4.

		All meetings		Meeting 2	Meeting 3	Meeting 4	Meeting 5	Meeting 6
Question	About the semantics	4	2	1	0	1	0	0
	About the concept		2	0	1	1	0	0
Mis-understanding	While reading	7	1	0	0	1	0	0
	While drawing		4	0	0	4	0	0
	While talking		2	1	1	0	0	0
Understanding	Specific word	13	9	1	0	5	3	0
	Unspecific word		4	0	0	2	2	0
Remark		6	6	2	0	2	2	0
Sum		30	30	5	2	16	7	0

Table 5: Overview of the coding

The CORAS language is mainly used when the threat diagrams are drawn, and it is therefore not surprising that most coded events occurred during meeting 4 (Risk identification). We will therefore first look at all the data, and thereafter look more closely on the events of the fourth meeting.

4.3.1 Evaluation of propositions

In Section 4.2.3 we stated the following propositions:

- "The duration of the discussions related to the meaning of the language and its concepts, beyond the ones during the presentation of the concepts, is negligible."
- "There are few misunderstandings in the discussion at the meetings originating from the use of the language and its concepts."
- "The participants of the meetings seldom have to ask the analysis leader about what concepts, language elements or diagrams mean."

In the following we look at these propositions in the light of the analysis of the video of the meetings.

4.3.1.1 Short time

As already mentioned, the events were so short that there was no point of coding their starting and ending points. The events were composed of one to two sentences from both the client and the analyst. In some cases the clients had already found the answer themselves right after they had articulated the question.

4.3.1.2 Few cases

The CORAS language was used in three of the meetings of the security risk analysis with a total duration of 575 minutes (9.5 hours). During this period there were 4 questions and 7 misunderstandings (see Table 6). This means that the frequency of misunderstandings and questions is 1 per 52 minute, and the frequency of misunderstandings only is 1 per 82 minute.

Question	About the semantics	4	2
	About the concept		2
Misunderstanding	While reading	7	1
	While drawing		4
	While talking		2

Table 6: Overview of the coded questions and misunderstandings in total

4.3.1.3 Stable frequency

In Figure 1 we see the coded events marked against the timeline for the parts of the security risk analysis when the CORAS language was used. We can see quite clearly that the frequency of neither misunderstandings nor questions is stable.

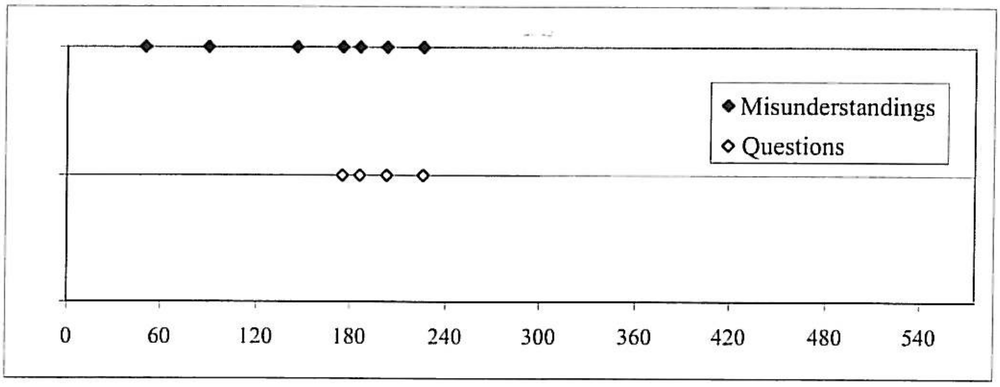


Figure 1: Questions and misunderstandings

4.3.1.4 No patterns, no repetitions

The division of misunderstandings in their sub-categories is shown in Table 7. The misunderstandings occurred mainly when the clients were asked to draw the diagrams (indirectly in the sense that they were telling the analysts what to write and draw.)

