SN Computer Science (2023) 4:488
https://doi.org/10.1007/s42979-023-01968-x

ORIGINAL RESEARCH

A Case Study of Introducing Security Risk Assessment in Requirements Engineering in a Large Organization

Shanai Ardi[1] · **Kristian Sandahl**[2] · **Mats Gustafsson**[1]

Received: 28 June 2021 / Accepted: 27 May 2023

Abstract
Software products are increasingly used in critical infrastructures, and verifying the security of these products has become a necessary part of every software development project. Effective and practical methods and processes are needed by software vendors and infrastructure operators to meet the existing extensive demand for security. This article describes a lightweight security risk assessment method that flags security issues as early as possible in the software project, namely during requirements analysis. The method requires minimal training effort, adds low overhead, and makes it possible to show immediate results to affected stakeholders. We present a longitudinal case study of how a large enterprise developing complex telecom products adopted this method all the way from pilot studies to full-scale regular use. Lessons learned from the case study provide knowledge about the impact that upskilling and training of requirements engineers have on reducing the risk of malfunctions or security vulnerabilities in situations where it is not possible to have security experts go through all requirements. The case study highlights the challenges of process changes in large organizations as well as the pros and cons of having centralized, distributed, or semi-distributed workforce for security assurance in requirements engineering.

Keywords Security · Requirements analysis · Risk assessment · Process improvement

Introduction

Many software products are used in sensitive infrastructures where software malfunctions or security vulnerabilities can have significant consequences. Various factors like increased digitalization and geopolitical tensions contribute to the challenges faced by software vendors and result in a significant increase of exposure of software products and an increased risk that an adversary can take advantage of the situation. The increase in exposure and attack surface means that individuals and organizations need to deal with a higher risk to any asset they value in the cyber world.

To mitigate such risks, governments and legislators place ever-increasing demands for security assurance on infrastructure operators and equipment and system vendors, which requires these actors to review and strengthen their processes and methods extensively.

Requirements engineering is one of the earliest phases in the software development life cycle in which software vulnerabilities can be introduced into software products if requirements specifications are inadequate. In recent years, efforts have been made to integrate security risk assessment into requirements engineering activities [1–7]. The expected benefit from this is that exposing potential risks early in the requirement engineering phase allows more time for finding solutions to manage the risk. Failure to identify risk in this phase will decrease the overall probability of detecting and preventing vulnerabilities in the product with acceptable costs.

Defining the security objectives of a software product, identifying threats to system assets, estimating the risk level caused by identified threats, and coming up with countermeasures are crucial steps in the process of correctly defining the requirements that will ensure the security of a software product.

Security risk assessment is one of the well-known security activities that is recommended by several software security approaches [8–15] and is a common denominator of

✉ Kristian Sandahl
kristian.sandahl@liu.se

[1] Ericsson AB, Linköping, Sweden

[2] Department of Computer and Information Science, Linköping University, Linköping, Sweden

Published online: 27 June 2023

several security standards [16–18]. Getting an understanding of the involved risks by understanding the involved threat models [14] and problems related to the use cases and misuse cases [5] enables early detection of potential issues. This understanding also provides a rationale for security-related decisions and for security activities designed and introduced to address security issues.

One common characteristic of most existing approaches is that the assessment activities are performed through heavyweight activities that in many cases are validated in theoretical case studies [11, 19]. However, such approaches often turn into key challenges for software vendors that operate using a development context characterized by:

- Many small feature-oriented teams;
- Teams of developers working according to agile methods and with only basic security knowledge;
- Frequent iterations, each comprising a limited number of requirements at a low level of abstraction.

To reconcile the apparent conflict between lean and agile development practices on one side, and traditional heavyweight risk assessment practices on the other, we need to seek out, introduce, and evaluate new practices. Introducing a new method often involves adoption of new technology, changes in work practices, and an additional workload [20]. This can lead to an acceptance issue for the proposed changes. This issue is especially magnified when competing goals and quality factors must be considered when specifying, designing, and implementing new software solutions. Since these requirements often do not add to functionality, development teams tend to lower their priority to meet deadlines [21].

Another aspect we have focused on is the challenge of ensuring access to adequate security expertise and security competence. It is widely understood in security community that the basis for security assurance is security awareness among all members of a development organization [1, 2]. Considering the increased need for security in software products, the demand for competent expert support and building strong cybersecurity teams in organizations is increasing globally. This has resulted in a security skills gap that is getting bigger every year, according to the International Information System Security Certification Consortium (ISC)2. For these reasons, it is crucial for software and information system vendors to utilize existing cybersecurity expertise to meet cybersecurity requirements.

Forming a central security team to ensure that security activities are handled well has been recommended by several researchers [2, 5], but this can introduce bottlenecks, especially in large organizations developing complex products. At the same time, security awareness among developers of a product is the basis for ensuring the security assurance of that product. This poses a knowledge management challenge in the sense that security experts, generally a scarce resource, need to support an often large community of requirements engineers, who are the specialists in the details of the different layers of abstraction in the product. It is usually the case that knowledge about the lowest level of detail is found in development teams. They, not the security experts, are also responsible for implementing and keeping track of the fulfillment of the requirements. The remaining challenge is to find a proper distribution of responsibility between security experts and developers for performing security-related activities.

Another perspective we have included in our approach is to consider security risks associated with requirements in general. This breaks with the traditional mindset in the software development community of considering security separately in requirements engineering and of classifying security requirements as a subgroup of software requirements. Most software requirements are developed in terms of what must happen, but security requirements are driven by a need to mitigate risks and threats to system assets and must be specified in terms of what must not be allowed to happen [22]. Various methods for eliciting, analyzing, and specifying security requirements have been proposed by researchers [22–24]. However, identifying concrete advice for immediate deployment of such methods by software vendors is still challenging, especially in the complex context of large-scale software engineering [25, 26].

Most contemporary methods use risk assessment for security requirements engineering by focusing primarily on the risks involved with stakeholder goals and/or system-level risks introduced by functional requirements and identifying non-functional security requirements [8–12]. This is vital, but we believe that security cuts across abstraction levels and is also a concern at lower levels of detail in the design of a product. We have seen examples wherein vulnerabilities are introduced into a design at the lowest abstraction levels, as shown in the example presented in in this article.

Based on all above-mentioned considerations this article aims at supporting software vendors by proposing a lightweight security risk assessment method to be applied during requirements engineering phase. The technical contribution presented in this article is consisting of getting high-level product requirements, breaking them down to lower abstraction levels (functionalities) during requirement engineering phase and performing a lightweight security risk assessment to fine-tune the functional requirements to address possible security risks. The security fine-tuned requirements then are used in design and implementation (iteratively or depending on the software development process) reducing the probability of introducing associated risks into the end product. In this contribution, we also experiment the pros and cons of

performing these activities by security experts in one end vs requirement engineers in the other end (see Fig. 1).

We focus on the challenges with the utilization of security competence and through a longitudinal case study evaluate the introduction of our security risk assessment method in a large-scale industrial setting to answer the following research questions:

1. What are the difficulties of introducing a security risk assessment method in a big organization that is developing complex systems?
2. When performing security risk assessment, can we bridge the gap between security experts and requirement engineers who are not specialized in security? In that case, what is an efficient distribution of tasks between security experts and requirement engineers?
3. What are the considerations (introduction or training) needed for engineers to achieve acceptable results?

The remainder of the article is as follows: in section "Security Risk Assessment" we provide a thorough description of our lightweight risk assessment method, security risk assessment (SRA). Section "Case Study" contains a description of the case study of the introduction of SRA in a large infrastructure development organization is presented in Section "Case Study". A discussion around findings from the case study and a survey of related work are provided in section "Discussion" and section "Related work", respectively.

Security Risk Assessment

In this section, we present the SRA method and how it is applied during requirements engineering. The SRA method is designed considering the complexity of the product to be developed and aims at providing a simple method for requirement engineers who are not specialized in security to get better understanding of security risks.

The inputs to the SRA method are requirements that emanate directly from customer requests for added functionality and/or from updates and improvements proposed by developers working on the product. Product managers receive customer requirements that are usually goal-like at a product strategy level. Requirement engineers study these requirements, check their feasibility, priority, and the cost of implementation, break them down to requirements in lower abstraction, and define well-defined and testable requirements that initiate the development project and lead to implementation of the required functionality in the final product.

We use the requirement abstraction model (RAM) by Gorschek et al. [27] and define four abstraction levels for requirements: product level, feature level, function level, and component level. Product level is the most abstract level and is comparable directly to the product strategies and indirectly to the organizational strategies. An example of a product-level requirement could be "the system shall provide intrusion detection support". Feature-level requirements are features that the product should support and are an abstract description of the feature itself. An example requirement at this level is "the system shall provide the possibility to log and report security-related events". The function level is, as the name suggests, a repository for functional requirements and describes what a user should be able to perform/do, for example, "users shall be able to subscribe remotely to receive logs of security-related events". The component level is of a detailed nature containing information that is closer to how something should be solved, i.e., on the boundary of design information [27], e.g., "the feature state is enabled/disabled by changing the featureState parameter".

