Threat modelling and agile software development: Identified practice in four Norwegian organisations

Karin Bernsmed
Software Engineering, Safety and Security
SINTEF Digital
Trondheim, Norway
karin.bernsmed at sintef.no

Martin Gilje Jaatun
Software Engineering, Safety and Security
SINTEF Digital
Trondheim, Norway
martin.g.jaatun at sintef.no

***Abstract*—Threat modelling is considered a key activity in secure software engineering. However, despite its documented benefits it has not (yet) been widely adopted by agile software development projects. In this paper we present results from a qualitative study of how it is performed in practice by four different organisations. The findings show that, even though they all consider threat modelling to lead to a more secure product, they all struggle with practical aspects of the established theory.**

***Index Terms*—threat modelling, software, agile**

I. Introduction

Threat modelling is considered a key activity in secure software engineering. It consists of three main activities [1]:

- Asset identification, which includes identifying information and/or services that are essential or critical for the system;
- Creating an overview over how assets are stored, processed or otherwise interact with the system, which includes systems interfaces and potential attack surfaces. This is usually visualised in a Data Flow Diagram (DFD);
- Identifying threats that will affect one or more of the identified system assets. Threats can be identified through the use of existing frameworks, such as Microsoft STRIDE [2], and further analysed and visualised in, for example, attack trees [3]

These activities are often performed as part of a risk assessment, in which the threats with unacceptable high risk are mitigated through the identification, implementation and deployment of suitable countermeasures in the system.

In agile software development, adoption of security practices in general poses several challenges, often because security activities are not prioritized, or because the practitioners are not able to see the relevance and importance of the activities; i.e., they cannot see how they contribute to the improvement of the security in the project [4]. The same holds true for threat modelling; the practice is not widespread, and the agile practitioners have few sources of recommendations on how to proceed to adopt the practice in their development process. Still, renowned security experts such as Michael Howard & Steve Lipner [5] and Gary McGraw [6] extol threat modelling (also known as architectural risk analysis) as the single most effective activity to improve software security. If we accept this wisdom, we must also accept that it would be a "Good thing™" if more software development organisations made threat modelling a part of their regular software development lifecycle.

The work presented in this paper has been performed in the SoS-Agile project, which is funded by the Research Council of Norway's IKTPLUSS program, grant number 247678.

The following research questions were thus defined for our study:

- What are the main challenges to applying threat modelling in agile software development?
- What are the best practices adopted by agile software development teams?

In this study we collected empirical information by conducting qualitative interviews and analysing the results. The participating organisations in this study are all based in Norway. Their core product is software and they all employ agile methods by their development teams. We found it interesting to examine how and to what degree these organisations perform threat modelling as part of their software development activities and what challenges they face. As will be seen, the participating organisations have quite different organisational structures, which we believed could lead to interesting findings. In particular we were interested in finding out whether there are any best practices in applying threat modelling techniques, which are common to all the four participating organisations, despite their differences.

II. Background

A decade ago, Shostack shared experiences of threat modelling at Microsoft [7]. Shostack takes a holistic (or even ecumenical) approach to threat modelling, stating that there is no right or wrong way to do it. Indeed, there are numerous approaches documented in the literature. McGraw [6] presents an approach that involves four (one plus three) main phases: Establishing a forest-level view of the architecture, determining attack resistance, performing ambiguity analysis, and assessing vulnerabilities of underlying frameworks. Comparing Shostack and McGraw, there may be a perception that the former is more network-centric than the latter, and this impression is reinforced when playing the Elevation of Privilege card game [8].

Assal and Chiasson [9] studied 15 teams of developers in organizations that they classified as either "Security Inattentive" (i.e., not concerned with or aware of security issues) or "Security Adopters" (i.e., working consciously with software security issues). They found that security was often not considered in the design stage, and if considered, it was frequently treated in an ad-hoc manner. Specifically, they noted that in the Security Inattentive teams, threat modelling was triggered if a component was found to handle sensitive information. On the other hand, the Security Adopters follow good security practices including formal threat modelling as part of the security requirements elicitation process. Hence, when Assal and Chiasson [9] list 12 application security best practices, Threat modelling figures prominently, and they also highlight that security should be applied to *all* applications.

Threat modelling in agile software development teams has also been studied, most recently by Galvez and Gurses [10], who analysed challenges with applying threat modelling for privacy in agile teams, and Cruzes et al. [4], who observed the application of threat modelling by five different agile teams in a single company, and opined that threat modeling will lead to fewer emergency patching situations and fewer successful attacks.We aim to add to these studies, but providing additional insights into how agile teams do threat modelling as part of their software development work.