Misunderstanding	**While reading**	1
	While drawing	4
	While talking	2

Table 7: Overview of misunderstandings

4.3.2 Analysis by expression

When analysing the understanding and use of icons by the clients, one must keep in mind how much time and experience the client had with the icons. We can then note two interesting situations:

- The clients face many new icons when the threat diagrams are drawn.
- The clients are meeting only one or two new icons when the asset diagrams and the treatment diagrams are drawn.

	Diagrams				
	Asset overview	Threat	Risk	Treatment overview	Treatment
Asset	X	X	X	X	X
Party	X				
Threat		X	X	X	X
Vulnerability		X			X
Threat scenario		X			X
Unwanted incident		X			X
Risk			X	X	X
Treatment scenario				X	X

Table 8: Use of icons in each diagram

The use of the icons in the different diagram types is shown in Table 8. All the icons and expressions were used during the security risk analysis. In the following we look at the different events coded for each icon and expression.

Some icons do not have any coded events, but this does not mean that the icon was not used during the meetings. When something is intuitive, we take it for granted, and therefore we do not mention it. It is only when something is special or unusual that we mention it. We may therefore conclude that those icons and concepts which were the most intuitive for the clients are those that are did not create special events that could have been coded. Three concepts do not appear in our coded events. These absences have different reasons:

- **Risk.** During a security *risk* analysis, the word "risk" is used all the time, one may say it is an "overused word": risk estimation, exposed to risk, risk matrix, etc. We may therefore assume that the expression was familiar to the clients.
- **Party.** The party icon is one of the first ones to be introduced, when there are still few icons, and it is used only in the beginning of the analysis process. Since it is representing the company of the client, the identification is easy.
- **Treatment scenario.** The treatment icon, contrary to the party icon, is one of the lasts ones to be introduced, when all the other icons have already been introduced and are in use.

In the following sub-sections we summarise the coding of the remaining concepts.

4.3.2.1 Asset

The semantics: There are no positive or negative events concerning the semantics of asset. It leads us to believe that the icon was intuitive.

The concept: The first reaction of the client towards the word "asset" was question about its definition. The concept of asset seemed to be too wide.

Meeting	Coding		Event
3	Question	Concept	"What do you put in the word assets?"
	Mis.	Talk	"To lose concession, is that an asset?"
4	Underst.	Spes. word	"Asset, how often do you use that in everyday life?"

Table 9: Events related to the concept of asset

4.3.2.2 Vulnerability

The semantics: The semantics of vulnerabilities was not mentioned during the analysis. This indicates that the icon was intuitive.

The concept: All the four events are evidence confirming that the concept of vulnerability was understood. The use of “vulnerability” throughout the analysis, tells us that the concept was familiar to the clients.

Meeting	Coding		Event
4	Underst.	Spes. word	“The vulnerability is if we make a copy without encrypting it. “
			“That it is not scalable, yes that’s one of our vulnerabilities.”
5			“The vulnerability is that one makes systems that often change.”
			“How we configure it; that is our vulnerability.”

Table 10: Events related to the concept of vulnerability

4.3.2.3 Threat

The semantics: The semantic difference between deliberate and accidental threats seems to be clear. The difference between human and non-human threats, however, needed some clarification.

Meeting	Coding		Event
2	Question	Semantic	First reaction to the threat icons: “I see those threat buddies: there is one that is mean, one that is good – and there is a flag?”

Table 11: Events related to the semantics of threat

The concept: As we can see from Table 12, the clients had a hard time with the idea of deliberate vs. accidental threats; instead they were fixed with the idea of authorised vs. unauthorised person. Instead of “deliberate” and “accidental”, they used expressions like “person with goodwill”, “mean intention”.

When discussing threats, clients already think of the consequences. Clients manage to find consequences, while not always identifying the threat.