Method Overview

The SRA method is designed, inspired by the definition of risk by Kaplan and Garrick [28] where the risk is defined by the answers to three questions:

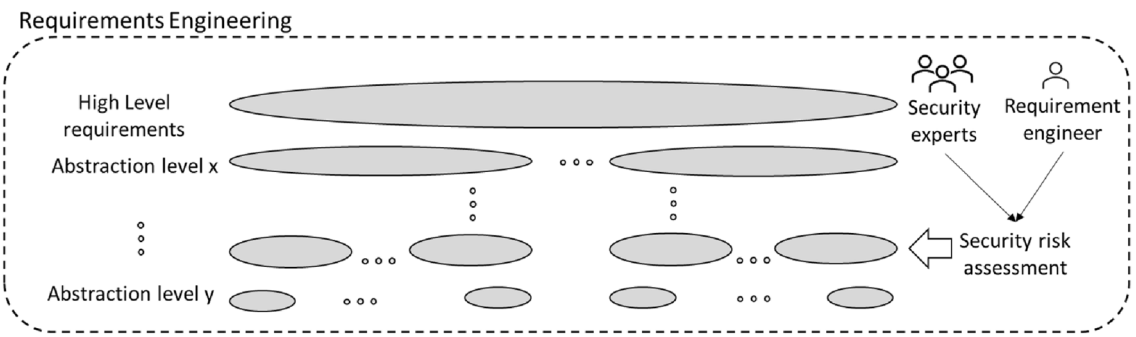

Fig. 1 High-level view of the technical approach

1. What can go wrong?
2. How likely is it to go wrong?
3. If it does go wrong, what are the consequences?

To answer these questions, we need to understand the scenario or undesirable event that may occur during a product's runtime. As shown in Fig. 2, the SRA method consists of three main steps: risk assessment content establishment, risk identification and estimation, and requirements analysis and specification. The preliminary assumption is that there is a feature-level requirement as a starting point to use SRA.

Risk assessment content establishment starts with requirements engineering team getting a feature-level requirement and determining if it is possible to be implemented, what the different implementation alternatives are, and what these alternatives would cost. One of these alternative solutions is then chosen after discussion with the product manager and the function-level/detailed requirements are identified and documented. Such requirements are of a detailed nature representing information that is closer to a description of how something will be implemented. At this step, it is crucial to ensure the following factors while establishing the risk assessment content:

- Security objectives of the software project are known for the requirements engineering team.
- Assumptions in terms of the users of the functionality and the environment in which the feature will function are defined.
- Initial security status of the underlying system is known (in the case of incremental development) based on the security risk assessment of the legacy system (performed in previous releases).
- Use cases of the feature have been identified.
- Requirements engineers are familiar with basic security concepts, such as the three key security requirements for any asset, namely, the CIA criteria: *Confidentiality, Integrity and Availability* in the context of the system they are working with [29].

Risk identification and estimation is done by going through every detailed requirement, documented during risk assessment content establishment, and identifying assets involved in the required functionality, system entry points. Additionaly, attention is given to attacker's capabilities in terms of misusing the functionality, likelihood and impact of functionality misuse, and possible misuse cases involving harm to the identified assets. This is done by answering the following questions:

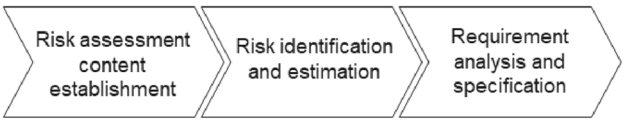

Fig. 2 SRA steps

1. What is the asset (to be protected) in the detailed requirement? An asset is something that is valuable for the feature, for example, the functionality provided by the feature, any new data introduced, variables, control parameters, interfaces, protocols, and/or anything that is included in the use case of the feature.
2. Who has access to the asset and how? The goal is to identify the actors that have access to the asset identified in question 1, for example, end users, developers, and any outsider who might have access to certain variables/parameters through system entry points are considered to be actors.
3. Can the actor/user identified in question 2 misuse the asset? Considering system and environmental assumptions as well as confidentiality, integrity, and availability criteria, can anyone harm the asset (any scenario)?
4. How difficult is it to harm the asset? What is the probability over a certain period (e.g., 1 year) and what is the impact of harm?

The answers to the above-mentioned questions are used to define the risk level using the matrix in Fig. 3. The flowchart in Fig. 4 shows the overview of the SRA process including the above-mentioned steps. In the following section, we illustrate the use of SRA in an example..

Requirement analysis and specification is the last step, where the requirements engineering team uses information of the risks identified for every detailed requirement to fine-tune the use cases covered by the requirement so that the risk would be addressed/prevented. This is done either by reformulating the requirement so that the risk is mitigated, or by defining the corresponding component-level requirements to enforce the risk mitigation.

The flowchart in Fig. 4 shows the overview of the SRA process including the above-mentioned steps. In the following section, we illustrate the use of SRA in an example.

Example

Risk assessment content establishment: Telecom networks consist of several subsystems interacting with each other. These subsystems (nodes) are used by operators to serve their customers (subscribers) with fixed and/or mobile services. The overall system is large and is commercially active over several releases. Consequently, including security in such a system and sustaining security throughout the whole life cycle could be a challenging issue requiring continuous improvements. One example set of features requested by operators is features for self-organizing networks (SON), including self-configuration and self-optimization of the

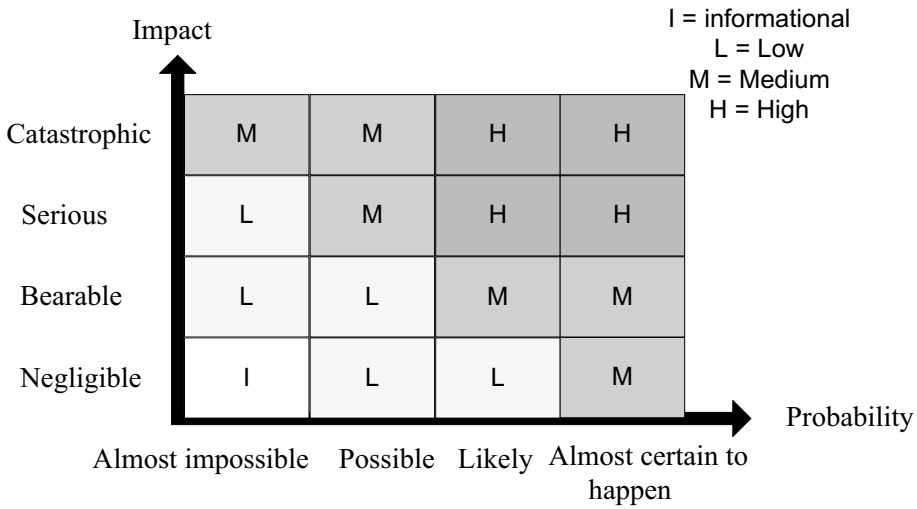

Fig. 3 Risk-level matrix

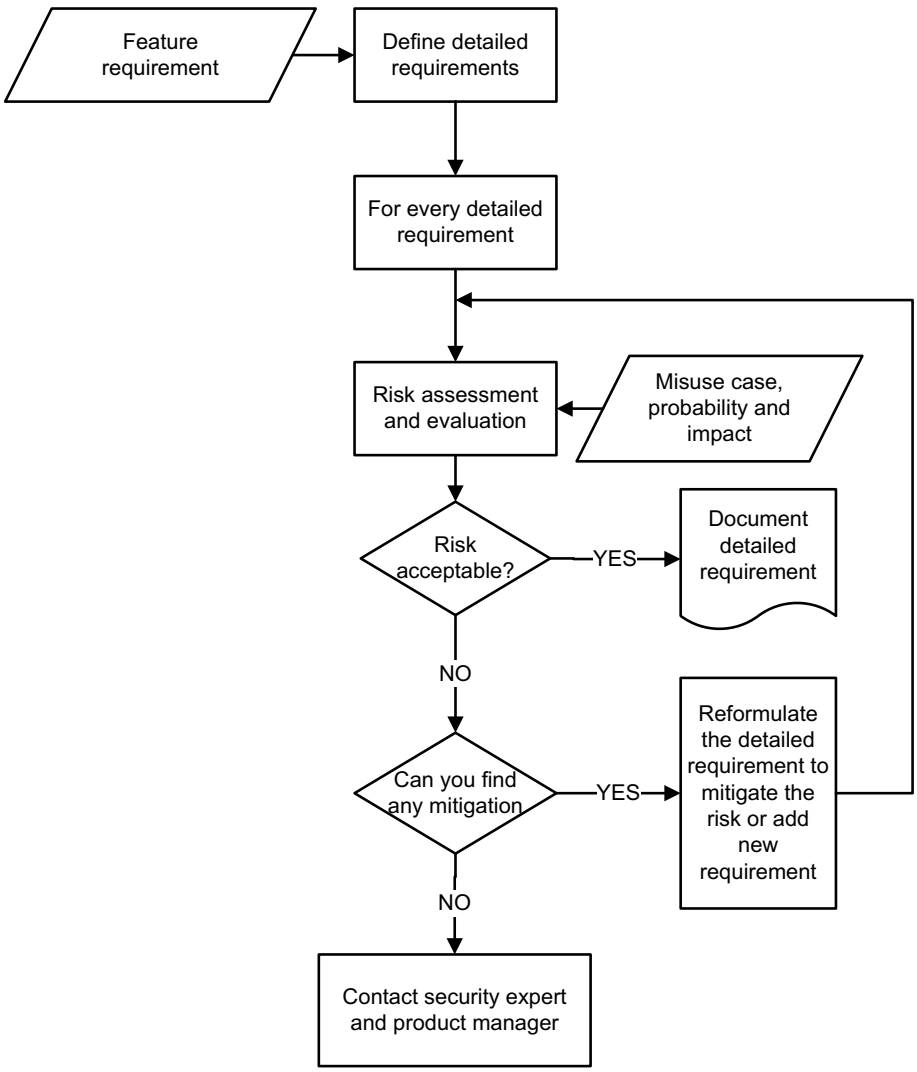

Fig. 4 SRA process

Table 1 Example requirements

Detailed requirement ID	Requirement text
1	Operator shall be able to disable/enable one node or multiple nodes' ANR function when needed
2	The node shall collect and log ANR measurement results from the UE selected for reporting
3	Collecting measurement reports from UE shall be disabled if the maximum number of neighbor relations is reached