III. Method

In our study we sought to extract in-depth information on how organisations perform threat modelling in practice. We therefore chose a qualitative research method based on relatively few informants, which enabled us to perform a rich and detailed analysis of the selected organisations. Further, since we wanted to derive patterns from our observations, rather than evaluating existing hypotheses, we used an inductive research approach[1].

A. Case studies: Interviews

A case study is an empirical inquiry that investigates a contemporary phenomenon in depth within its real-life context [12], [13]. We performed a multiple case study involving four different organisations, and we used semi-structured interviews as the main source of information.

Interviews are a well-known and powerful tool for information collection in qualitative research. They give the researchers insights into the research topic from the interviewees perspective [14]. We used what is referred to in the literature as semi-structured interviews, which are driven by open questions, have a limited degree of structure, and tend to focus on specific situations and experiences made by the interviewee [15]. The interviews were performed either face-to-face or over Skype. No personal data was recorded during the interviews, and we have taken care not to include any details that could compromise the anonymity of the interviewees or the participating organisations in this paper.

[1]In inductive research the researchers perform field studies followed by deriving theories from observations [11].

B. Industrial context

Organisation A is a supplier of IT and electronic ticketing systems to the public transport sector. They have around 100 employees, whereof the majority are stationed in its Norwegian headquarter. The organisation also operates two offices abroad, where some of the developers are working. We interviewed the Security Champion[2] of one of the development teams stationed in Norway. This particular team is responsible for maintenance of the organisation's existing software product; they rarely develop any new services.

Organisation B is a provider of digital identification and authentication services. The company has around 30 developers. These are stationed in Norway as well as abroad. Three persons participated in the interview; the Chief Information Security Officer (CISO) and two of the security managers.

Organisation C is a small start-up, which offers products and services to help improve the digital security of their customers. We interviewed the person responsible for integrating security in the organisation's development and deployment processes (who referred to himself as the "SecDevOp"), who in addition also is the Chief Executive Officer (CEO) of the company.

Organisation D is a software development company that delivers software and related services to customers in the energy sector all around the world. They have around 20 developers, whereof ten are stationed in Norway. Three persons participated in the inteview: a technical consultant, the Software Development Manager and the Chief Architect (the latter two participated through Skype).

C. Data analysis

For the data analysis we used the general inductive approach described by Thomas [17]. This paper presents a systematic set of procedures for analysing qualitative data and explains a straightforward approach for deriving findings guided by research questions. Inductive analysis is often guided by predefined research objectives. The use of research questions as guidance in data analysis undoubtedly sets constraints on the number of possible interpretations and outcomes as it draws attention to specific aspects of the data. However, using the general inductive approach rather than a stricter and more structured methodology, enabled findings to emerge from themes inherent in the raw data despite the pre-set research questions. Also, by using this approach, findings were not restricted by the method used.

IV. Findings

The findings from the four cases are presented in this section. For each organisation, we have organised the findings under five different themes; 1) What and why, 2) How and when, 3) Time spent and "Definition of Done", 4) Documenting and utilizing the results, and 5) Challenges and benefits.

[2]A security champion is usually a developer with a particular interest in (and aptitude for) security, who in addition to regular developer tasks is given responsibility for helping to discover and solve software security problems during the development [16].

A. Organisation A

1) What and why: The starting point for the threat modelling activities in this organisation is the identification and valuation of assets, which is performed by the development teams for the software that they are currently working on. Based on the identified assets, the team then creates a Data Flow Diagram, which consists of system components, actors and interfaces. To restrict the scope of the analysis, they only consider a limited part of the product at a time. They then use Microsoft STRIDE [2] and OWASP Top 10 [18] to identify relevant threats. The analysis is supplemented by a table that includes relevant threat actors and their motivations. The results from the threat analysis then ends up in a number of risks, for which mitigation measures can be identified. Finally, the risks and the mitigation measures are documented in Confluence (the team's collaboration software).

When asked, the security champion did not have a clear answer to *why* these activities were performed, other than that it was a compulsory activity.

2) How and when: All the threat modelling activities were performed through brainstorming meetings in the development team where everyone was physically present. The Data Flow Diagram (DFD) was created with the Microsoft Threat Modelling Tool[3]. The security champion led the work; initially by himself when identifying and valuating assets. When identifying threats and risks he was assisted by an external consultant.

This particular team had only done the threat modelling activities once. At the time of writing, they are starting a second round, in which they have already reviewed the existing list of assets. They are now planning to improve the DFD created in the first round by adding more details and by narrowing the scope of the analysis to only focus on the most critical assets.

3) Time spent and Definition of Done: The time needed for the threat modelling activities have so far been around 1-2 hours for the actual brainstorming meetings, in addition to approximately 10 hours of preparatory work (identifying assets and drawing DFDs) and 4 hours to follow up on the brainstorming (documenting the risks and mitigation measures). The organisation does not have any specific budget for the threat modelling; for them it is mostly a question of finding available time to work on these activities.