Meeting	Coding		Event
2	Mis	Talking	“What is important here, the threat or the usability of the system?”
4	Underst.	Unspecific word.	“That the data can be stolen; it depends if it is with intent or not.”
	Question	Concept	“Is not a deliberate threat also an authorised person?”
	Underst.	Specific word	“…a person that does it with mean intention…”
	Mis.	Reading	“Why is the unauthorised person a threat? Oh, yes because he is stealing.”
		Drawing	“The threat is that the clients use the competitor’s because ours is not available”
			Looking for deliberate threats: “Others can be internal people like me who accidentally take down the server.”
5	Underst.	Unspecific work	Pointing: “That person…”
	Remark		“An unauthorised person does not have to be some strange man, but it can be one of our consultants.”

Table 12: Events related to the concept of threat

4.3.2.4 Unwanted incident

<u>The semantics:</u> Since the unwanted incident is introduced at the same time as the threat scenario, one of the clients pointed out the unwanted incident icon’s similarity to that of the threat scenario. It only took 5 minutes for the client to answer his own question, about their differences.

Meeting	Coding		Event
4	Question	Semantics.	What is the difference between the box with the triangle and the one with a star?
	Underst.	Unspecific word	“Now I get it – the star, it is exploding. And the triangle is the warning.”

Table 13: Events related to the semantics of unwanted incident

<u>The concept:</u> Unwanted incident is mentioned throughout the security risk analysis. The clients have no problems in understanding what the concept is.

Meeting	Coding		Event
2	Underst.	Specific word	“An unwanted incident is for instance that we get 10 applications a day.”
5			“The unwanted incident would be that someone manages to pick up the data.”

Table 14: Events related to the concept of unwanted incident

4.3.2.5 Relations

<u>The semantics:</u> The clients understood that an arrow symbolises some relation between two icons.

<u>The concept:</u> The kind of relation an arrow stands for seemed to be less obvious. They therefore used only the word “arrow” to describe them.

Meeting	Coding		Event
4	Mis.	Drawing	“We should have an arrow from here to there…”
			Proposing to have an arrow directly from vulnerability to asset.
5	Underst.	Unspecific word	“There is no arrow there…”

Table 15: Events related to the concept of relation

4.3.2.6 Frequency

The semantics: There were no problems concerning the semantics of frequency.

The concept: The client’s remarks about the use of frequencies in a security risk analysis may show how well CORAS manages to motivate the use of frequencies.

Meeting	Coding	Event
5	Remark	“Why do we have to put frequencies at all?”

Table 16: Events related to the concept of frequency

4.3.2.7 Summary

Table 17 summarises the above analysis of the intuitiveness the CORAS language, and states in what way we can say that the concepts and language elements are intuitive and in what way we can say that they are not intuitive.

	Intuitive	Not intuitive
Asset	The semantics is understood.	The concept is a bit wide. Unusual expression.
Party	*Not coded*	*Not coded*
Human threat	The semantics clearly shows the “human aspect” of the threat.	To identify that the difference between them is the intention.
Non human threat		Unclear semantics about the kind of the threat.
Vulnerability	Familiar expression.	
Threat scenario	There is enough time to get familiar with it.	Resemble each other. The client is introduced to four new icons at once, including these two.
Unwanted incident	Familiar expression and understood concept.	
Risk	*Not coded*	*Not coded*
Treatment scenario	*Not coded*	*Not coded*

Table 17: Summary table of the intuitiveness of the icons

5 Conclusions

This report evaluates the 1st DIGIT field trial, a security risk analysis conducted for Santander in the autumn of 2007. The evaluation has two parts; a summary of the lessons learned from the analysis and an empirical investigation into the use of the CORAS language in the security risk analysis meetings. The more informal lessons learned evaluation is similar to the informal evaluation we have made of earlier security risk analysis trials in projects like CORAS and SECURIS, while the empirical evaluation is a first attempt at making our evaluations more formal and scientific.

The main lessons learned can be summarised as follows:

- The CORAS method in its current form is starting to stabilise. By following the guidelines of the CORAS method strictly, we were able to conduct a successful security risk analysis.
- Though the guidelines are starting to stabilise, there seems to be a potential for improvement with respect to:
    - Formalising the system and target specifications.
    - Definition of likelihood and consequence scales.
- The tool support of the CORAS method is still somewhat weak, and the CORAS diagram editor has still a large potential for improvements.