nodes in a telecom network. The self-configuration function enables the network to automatically perform installation procedures (plug and play) on the nodes, and self-optimization enables the network to auto-tune its operational parameters using performance measurements that are either performed by the node itself or received from user equipment (UE) [30].

One type of functionality that is provided as a part of self-configuration is the automatic neighbor relations (ANR) feature. ANR is a known feature in the telecom industry [31] for automating operation and maintenance of a specific function (handover functionality) when neighbor nodes provide a telecom service to subscribers jointly. The feature-level requirement in this case is *the node shall support ANR functionality*.

To initiate the development of this feature, we need to define the detailed requirements at the function level. Some examples of these requirements are presented in Table 1.

Risk identification and estimation: We start from the first detailed requirement and perform the risk assessment by answering the questions mentioned above.

Detailed Requirement 1

1. What is the asset? What shall be protected?

   Asset: disable/enable functionality of the ANR function on one or multiple nodes.

2. Who has access to the asset and how?

   Operators (who configure the features), using a configuration GUI.

3. Can the actor/user, identified in the previous question, misuse the asset?

   This is not likely since the assumption is that operators will not harm their own products/network.

4. How difficult is it to harm the asset? What is the probability over a certain time period (e.g., 1 year) and what is the impact of harm?

   The probability is "almost impossible", but the impact is "serious" because the ANR functionality would not be available. According to the matrix in Fig. 3, this will be a low risk.

Detailed Requirement 2

1. What is the asset? What shall be protected?

   Asset: ANR measurement results from the selected UE.

2. Who has access to the asset and how?

   End user (using UE).

3. Can the actor/user, identified in the previous question, misuse the asset?

   It is possible that a malicious actor could modify measurement reports.

4. How difficult is it to harm the asset? What is the probability over a certain time period (e.g., 1 year) and what is the impact of harm?

   The probability is "possible" and the impact is "serious", since the measurement reports are used for certain network planning decisions. This is a medium risk and the requirements engineering team shall revisit the requirement. Depending on the system architecture, there could be different alternatives: for example, if it is possible to get required ANR measurements from a source other than UE the initial design can be modified. If getting measurements via UE is the only way (e.g., as a standard method for all telecom vendors), then an additional requirement shall be defined to validate the received values and minimize the impact of malicious reports.

Detailed Requirement 3

1. What is the asset? What shall be protected?

   Asset 1: collecting reports functionality.
   Asset 2: maximum number of relations (variable).

Table 2 Risk assessment results

Detailed req.	Asset	Actor	Misuse scenario	Probability– impact	Risk level
1	Enable/disable functionality	Malicious operator	An operator can misuse its authority and mess with enabling/disabling	Almost impossible–serious	Low
2	Values in the report	UE	A malicious UE may send modified/misleading data	Possible–serious	Medium
3	Maximum number of neighbors	Malicious operator	An operator can misuse its authority and mess with maximum neighbor value	Almost impossible–serious	Low

2. Who has access to the asset and how?

Developer has access to both assets through the code that implements the functionality and defines the maximum number value.

3. Can the actor/user, identified in the previous question, misuse the asset?

Only if the implementation is modified, the asset can be misused.

4. How difficult is it to harm the asset? What is the probability over a certain time period (e.g., 1 year) and what is the impact of harm?

Developer has access to both assets through the code that implements the functionality and defines the maximum number value. The probability is "almost impossible" because the functionality will be tested to ensure that the requirement is fulfilled, and the impact is "serious" since the functionality will not be available and the feature-level requirement will not be fulfilled. This is a low risk according to the risk-level matrix.

Requirement analysis and specification: The results of the risk assessment are shown in Table 2. This table can be used for residual risk management, helping product managers to decide if the cost of mitigating the risks is acceptable or if the risk is relatively low compared to the mitigation cost and the feature can be delivered as it is. For example, if product management decides to address only the medium risk, it can be addressed in different ways. Example alternatives could be to:

- Define a criterion to accept measurement reports from approved UEs and add this as a new detailed requirement. Also adjust detailed requirement 2 to cover the criteria.
- Accept the risk of getting untrusted data from some UEs, and to minimize the risk get the reports from more than one UE and compare the values before using them. This will lead to several other detailed requirements.

Note that the table can also be used to track the identified risks during the whole development process. This table is reported as a part of the documentation of the requirement engineering phase.

Case Study

We introduced the SRA method in a software development process in a telecom company that is developing complex products using agile practices (e.g., Scrum or a combination of other agile flavors). To evaluate the application of SRA and find answers to our research questions, we performed a case study. One reason for choosing a case study as the evaluation method was to study the problem in its context and evaluate how our proposed method was used in this context. Another reason was to develop an understanding on how a process improvement attempt through introduction of SRA was received in real-life industrial setup. The case study context was as follows: the target organization is a large enterprise offering telecom and multimedia solutions in a highly competitive market. The setup of the team is a mix of both co-located and remote workers, distributed in different locations. The company has around a hundred thousand employees and the unit supporting the case study consists of around 150 engineers. The development model is a combination of customer and market-driven processes in the sense that requirements are collected from both existing and potential customers. The market demands highly customized solutions with requirements that are compliant with domain-specific standards. There are dozens of development teams working on the subject project. The project time may vary between 6 and 12 months and the requirements engineering activity may take up to 4 weeks. The projects are integrated with the previous baseline of the system and only one product exists at the time.

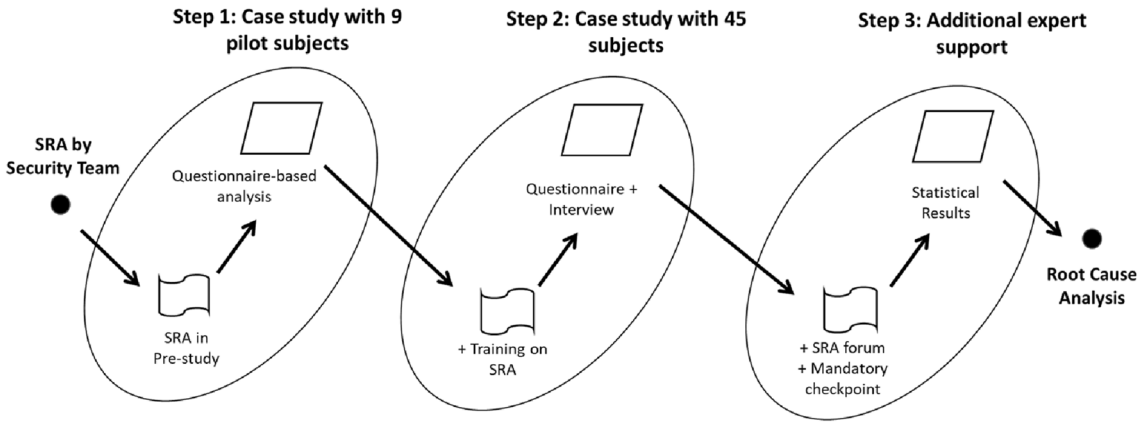

Fig. 5 Case study process

The target organization has a security framework that includes security design rules and generic security requirements to be followed and fulfilled during the software development life cycle. Security risk assessment as part of the organizational security framework is performed for all the products in the company portfolio. A central security team consisting of security experts performs security risk assessments on all feature requirements and feeds the findings back to the development process.

We defined the following variables to be studied to answer the research questions:

1. Deployment of the method in the requirement analysis (during requirements engineering):
   a. Comprehensibility of the documentation introducing the method.
   b. The overhead of applying the method.
2. Applicability of the method in an industrial setup as explained above in the case study context:
   a. Acceptance by requirements engineers.
   b. Shortcomings and improvement possibilities.
3. Effects/benefits of applying the method:
   a. Number of identified vs missed risks.
   b. Awareness.