In Organisation A, the Definition of Done is mostly based on gut-feeling. For example, the brainstorming meeting stopped when all parts of STRIDE had been completed and they felt that they had used what the security champion referred to as "sufficient amount of time" on this activity.

4) Documenting and utilizing the results: When asked how the results from the threat modelling activities are taken into the development process, the spontaneous answer was "*They are not*". However, the security champion then explained that all the identified risks were evaluated in terms of how important they are (by ranking them in terms of probability and impact) and how easy they are to address. The risks, and their corresponding mitigation measures were then registered as tasks in Jira (the organisation's bug tracking system). These mitigation measures will then be followed up and implemented by the developers. For each risk that is not to be treated, a note was added that explains why it did not need to be addressed.

5) Challenges and benefits: According to the security champion, the most demanding part of threat modelling was to identify all components and their interactions in the system. He also found it difficult to identify all potential threats and to decide which of these are relevant. Even though he found the OWASP Top 10 list to be helpful, he found it difficult to transforms these general items to concrete vulnerabilities in their system. A major drawback, he said, is that all these activities are time-consuming and that they require creativity and insight. The security champion said he lacked both experience and guidance for arranging these activities. He found it easier to answer questions than to identify relevant threats by himself.

An additional challenge was related to the ambitions[4] and results from the different teams. In Organisation A, these were highly variable; for example, almost 50 percent of all the registered risks originated from a single team and the security champion did not believe that this reflects reality.

On the positive side, the security champion considered threat modelling to be very good for learning purposes, which he thinks is always valuable, even though he questioned whether it in the end would also be valuable for their customers.

When asked, the security champion said he believes that threat modelling leads to a more secure product, at least when the identified risks have been addressed. However, he also said that

> *"If the goal is to have a secure product, we still have a long way to go. But I don't think we will ever be there. The system is so old, and has been in operation for so long, so to be completely secure it needs to be thrown away and started over. This is not realistic. For some components it will cost too much. So we need to accept the situation."*

B. Organisation B

1) What and why: In Organisation B, asset identification is done on the organisational level, starting by analysing the business processes. Each identified asset, which is either information or a service, is then classified and evaluated according to its need for confidentiality, integrity and availability. Since the organisation complies with ISO/IEC 27001 [19], the asset identification and subsequent threat modelling activities are performed in accordance to the processes defined in this standard. According to the Chief Information Security Officer (CISO), complying with the standard helps by structuring the work with information security and it also demonstrates to

[3]https://docs.microsoft.com/en-us/azure/security/azure-security-threat-modeling-tool

[4]According to their experience, ambitious teams tend to identify many more threats, but this does not necessarily mean that there in reality are more threats.

their customers that the organisation is in control. Contractual obligations to some of their customers, as well as compliance with GDPR [20], also mandates compliance with this standard. The CISO thinks that all these activities improves security and that they have a positive impact on product quality.

2) How and when: Asset identification is done by the CISO and the security managers. The organisation maintains a central asset repository, which is available to all employees. The remaining threat modelling activities are performed by the developers. How they do it depends on the individual; many choose to draw use cases, some also create Data Flow Diagrams. They all use Microsoft STRIDE as a baseline for their work. What the developers do, in practice, is the completion of a full risk assessment cycle, which include modelling the system/service and identifying and assessing relevant threats and risks.

In Organisation B it is up to the developers to identify the need for a risk assessment. The organisation has clear guidelines, which state it needs to be done for all new features and major changes in the system[5]. One of the security managers will then do a Quality Assurance (QA) of the results. There are some drawbacks with this approach though, the CISO pointed out, suggesting that the process might be improved by instead letting the Product Owner make the decision on whether a risk assessment needs to be done. Then the developer can take it forward.

In Organisation B, the developers are hence responsible for performing the threat modelling activities. Depending on the team, the developer may do this him/herself, or he/she may involve the rest of the team. The security managers are included in the process as security experts (as well as for QA, as mentioned above). They are usually involved through Skype. According to the security managers, it is however often difficult to work efficiently through Skype, since they do not have access to a whiteboard, which they consider good for explaining ambiguities. They have not tried out any remote whiteboard tools ("*Do you know of any good examples*?"). No external consultancies are employed for the threat modelling activities.

According to Organisation B, the threat modelling activities should start as early as possible in the development process. However, in many cases the need is not identified until the software is about to be deployed. In such cases, it is the deployment team that identifies the need for a risk assessment and the Operations Manager that performs the work.

They have not done any dedicated threat actor analysis, but wish they had.