The empirical evaluation, without being conclusive, gives us some indications with respect to the usability of the CORAS language:

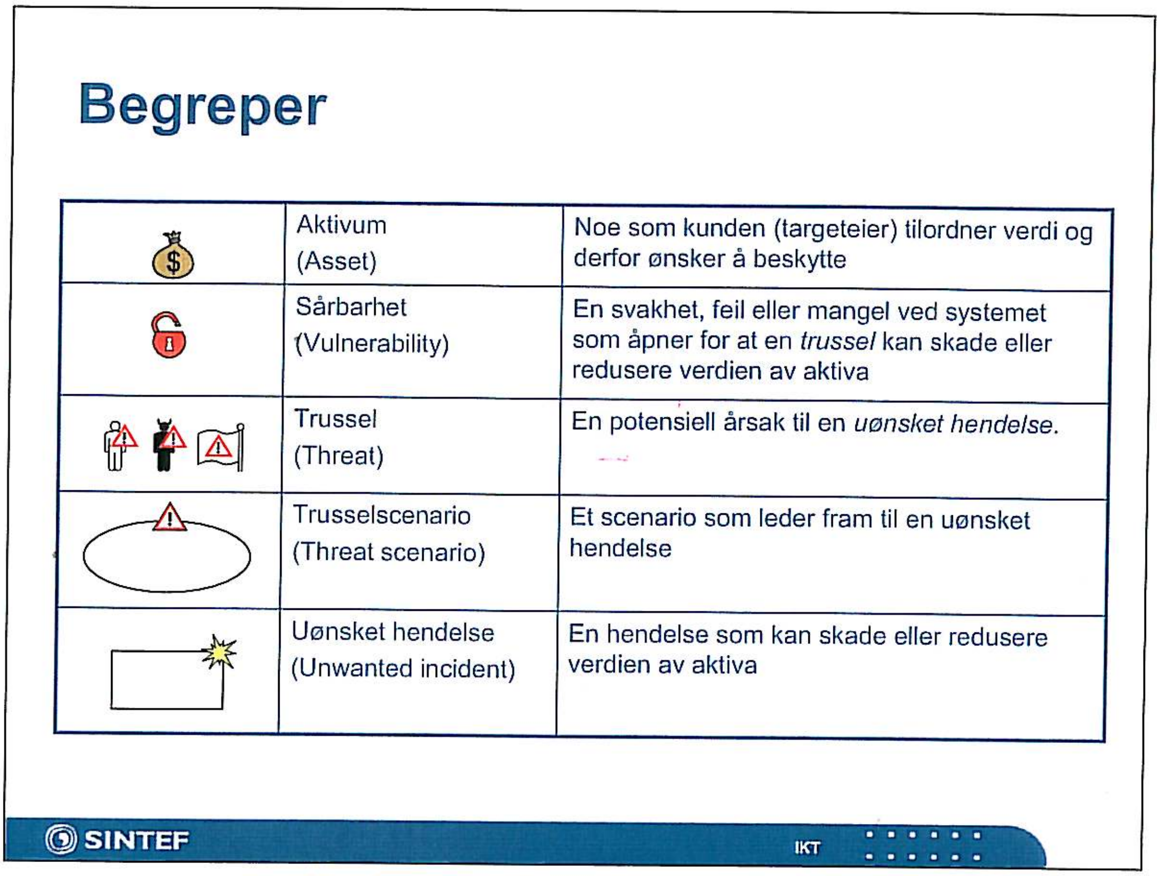
- The "negative events", i.e. misunderstandings and questions, with respect to the language were relatively few. This indicates that the language is relatively intuitive for the group of IT professionals that participated in the security risk analysis.
- The frequency of these "negative events" seems to be higher when new concepts and language elements are introduced, which indicates that there is a learning curve for the participants.
- The time spent on "resolving negative events" is negligible, which indicates that even though there is a learning curve for the use of the CORAS language, this is not a major problem.
- An interpretive analysis of events during the security risk analysis indicates that most of the concepts and language elements of the CORAS language is intuitively understandable for the IT professionals in the analysis. The two notable exceptions are the categorisation of threats into "human deliberate", "human accidental" and "non-human", and the semantics of, and difference between, the relations that are represented as arrows in the diagrams.