We used a single case study design as defined in [32], with the telecom company being the overall context and performed the case study in three iterations followed by a final root cause analysis on the findings of the third iteration to identify the way forward, as shown in Fig. 5 and over a period of 4 years. We used the process as described by Runeson et al. to design the case study with a flexible design, based on qualitative data [33]. The first case started with applying the SRA in a certain context by pilot subjects and the contexts of succeeding cases were adjusted after analyzing the results of the previous cases. The initial state of the iterations was a team of security experts performing the SRA activity. We then examined the consequences of fully distributing this task to non-security-expert requirements engineers, and finally a semi-distributed setup where an SRA forum would perform further analysis of results by requirements engineers if needed.

For the first iteration, nine pilot subjects were identified using a focus group [34] of five technical team leaders, who received a presentation of the goal of the case study and, in an open discussion session moderated by the researcher, nominated candidates to be pilot subjects. The selected subjects had deep knowledge of the software product's architecture and its value to the customers. The subjects worked either alone or in a team of two or more engineers. The subjects applied SRA during requirements engineering activity and answered a questionnaire about method conformance, domain conformance, and general feedback.

Process conformance questions focused on characterization of the method and an assessment of how it is performed. Domain conformance questions focus on learning about subjects' knowledge concerning security and requirements engineering, and finally to get general feedback for improving the method.

We analyzed the final feedback from pilot subjects, based on the variables we had defined.

Deployment of the Method

An average of 6 h was spent on performing the risk assessment and documenting the requirements and risks. According to six of the participants, analysis time overhead was considered acceptable with respect to the planned time.

One of the participants mentioned that it took a long time to perform the analysis and one of the participants answered that the time could vary based on complexity of the feature. Seven participants saw no specific hindrance to deploying the method, and one of the subjects felt that the study team's lack of security knowledge could be a major hurdle.

Applicability of the Method

All participants found the method beneficial in finding the risks and that it should be used for all requirements. According to all participants, the introduction presentation was enough to start using the method. In the general feedback, one subject was interested in getting a presentation of the existing security capabilities of the system, as this would help system engineers reuse the already existing mechanisms as risk mitigations. Another suggestion was to provide a list of security best practices to be considered by system engineers when, for example, new attributes and new interfaces are introduced by a feature. The example-driven nature of the method was important for understanding the usability of the method, according to one of the participants.

Effects/Benefits of Applying the Method

One of the participants mentioned that even if there might be no or low security risk, the assessment helped in reaching that conclusion. Only one of the participants was already familiar with security topics and for the rest of them it was their first time thinking about security issues. One of the participants found no risk, four of them identified two medium-level risks each, and the rest identified only low-level risks. Based on the results we concluded that the method could help system engineers to consider the security aspects of technical solutions using the proposed method with minimum overhead. All medium-level risks were reported to product managers to discuss a cost-effective mitigation or to be considered in negotiations with customers if required. Two of the risks resulted in new feature-level requirements from product managers and the rest of them were not prioritized in the upcoming release from a business point of view.

Iteration 2: Case Study with 45 Subjects

After the first iteration was completed, the decision was to extend the scope and apply the method to an entire release project, wherein all of the system engineers on that project would apply the method to all of its features in that project.

Subject and Case Selection

The target release project consisted of 45 features to be implemented and integrated into a legacy telecom product. In this iteration, we provided a 1-h training for all system engineers studying these 45 feature requirements to present the security risk assessment method. We also modified the security impact chapter in the mandatory document that was to be written in the pre-study phase. This document describes the systemization of the feature and includes the list of detailed requirements. The security chapter was updated to require that the results of the security risk assessment be documented and reported in the chapter.

Data Collection

We had three sources for data collection. We used a two-step qualitative data collection method in this iteration, which took the form of a questionnaire to be answered by the subjects, followed by individual interviews to get a more in-depth view of the subjects' opinions. In parallel, all the reports were systematically reviewed by the central security team and the data provided in the security impact chapter were reviewed. The goal was to analyze the outcome of the modifications to the pre-study process and compare the results of the security risk assessments done by the subjects with the results of the same analysis as performed by the central security team. This approach helped to determine whether all the risks had been identified by the subjects. To ensure ethical considerations, all subjects were informed about the purpose of the activity and asked to give consent on the use of their contributions in this research approach. This included the information they provided about their own technical background and experience.

As with the first iteration, the questionnaire included questions about process conformance (PC), domain conformance (DC), and general feedback (GF).

Process conformance (PC) and general feedback (GF)

The questions on PC and GF, which placed more focus on gathering statistical data about the application of the method, were as follows:

1. (PC) Did you attend training on the method and the new document template?
2. (PC) Did you use the proposed method?
    a. Yes: describe the differences you see between this template and the old one.
    b. No: why not?
3. (PC) How long did it take for you to perform the risk analysis on the detailed requirements and document it? How long was the whole study?
4. (GF) What are the pros and cons you see in this method, as mentioned in the security impact chapter?

5. (GF) Talk about your opinion regarding any significant problems that might hinder the deployment and use of the method.
6. (PC) How much training do you think is needed to be able to use the method?
7. (PC) Did you identify any risk for your feature and what was the level of the risk?
   a. Yes: what did you do with the risks you identified?
   b. No: why do you think you did not identify any risk?
8. (GF) What are your suggestions to improve the risk analysis method and instruction document?
9. (GF) Is there anything more you would like to add?

Questions about the subjects' pre-knowledge in security and their depth of knowledge about the product were handed out in a separate set.

Execution

We organized training sessions and presented the SRA method and examples of how to apply it to all subjects. During the roadshow, we also went through the changes applied to the "security impact" chapter in the pre-study report document template.

After the project was closed, the subjects were asked to answer the questionnaire and then invited to a 30 min interview (for each functional requirement). A total of 41 subjects responded to the questionnaire and participated in interviews. The interview sessions were semi-structured [34] with a mix of open and closed questions. The interview agenda is:

- Meeting starts with a presentation of the interviewer.
- The interviewer explains the goal of the interview.
- The subject is asked to sign the statement of consent to use the data in the research project.
- The subject is asked to present information about their own background.
- The subject provides information on what the study is about.
- The interviewer walks through the answers provided by the subject and takes notes of the reasons for the answers.
- The interviewer provides information on how the data will be analyzed.

The notes from each interview were sent to the subject after the interview for a second review.

Table 3 Four categories of subjects in iteration 2

Categories of subjects	Attended the roadshow	Not attended the roadshow
Applied the method	$A=16$	$C=10$
Not applied the method	$B=4$	$D=11$

Results

The questionnaires were printed, and the answers provided by the respondents were independently analyzed and categorized by the three authors. The independent analyses were subsequently compared and reconciled with only minor inconsistencies noted that could all be resolved through a joint review of the interpretation of the answers and clarifications given in the interviews. There were four categories of subjects as shown in Table 3.

Deployment of the Method

The studies in which the method was applied were of varying complexity and length. One subject reported having spent 5 min out of 3 months, and another reported having spent half a day out of 2 weeks. A majority (18 out of 26) of the subjects that used the method report having spent 2 h or less on applying the method. Four subjects reported having spent more than 2 h. Four subjects did not answer the question about how much time was spent applying the method. The subject spending half a day out of 2 weeks had not attended the roadshow and reported having to overcome a threshold for using the method for the first time.

In the questionnaire, most respondents provided feedback and suggestions for improvements. The most frequent feedback (from more than half of all subjects) contained a suggestion to introduce a concept of "No impact" to be used when a simple review makes it obvious that the change being studied will not introduce new risks. Other common items of feedback were each expressed by about a quarter of the subjects: suggestions to provide more examples of risks as it might occur in different types of system features, a concern over slip-through or that risks might be introduced in later stages of the process, that the context in which risk was to be assessed needed to be better defined, and mention of the use or need for a subject matter expert to complete the assessment.

In the context of knowledge supply, approximately, one-third of the subjects expressed a need for getting expert support when needed and one-third expected improved/additional training.

Other feedback included suggestions for using structured queries, the need for continuous training, and the need to

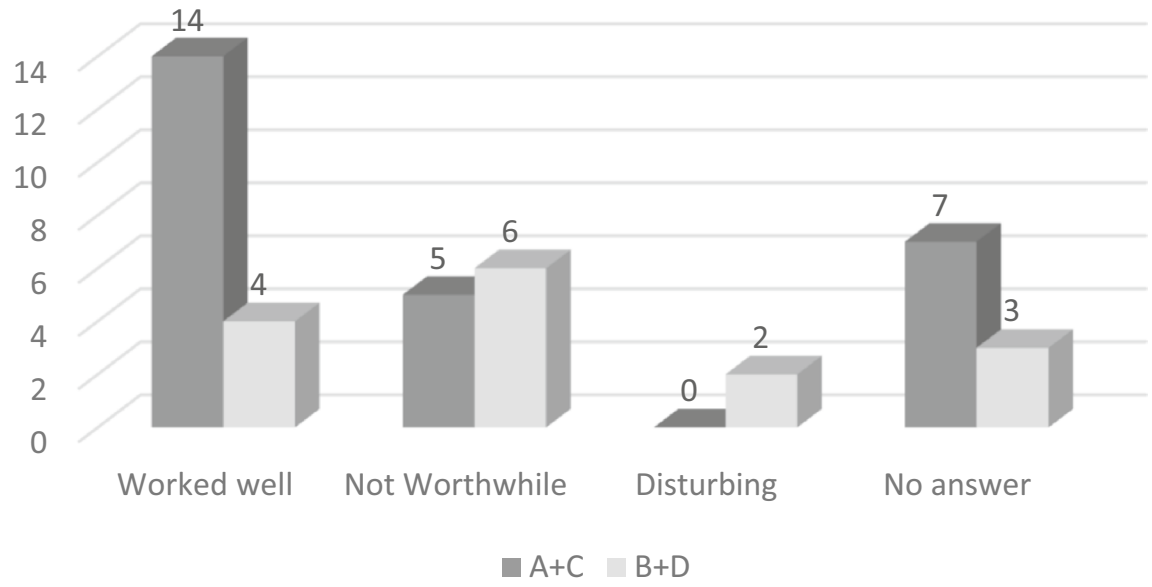

Fig. 6 Applicability of the method according to subject categories

also consider risks holistically at the system level and not just at the feature level.

Applicability of the Method

Among subjects in categories A and C (see Table 3) that have applied the method ($n = 26$), our analysis shows a generally positive or neutral attitude toward the setup. 14 subjects categorized their impression of the method as 'worked well' or 'worthwhile', while 5 subjects expressed an opinion of 'not worthwhile'. The remaining seven subjects that used the method did not state any valuation. See Fig. 6.

Of the 15 subjects in categories B and D who did not apply the method, six found the method 'not worthwhile' and two found it somehow disturbing. Four subjects had applied parts of the method and thought it 'worked well'. The remaining three did not express an opinion about the method.

Effects/Benefits of Applying the Method

We analyzed the results to list the risks identified by the subjects. The pre-studies were then reviewed by security experts and the cases where additional risks were listed by security experts were identified. Figure 7 shows the number of risks identified by subjects vs. security experts based on subject categories.

We also went through the collected data to identify the possible benefits of applying this method regarding increasing security awareness among requirements engineers. We were able to categorize the answers into three main categories as in Fig. 8:

a. 22 subjects who clearly stated that they had no security background and became aware of security and security issues during this case study.
b. 13 subjects who had at least basic security knowledge prior to the case study, but who also found it useful to be given instructions on how to perform security assessments.
c. Six subjects who did not have basic security knowledge and it was not evident that the proposed approach and case study affected their security awareness.

Concluding Remarks: Iteration 2

In this study, we observed that those who have participated in the training and tried to use the method found almost the same number of risks as the security experts. The cost of applying the method was acceptable. When the results were presented to the company, two things were concluded:

1. The value of making early security risk analysis on a detailed level of requirements is high and should be continued.
2. However, as many subjects indicated, more training and support from security experts were necessary.

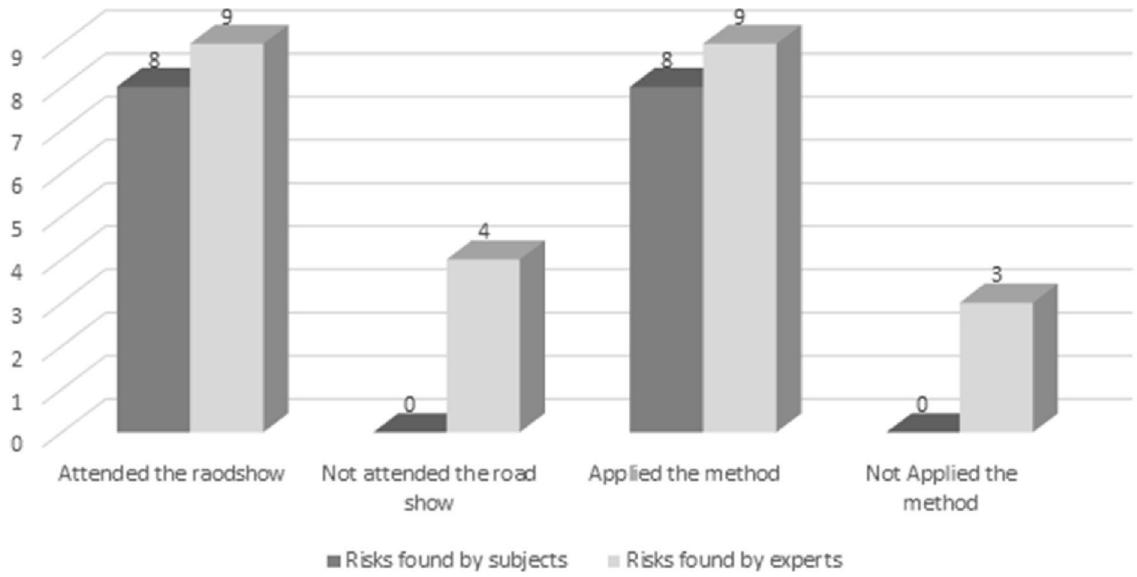

Fig. 7 Risks found by subject categories vs. security experts

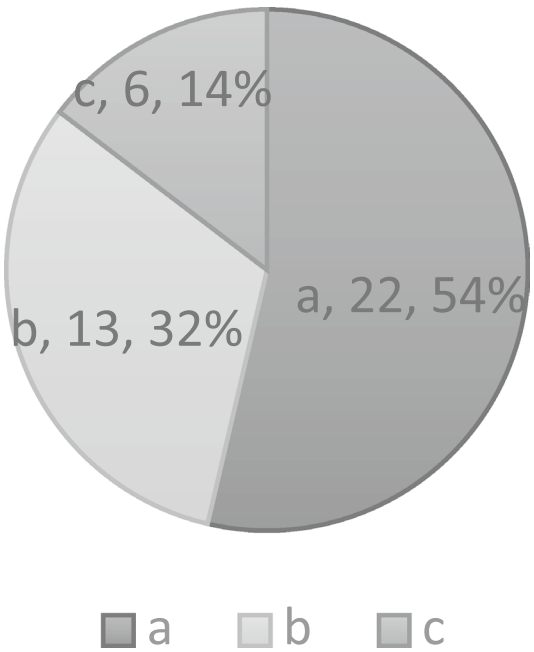

Fig. 8 Security awareness categories

Iteration 3: Additional Expert Support

As mentioned earlier, the initial state for starting the process of proposing our approach was risk assessments of implemented features conducted by a central security team consisting of security experts. We then studied the impact of decentralizing this activity to system engineers with no specialization in security. This way of working continued in the same way as in iteration 2 above with more training and dialog with security experts. Feedback was continually collected. As the final step in the third iteration, we modified the proposal to provide security expert support to system engineers performing the risk assessment when needed.

In this iteration, the risk assessment activity was included as a mandatory checkpoint in the pre-study process during requirement analysis and it was defined as a part of the definition of "done" for the pre-studies. This strengthens the requirements on using the method compared to iteration 2. The security expert team was renamed to security risk assessment (SRA) forum and an improved workflow was defined as shown in Fig. 9.

In this workflow, requirement engineers perform a risk assessment according to the method in section "Security Risk Assessment" and sends the results to the SRA forum. A security expert then goes through the results and either approves them or identifies the need for expert involvement and in-depth analysis. If necessary, the in-depth analysis is then done in an SRA workshop and the security expert team assists the requirements engineers with in-depth analysis. In this way, there is already a quality control process being performed on the assessments done by the requirements engineers. We applied this process in eight releases projects in the subject organization.

To evaluate the outcome of these changes, we used the statistical data collected by the organization for follow-up purposes. The organization uses this data to go through the pre-study documentations and review the security risk assessment results. As a result of this review, all studies must have a proper security risk assessment

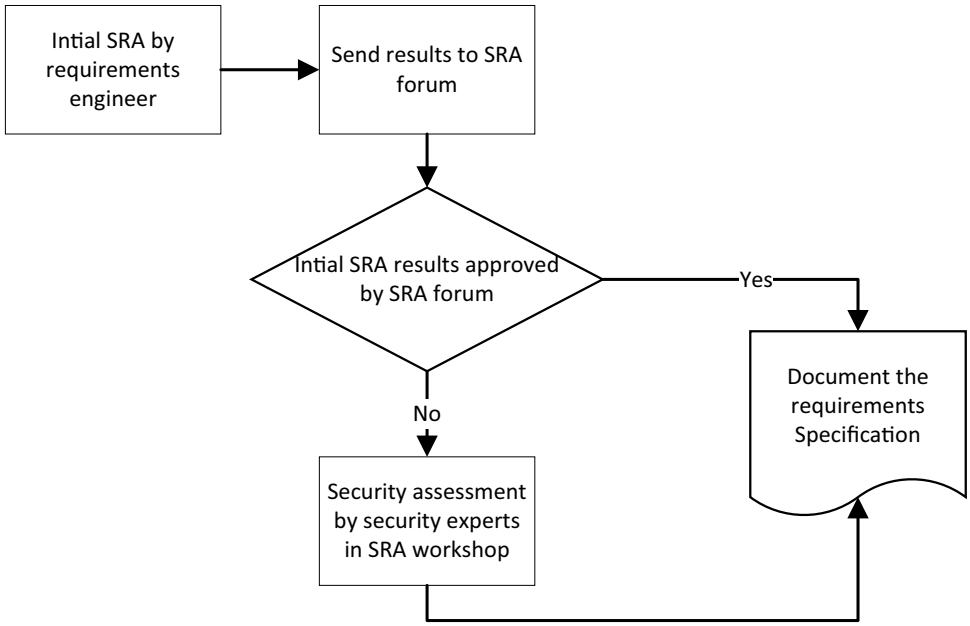

Fig. 9 SRA workflow

documentation, approved by the SRA forum. By analyzing this statistic, we identified three categories of studies:

1. Studies with missing security risk assessment documentation (no assessment reported in pre-study documentation).
2. Studies with incomplete security risk assessment data provided in the pre-study documentation and no communication with the SRA forum.
3. Studies with proper security risk assessment provided in the pre-study document, which had passed through the SRA forum (with the results of the analysis performed by the requirement engineers either being approved directly or after expert involvement through an SRA workshop).