It should be noted that the data of this evaluation can be seen as somewhat sparse, and that for this reason it can be difficult to draw strong conclusions. It should also be noted that this is the first time we have carried out this particular kind of evaluation/case study, and that the learning curve has been steep for the evaluators. We hope and believe that our new knowledge and insights on case studies can help us conduct more conclusive empirical evaluations in the future.

References

1. AS/NZS 4360:2004. Australian/New Zealand Standard for Risk Management, 2004.
2. Folker den Braber, Gyrd Brændeland, Heidi E. I. Dahl, Iselin Engan, Ida Hogganvik, Mass Soldal Lund, Bjørnar Solhaug, Ketil Stølen, Fredrik Vraalsen. The CORAS model-based method for security analysis. SINTEF, 2007.
3. Folker den Braber, Ida Hogganvik, Mass Soldal Lund, Ketil Stølen, Fredrik Vraalsen. Model-based security analysis in seven steps — a guided tour to the CORAS method. BT Technology Journal, 25(1):101-117, January 2007.
4. CORAS tool http://coras.sourceforge.net/ (retrieved November 2007)
5. Heidi E. I. Dahl, Ida Hogganvik, and Ketil Stølen. Structured semantics for the CORAS security risk modelling language. Technical Report A970, SINTEF ICT, 2007.
6. Ida Hogganvik. A graphical approach to security risk analysis. PhD thesis, Faculty of Mathematics and Natural Sciences University of Oslo, 2007.
7. Ida Hogganvik, Mass Soldal Lund, Ketil Stølen. Quality evaluation of the CORAS UML profile. Technical report A2199, SINTEF ICT, 2007.
8. Ida Hogganvik, Ketil Stølen. On the Comprehension of Security Risk Scenarios. In Proc. 13th International Workshop on Program Comprehension (IWPC 2005), pages 115-124, IEEE Computer Society, 2005.
9. Ida Hogganvik, Ketil Stølen. Risk analysis terminology for IT-systems: Does it match intuition? In Proc. 4th International Symposium on Empirical Software Engineering (ISESE 2005), pages 13-23, ISBN: 0-7803-9508-5, IEEE Computer Society, 2005.
10. Ida Hogganvik, Ketil Stølen. A Graphical Approach to Risk Identification, Motivated by Empirical Investigations. In 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2006), number 4199 in Lecture Notes in Computer Science, pages 574-588, Springer, 2006.
11. Mass Soldal Lund. The CORAS Language – History and status. Working note dated 2006-03-07, SINTEF ICT, 2006.
12. Mass Soldal Lund, Ida Hogganvik, Fredrik Seehusen, Ketil Stølen. UML profile for security assessment. Technical report STF40 A03066, SINTEF Telecom and Informatics, December 2003.
13. Noldus Information Technology. The Observer XT. Quick start guide for usability, 2006.
14. Robert K. Yin. Case study research. Design and methods. 3rd edition. SAGE, 2003.