Table 4 shows the statistics for these categories. The requirements engineers responsible for the studies in categories 1 and 2 were invited to a root cause analysis workshop and their input on the causes of identified issues was discussed with them. All the participants were encouraged to share their ideas and give feedback about the method, and a recorder took notes on the board to capture all of the input.

The five whys method [36] was then used to identify the root causes of the identified issues. The following root causes were identified:

- Security awareness/competence: Due to reorganizations, and responsibility relocations, new teams started on the project without getting the planned training on applying the method. The statistics in different releases have a correlation with changes in the organization.
- The training material is old and needs to be refreshed and adjusted to the agile teams' way of working which changes regularly.
- The old template for the pre-study document (without the security chapter) was used for documenting the pre-study in some of the studies.
- There was a lack of communication to pre-study drivers that it is mandatory to complete the security chapter.
- The SRA forum was not sufficiently introduced to the new pre-study drivers.
- The pre-study documentation including the security chapter (to include security risk assessment) exists, but is not linked into project management tools, which led to missing documentation when working on the statistics above.
- The method is focused on the new (delta) functionality in the product, since the new requirements are used as an input to the security risk assessment. However, the study driver must have access to the security risks that were identified for the legacy system when the new feature is an incremental change in functionality.

Based on these findings several corrective actions were identified: security guardian(s) were appointed in each project to ensure that the security risk assessment would be in place before respective project milestones/checkpoints. The guardian also supports the function of the security risk assessment forum representing the respective project. Security guardians are project managers that ensure mandatory project activities are performed, including security.

It was ensured that the release project checklist is updated and includes security risk assessment as a mandatory

Table 4 Statistics on security risk assessments

Release date	# of features in release project	Studies lacking security assessment		Studies with proper security risk assessment (%)
		Not done assessment (%)	Incomplete assessment (%)	
17.Q1	62	11	27	62
17.Q2	57	11	7	82
17.Q3	83	27	11	62
17.Q4	65	34	23	43
18.Q1	70	21	28	51
18.Q2	101	30	22	48
18.Q3	105	26	16	58
18.Q4	55	16	4	80
19.Q1	198	25	0	75

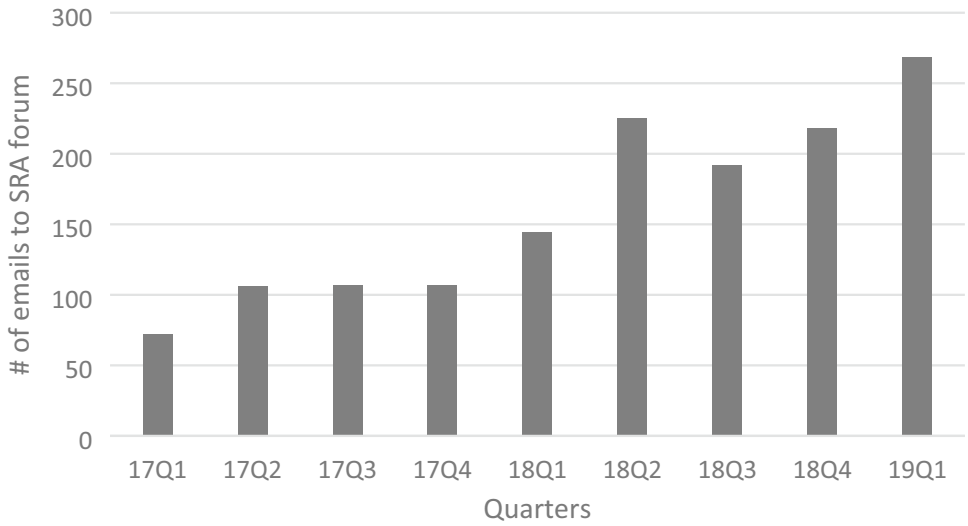

Fig. 10 Statistics on number of emails sent to SRA forum by unique individual senders

checkpoint to pass. It was also ensured that the SRA forum representative is invited to the final review of the pre-study documentation. Training material was updated and the training program was improved to include SRA forum information to cover all new engineers. It was ensured that the documentation template would be updated as required.

We also analyzed the email throughput in the SRA forum mailbox and listed the statistics about the emails sent to the SRA forum by unique individual senders (requests for SRA forum support) as shown in Fig. 10. The increasing trend in the number of emails can be interpreted as a sign of increased security awareness among pre-study drivers and the increasing number of security risk assessments performed for studies that require approval from the SRA forum.

Discussion

We proposed the application of a method in a telecom company and studied different aspects of introducing such a method in the context of the target company. This approach examined the target company's journey through several steps of changes, based on a continuous improvement mindset.

The journey started from an initial status of security risk assessments being performed by a centralized team of security experts who did not have deep technical knowledge of the lower abstraction level of the respective functionality of the system under assessment. This assessment was performed in requirement verification. During our research journey, we examined introducing security risk assessment activity to be performed during requirements engineering and through a completely distributed approach, by letting requirements engineers with deep technical knowledge, but no specialization in security, perform the assessment. One of the goals was to ensure that introduction of security

considerations into the product's functionality as early as possible, and the other goal was to eliminate the bottleneck of a central team conducting the security assessment activity for all features of the project. In the third round, the approach was modified to examine a cooperative setup involving both individual requirements engineers as well as the central security team. The first iteration of our case study can be defined as exploratory [33], which helped us to find out how the requirements engineering was performed and if SRA could be applied by pilot subjects. The output from this study led us to the idea of training subjects on how to apply SRA and study the outcome in the second iteration. The second iteration could be defined as explanatory [33], as it was used to seek an explanation of the outcome of iteration 1 (subjects performing SRA without receiving any training) with the case where subjects did get a training on the method. The output from this study helped us to define the type of support needed by security experts. The third iteration focused on a descriptive study of the organization in the last 2 years with requirements engineers using the SRA method and improved the application of the method by involving security experts to triage when needed.

The results of our extensive case study allowed us to answer the research questions we had defined:

What are the difficulties of introducing such a method in a big organization that is developing complex systems? We realized in each case study iteration that by adding SRA to the existing way of working and to the development artifacts in the company, the application of the method is impacted by the efficiency/deficiency of the original artifacts. Changes in the organization must also be monitored to adjust the proposed method.

When performing security risk assessment, can we bridge the gap between security experts and requirement engineers who are not specialized in security? In that case, what is an efficient distribution of tasks between security experts and requirement engineers? The case study showed that expert involvement could not be eliminated to ensure that the quality of the risk assessment is acceptable and that all risks are identified. Based on this finding, we also learned that changes of this type must be managed over time to achieve the desired results. It was also observed that the bottleneck issue could be solved in a cooperative approach and, as we see in the results of iteration 2, most subjects reported manageable overhead with respect to total time of the prestudy. Considering the increasing number of features to be implemented (see Table 4) in a project, the overhead factor became important.

What considerations (introduction or training) are needed for engineers to achieve acceptable results? During all iterations of the case study, one of the main elements of feedbacks we received was related to training and providing examples and background material for requirement engineers as well as the possibility of supervision/consulting supported by security experts. The results clearly showed that having basic security knowledge, as well as understanding the purpose and expected outcome of the security risk assessment is a crucial prerequisite to achieving the desired results. It is also important to ensure that training is refreshed continuously and is adapted to the changes in the organization, development processes, and daily way of working. Note that the emphasis in our case study has always been on basic security knowledge and understanding what security principles are in terms of confidentiality, integrity, and availability rather than knowledge of sophisticated attack patterns, threat models, etc.

In summary, despite all obstacles, comparing the initial state with the existing state, we see an obvious increase in security awareness in the company and among developers, since everyone is expected to see security considerations as a part of the functional requirements to be developed in the final product. We have effectively shifted security risk assessment that had previously been done in later stages of the development to the earliest stage where the requirements are elicited to implement the functionality. Through the continuous improvement process, we managed to reform the central team of subject matter experts, who were serving the development activities in a support function to the SRA forum that acts in a corrective function.

As stated by Runeson et al. [33], about the nature of the case studies, the case study methodology can primarily be used for exploratory purposes, but it can be used for explanatory and descriptive purposes if the generalizability of the situation or phenomenon is of secondary importance. During design and implementation of the case studies, our assumption was that the results could be transferable, and we believe the results provided a deeper understanding of the phenomena under study. We also believe that providing the details of factors defining the context of cases study (the size of the company, complexity of the product, the type of development process, the size of the project, and the abstraction level of requirements) supports transferability goals. It allows the readers of our results to make inferences about how our findings match their context and which part of our solution can be transferred to their respective settings [35]. Any software or system which has interfaces and/or is communicating with its surrounding is subject to risks and needs a security risk assessment to be prepared for being resilient. SRA can be performed on any system and in any abstraction level, on a whole system within its boundaries or on the components of any system and is not limited to telecom products.

We analyzed the validity threats of our results based on Runeson et al.'s checklist [33]. For construct validity and to ensure that researcher and subjects have the same interpretation of the operational measures, we used both questionnaire

and interview sessions to go through the answers to the questionnaire. The case study design and three different iterations of the case study contribute to the internal validity and help to ensure that various factors are considered in the findings. This includes the factors of the technical background and security competence of the subjects, as well as the organizational way of working and processes that are already in place, but which may differ from project to project. The same characteristics of our approach help with analysis of the external validity, since it is performed by different subjects in different projects over an extended period of time. To support the reliability of the findings, all steps of the case study activities were designed and reviewed by three researchers, and to reduce bias by individual researchers, we conducted data analysis after the third iteration by three researchers independently.

Related Work

To identify related work in risk-based requirements engineering, we performed a literature review and went through the publications on research approaches to security risk assessments applied in requirements engineering as well as similar empirical studies. During this literature study, we compared the novelty of our contribution with existing research contributions, considering that:

- We use security risk assessment in requirement analysis of all functional requirements, not just security requirements.
- Our focus is on assets at a lower abstraction level than similar approaches, which start mostly from strategic interests of stakeholders or objectives. Going from system-level to subsystem-level analysis highlights the functional aspects of the solution to be developed that might be missed in higher-level analysis [37]. Identifying risks at this level helps us to refine the solution to counter the risk by choosing a security-tuned solution.
- We emphasize the technical knowledge of requirements engineers supported with security training and security expert consulting (when needed), to distribute the overhead of security activities instead of using the limited resource of security experts.
- We have empirically verified the proposed approach in an industrial setup, over the course of several years and in large-scale software development projects using agile methods. This has provided a good understanding and lessons learned about the realities of introducing such approaches in a real-life setup.

Identifying system assets, formulating significant threats to the software system, and associating the probability and impacts of risks with the system requirements have been presented in several articles and in various dimensions [8–12, 38]. Franqueira et al. introduce an agile security risk management approach that addresses the topic of performing risk assessments in development process iterations. This approach focuses on supporting decision making on mitigations to be incorporated into the next iteration of development [8]. Asnar et al. propose a goal-oriented approach for analyzing risks along with stakeholder interests and identify countermeasures as a part of system requirements [13]. In a similar approach, Mayer et al. [9] propose using risk analysis in security requirements engineering of information systems that focus on business assets. Firesmith [22] presents different types of security requirements and provides guidelines for system engineers to specify security requirements. These guidelines are used to ensure that security requirements are not confused with architectural security mechanisms. Our approach is similar to these works in that it focuses on the knowledge of system engineers rather than security engineers. Laoufi [10] also aims at identification of security requirements for information systems from risk analysis and uses ontologies to do so. He also focuses only on security requirements and no empirical evaluation of his approach is presented. All the approaches mentioned aim at identifying security requirements using security risk assessment, compared to our approach that applies risk assessment to all requirements, resulting in requirements that have been fine-tuned for security. In this way, we ensure that the security considerations are built into the requirements and consequently into design of the system under development.

Note that there are various definitions for security requirements in the requirements engineering and security engineering communities. Within requirements engineering, security is often classified as a non-functional requirement [39, 40]. An example from the security engineering community, common criteria (CC) [41] distinguishes between two types of security requirements: functional and assurance. Security functional requirements describe security properties that users can detect by direct interaction with the system or by the systems' response to stimulus. Security assurance requirements are process requirements that require active investigation and evaluation by the IT system to determine their security properties [42].

We agree with the statement that "no common agreement exists on what a security requirement is" [23] and various approaches [22, 24], [43–47] define different extents for security considerations covered by security requirements and different levels of details on how to cope with security requirements. In our approach, we do not separate security and non-security requirements; instead, we propose to define and "security-tune" function-level requirements after considering the relevant security risks. Considering security, as a part of designing the

solution will ensure that security aspects are not ignored. Haley et al. [43] recommend security requirements "… to express what is to happen in the given situation, as opposed to what is not ever to happen in any situation." In our approach, we use the same mindset and propose risk analysis for every detailed requirement, considering the context in which the requirement is to be implemented.

Focusing on the assets, in a similar approach to ours, Vasilevskaya et al. use risk assessment (consequence assessment) to decide which asset to prioritize for protection, and this is used as an input to selection of security mechanism to protect the asset in embedded systems [48]. This approach also combines security expertise with embedded system engineering knowledge, although the approach does not target requirements engineering.

We reuse the concept of misuse cases [49] to detect the possibilities to abuse the functionality and identify the risks. Misuse cases are introduced by Sindre et al. [50] and extend traditional use cases by specifying behavior not wanted in the proposed system. Mwambe and Echizen [51] focus on supporting information systems security during the design phase. As an extension of unified modeling language (UML) activity diagrams, mal-activity diagrams (MAD) have also been used to model malicious and risk mitigation processes.

In our literature review and going through the relevant survey studies on information security risk analysis and security requirements engineering such as [52, 53], we found some similar empirical studies with industrial setups. Oyetoyan et al. [11] presented an empirical study with an extensive presentation of the case study and its results with partially overlapping research question. Challenges of applying threat modeling in agile development are presented by Cruzes et al. [54]. This approach uses a similar research method to ours and presents challenges to adoption of threat modeling as a security practice in a smaller development organization. The challenges identified by this contribution are mapped to our findings in some of the cases such as challenges with having distributed teams or the importance of providing security expert support in certain discussions.

Morrison et al. [55] surveyed several security-focused open source projects to collect evidences on adherence to the number of software development security practices. According to their findings, training is positively correlated with the use of these practices and we see a similar finding in our work as well: training system engineers improves the use of security risk assessment as a security best practice. In a similar way, we also observed that the use of a simplified security risk assessment method that is designed with ease of use in mind is impacted by various factors.

Conclusion

Security has become a critical part of nearly every software engineering project and identifying and performing proper activities to ensure security is one of the challenges of software vendors. The work presented in this article proposes the introduction of a risk assessment method in requirement engineering and studies the realities and challenges of applying this method in a real-life industrial setup. The goal of this validation step was to see if it is possible with a small effort to introduce such a risk assessment approach. In this approach, requirements engineers who are not specialized in security attempt to efficiently find security risks early in the development process as well as to gather information on the outcome. Lessons learned from this validation activity showing the need of systematic interaction between security experts and requirements engineers may provide a basis for being prepared and facilitating similar approaches.

The risk-based requirements engineering method provides incentives in the sense that system engineers find the risks involved with their proposed solutions immediately. When developing solutions, they can react accordingly by fine-tuning the solution or by adding new requirements. This is an immediate perceived benefit and is one of the factors that increases the acceptance of the method. In our industrial case study, we examined the applicability and usability of the method when used by distributed teams, developing complex products in agile ways. We started on a small scale, iteratively improved the application of the method, and increased the scale.

For future work, we are focusing on applying the method in a different organization to measure the correlation in findings. The next step of our research is to analyze the quality and quantity of risks identified by the subjects and compare them to similar case studies with security experts as the subjects. This could be performed in a quantitative approach to identify the risk coverage of the method. Another area to be considered as future work is to create a database of different types of known security risks which can be used as a reference during the assessment performed by requirement engineers. Such a database would of course need to be supported by known security modeling methods such as various threat models, attack trees [5], etc. to ease the navigation and usage. SRA is not limited to identify a specific type of risks and answering the mentioned three questions and the type of risk to the identified assets can result in any type of risks. By providing a starting point for requirement engineers through a list of example risks for similar systems, there is a possibility to minimize the probability of missing a risk type.

Acknowledgements Funding for this work was provided from Ericsson AB and Linköping University. Proofreading of a late draft of the article was done by Brittany Shahmehri and David Partain.

Author Contributions A comprehensive description of the security risk assessment (SRA) method; a longitudinal case study of introducing SRA in a large organization; a survey of related work.

Funding Open access funding provided by Linköping University. This research was funded by Ericsson AB and Linköping University.

Data Availability No raw data is available.

Code Availability Not applicable.

Declarations

Conflict of interest Ardi and Gustafsson are both employed at Ericsson AB, full time. Sandahl is employed full time at Linköping University and is a senior member of IEEE.