Appendix A: Example introduction slide



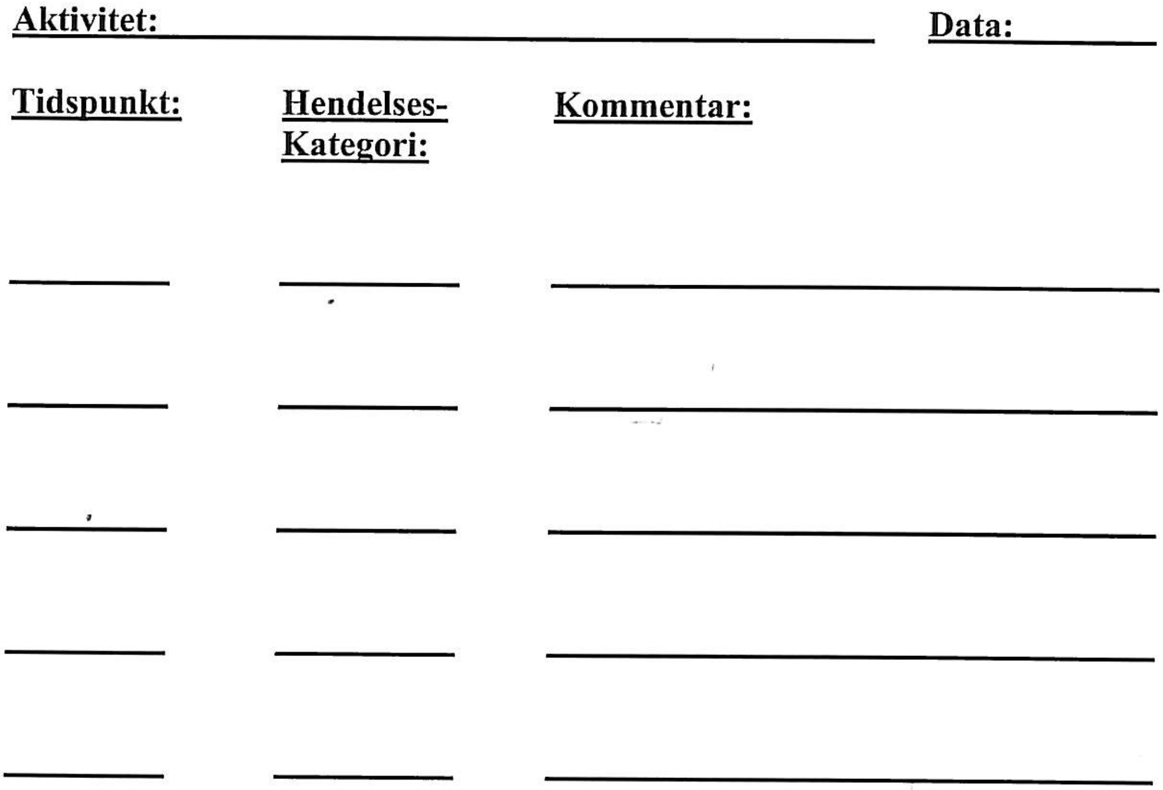
Begreper

$	Aktivum (Asset)	Noe som kunden (targeteier) tilordner verdi og derfor ønsker å beskytte
🔓	Sårbarhet (Vulnerability)	En svakhet, feil eller mangel ved systemet som åpner for at en *trussel* kan skade eller redusere verdien av aktiva
⚠ ⚠ ⚠	Trussel (Threat)	En potensiell årsak til en *uønsket hendelse.*
⚠ ○	Trusselscenario (Threat scenario)	Et scenario som leder fram til en uønsket hendelse
□	Uønsket hendelse (Unwanted incident)	En hendelse som kan skade eller redusere verdien av aktiva

SINTEF IKT

Figure 2: Example introduction slide

Appendix B: Evaluator form and event categories



Aktivitet: ______________________________ Data: __________

Tidspunkt: Hendelses-Kategori: Kommentar:

__________ __________ ______________________________

__________ __________ ______________________________

__________ __________ ______________________________

__________ __________ ______________________________

__________ __________ ______________________________

Figure 3: Evaluator form

Hendelseskategorier

D – Diskusjon mellom deltagerne om betydningen av et begrep eller symbol
S – Spørsmål til risikoanalyseleder om betydningen av et begrep eller symbol
R – Risikoanalyselederen retter på en deltager for feil bruk av et begrep
X – Annet