References

1. McGraw G. Software security. IEEE Secur Priv. 2004;2(2):80–3.
2. Howard M. Building more secure software. IEEE Secur Priv. 2004;2(6):63–5.
3. Ardi S, Byers D and Shahmehri N Towards a structured unified process for software security, Proc. Int. Workshop on Software Engineering for Secure Systems (SESS), Shanghai, China, pp. 3–10. (2006)
4. Lipner S. B The trustworthy computing security development lifecycle, Proc. ACSAC 04, 20th Annual Computer Security Applications Conference, Tucson, USA, pp. 2–13. (2004)
5. McGraw G. Software Security: Building Security In. Boston: Addison-Wesley; 2006.
6. Viega J, McGraw G (2011) Building Secure Software: How to Avoid Security Problems the Right Way. Boston: Addison-Wesley; 2011.
7. McGraw G, Migues S, West J, (2018) Building Security In Maturity Model (BSIMM 8), https://www.bsimm.com/. Accessed 2020–02–10
8. Franqueira V.N.L, Bakalova Z, Than Tun T, Daneva Towards agile security risk management in RE and beyond, Proc. 1st Int. Workshop on Empirical Requirements (EMPIRE), Trento, Italy, pp. 33–36. (2011)
9. Mayer N, Rifaut A, Dubois E Towards a risk-based security requirement engineering framework, Proc. 11th Int. Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'05), Los Alamitos, USA, pp 83–97. (2005)
10. Laoufi N, From risk analysis to the expression of security requirements for systems information, Proc. Fourth International conference on Cyber security, cyber warefare and digital forensics, Jakarta, Indonesia, pp 84–89. (2015)
11. Oyetoyan T. D, Soares Cruzes D, Gilje Jaatun M, An empirical study on relationship between software security skills, usage and training needs in agile settings, International Conference on Availability, Reliability and Security, Salzburg, Austria, pp 548–555. (2016)
12. Savola R. M, Väisänen T, Evesti A, Savolainen P, Kemppainen J, Kokemäki M. Towards risk-driven security measurement for android smartphone platform, International Information Security South Africa conference, Johannesburg, South Africa, pp. 1–8. (2013)
13. Asnar Y, Giorgini P, Mylopoulos J. Goal-driven risk assessment in requirements engineering. Requir Eng. 2011;16(2):101–16.
14. Howard M, Lipner S. The security development lifecycle. Redmond, Washington: Microsoft Press; 2006.
15. The CLASP application security process. Secure Software Inc. 2005. https://cwe.mitre.org/documents/sources/TheCLASPApplicationSecurityProcess.pdf. Accessed 23 Jun 2023.
16. Guide for conducting risk assessment, NIST Special Publication 800–30, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf. Accessed 23 Jun 2023.
17. ISO/IEC 27000: https://www.iso.org/standard/73906.html. Accessed 23 Jun 2023.
18. ISC2 Cybersecurity workforce study (2019), https://www.isc2.org/Research/Workforce-Study. Accessed 23 Jun 2023.
19. Herrmann A, Morali A RiskREP: Risk-Based Security Requirements Elicitation and Prioritization, Proc. Perspectives in Business Informatics Research, Riga, Latvia, pp. 155–162. (2011)
20. Umarji M, Seaman C. Predicting acceptance of software process improvement. SIGSOFT Softw Eng Notes. 2005;30(4):1–6.
21. Borg A, Yong A, Carlshamre P, Sandahl K The bad conscience of requirements engineering: an investigation in real-world treatment of non-functional requirements, Proc. 3rd International Conference on Software Engineering Research and Practice in Sweden (SERPS'03), pp. 1–8. (2003)
22. Firesmith DG. Engineering security requirements. J Object Technol. 2003;2(1):53–68.
23. Tondel A, Gilje Jaatun M, Meland PH. Security requirements for the rest of us: a survey. IEEE Softw. 2008;25(1):20–7.
24. Mead N.R, Houg E.D, Stehny T.R, Security quality requirements engineering (SQUARE) methodology, Software Eng. Inst. Carnegie Mellon University, Techncial Report CMU/SEI-2005-TR-009. (2005)
25. Luburic N, Sladic G, Milosaljevic B Applicability issues in security requirements engineering for agile development, Proc. International Conference on Applied Internet and Information Technologies, Bitola, Macedonia, DOI:https://doi.org/10.20544/AIIT2018.I02.(2018)
26. Poller A, Kocksch L, Turpe S, EPP F. A, Kinder-Kurlansa K. Can security become a routine?: A study of organizational changes in an agile software development group, Cumputer Supported Cooperative Work (CSCW), Portland, Oregon, USA, pp 2849-2503
27. Gorschek T, Wohlin C. Requirements abstraction model. Requirements Eng. 2007;11(1):79–101.
28. Kaplan S, Garrik BJ. On the quantitate definition of risk. Risk Anal. 1981;1(1):11–27.
29. Stalling W, Brown L. Computer security, principles and practice. Hoboken: Prentice hall; 2007.
30. http://www.3gpp.org/SON, Accessed 2021–03–09
31. http://www.3gpp.org/ftp/Specs/html-info/25484.htm, Accessed 2021–03–09

32. Yin R. K Case study research: Design and Methods, Fourth Edition, Applied Social Research Methods Series. (2009)
33. Runeson P, Höst M. Guidelines for conducting and reporting case study research in software engineering. J Empirical Softw Eng Springer. 2009;14(2):131–64.
34. Kitzinger J. Qualitative research: introducing focusgroups. BMJ. 1995. https://doi.org/10.1136/bmj.311.7000.299.
35. Polit DF, Tatano Beck C. Generalization in quantitative and qualitative research: myths and strategies. Int J Nurs Stud. 2010;47(11):1451–8.
36. Samuel JB, Marathamuthu MS, Murugaiah U. The use of 5-WHYs technique to eliminate OEE's speed loss in a manufacturing firm. J Qual Maint Eng. 2015;21(4):419–43.
37. Alexander I. Misuse cases: use cases with hostile intent. IEEE Softw. 2003;20(1):58–66.
38. Salehie M, Pasquale L, Omoronyia I, Ali R, and Nuseibeh B Requirements-driven adaptive security: Protecting variable assets at runtime, 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, pp. 111-120. (2012)
39. Chung L, Nixon BA, Yu E, Mylopoulos J. Non-functional requirements in software engineering. Dordrecht: Kluwer Academic Publishers; 2000.
40. Burge J, Brown D (2002) NFRs: fact or fiction? Worcester Polytechnic Institute, Technical Report, WPI-CS-TR-02–01.
41. http://commoncriteriaportal.org. Accessed 2019–03–30
42. Wilander J, Gustavsson J Security requirements - a field study of current practices, E-Proc. The Symposium on Requirements Engineering for Information Security (SREIS), Paris, France. (2005)
43. Haley CB, Laney R, Moffett J, Nuseibeh B. Security requirements engineering: a framework for representation and analysis. IEEE Trans Softw Eng. 2008;34(1):133–53.
44. Apvrille A, Pourzandi M. Security software development by example. IEEE Secur Priv. 2005;3(4):10–7.
45. Boström B, Wäyrynen J, Boden M (2006) Extending XP practices to support security requirements engineering International Workshop on Software Engineering for Secure Systems (SESS). Shanghai, pp. 11–18
46. Souag A, Mazo R, Salinesi C, Comyn-Wattiau I. Using the AMAN-DA method to generate security requirements: a case study in the maritime domain. Requirements Eng. 2018;23(1):557–80.
47. Villamizar H, Kalinowski M, Garcia A, Mendez D. An efficient approach for reviewing security-related aspects in agile requirements specifications of web applications. Requirements Eng. 2020;25:439–68.
48. Vasilevskaya M, Nadjm-Tehrani S Model-based Security Risk Analysis for Networked Embedded Systems, Proc. The 9th International Conference on Critical Information Infrastructures Security (CRITIS) Limassol, Cyprus, pp. 381–386. (2014)
49. Hope P, McGraw G, Anton AI. Misuse and abuse cases: getting past the positive. IEEE Secur Priv. 2004;2(3):90–2.
50. Sindre G, Opdahl A.L Eliciting security requirements with misuse cases, Proc. 37th International Conference on Technology of Object Oriented Languages and Systems (TOOLS-37'00) Sydney, Australia, pp. 120–131. (2000)
51. Mwambe O, Echisen I Security oriented malicious activity diagrams to support information systems security, International conference on advanced information networking and applications workshop Taipei, Taiwan, pp. 74–81. (2017)
52. Behnia A, Abd Rashid R, Chaudhry JA. A survey of information security risk analysis methods. Smart Comput Rev. 2012;2(1):79–64.
53. Souag A, Mazo A, Salinesi C, Comyn-Wattiau I. Reusable knowledge in security requirements engineering: a systematic mapping study. Requirement Eng. 2016;21:251–83.
54. Cruzes D, Gilje Jaatun M, Bernsmed K, Tondel I.A Challenges and experinces with applying Microsoft Threat Modeling in Agile Development Projects, 25th Australasian Software Engineering Conference, Adelaide, Australia, pp. 111–120. (2018)
55. Morrison P, Smith B.H, Williams L Surveying security practice adherence in software development, Proc. of the Hot Topics in Science of Security: Symposium and Bootcamp, ACM International Conference Proceeding Series Part F127186, New York, USA, pp. 85–94. (2017)