3) Time spent and Definition of Done: The time needed for the threat modelling activities has so far varied from two hours, which is the "happy scenario" where the developer uses one hour to create the diagrams and another hour to discuss them with the security manager, to up to four months, which was a "worst case" where the introduction of new technology required major changes in the software release process.

[5]According to the security managers, there are new features around four times a year and major changes around ten times a year.

When asked, Organisation B says they have a clear "definition of done"; the work is done when the risks have been accepted. Risks can be accepted when they have been mitigated and the remaining risk is accepted. All risks above a certain threshold need to be accepted by the risk owner (who in most cases is the product owner) and the responsible security manager.

Regarding use cases and Data Flow Diagrams, these are considered "done" when sufficient information has been provided so that everyone involved has a common understanding of what is going on.

4) Documenting and utilizing the results: Regarding documentation, Organisation B does not have any standardized approach. Developers often use whiteboards and regular handwriting when drawing diagrams and identifying threats. Also the way they do threat modelling differs; some draw Data Flow Diagrams, other use cases. As a result, the threat modelling activities are documented in many different formats. The organisation wishes to standardize how to perform and document these activities, but they do not want to enforce specific tools on the developers. As the CISO expressed it:

> *"Developers don't like tools being forced upon them like a straight-jacket."*

Regarding documentation, Organisation B therefore thinks there is potential for improvement. Today they use Jira[6], which links into Git[7]. Ideally, they want to be able to backtrack everything. For example, when errors are detected due to a code change, which was not identified during the risk assessment, the catalogue of threats should be updated. The catalogue is currently an Excel file, but, as the CISO stated, it would be nice to have this kind of overview in a dashboard.

In Organisation B, the threat modelling activities are done as part of the risk assessment that the developers do. The final results from the risk assessment is a set of identified mitigation measures, which are then implemented by the developer. The developer also writes a short summary of the risk assessment, which will be used in the release note of the software. After the implementation of the mitigation measures, the developer generates a "ticket", which includes the assessment of impact and likelihood of the risk(s) that has been mitigated. The ticket should also include information about what was done to mitigate the risk (but some developers skip this part). All generated tickets are transferred to the responsible QA. However, the organisation does not have any way to efficiently track whether the identified risks have been mitigated or not.

A drawback of not having any defined structure for documenting the results of the risk analysis is, the CISO said, that they do not have any mechanisms for reusing previous results.

5) Challenges and benefits: According to the CISO, the main challenge of this approach is to make sure the developers do their risk assessments. They do it when they have to, but they are not always motivated. The CISO suspects that sometimes changes that should have been defined as "major" are

[6]https://www.atlassian.com/software/jira

[7]https://github.com/

actually defined and processed as "minor" by the developers, to avoid having to do the risk assessment. Even though the organisation has guidelines for when and how to do a risk assessment, this is to some extent up to interpretation by the individual developers. Also, a challenge is that the developers believe they can solve security issues without having to go through the risk assessment process. In 70-80% of the cases, the CISO stated, this may be true. However, he believes that clearly defined processes and routines become more and more important and will help in particular novice employees to deliver secure software.

When asked, the CISO agreed that, since the developers are not involved in the asset identification process, it may be unclear whether they understand the value of the information their software use. Even though the central asset repository is available to everyone in the organisation, the developers have no clear relationship to it and the CISO does not know whether it is used in practice. However, he also explained that since the organisation does early awareness training (according to the requirements in ISO/IEC 27001), this will to some degree expose the developers to the importance and value of the organisation's assets. Also, since the developers are including the security managers in the risk assessment process, potential gaps related to this will be followed up. Still, he pointed out:

> *"The developers don't have the business perspective. They probably think this is lost working time, which they could have spent on development."*

In summary, the CISO thinks their way of working with software security works very well. The organisation has around ten new software releases each week and daily tasks, including the risk assessments, usually run smoothly. He believes they have a good security culture in the organisation, even though they all might benefit from raising awareness of why risk assessment is important, especially amongst the new employees. He considers threat modelling to be a good way of raising awareness, in particular since the developers are forced to let go of their "happy day" mindset and instead take a different perspective of the software they work on. He considers threat modelling to be worth the effort; the identified mitigation measures will make it more difficult to misuse the software and hence lead to a more secure product. Even though it takes time, the cost of a security breach is so high.

> *"We sell a security product [. . .] Trust is one of our main selling points. A loss of reputation may lead to that we lose 50% of our customers."*

C. Organisation C

1) What and why: Organisation C does asset identification by defining what they refer to as "key features", which are then prioritized in terms of their importance. Special attention is paid to the use of personal data, for which they do a dedicated Privacy Impact Assessment (PIA). They also create Data Flow Diagrams (DFDs); not for threat modelling purposes, but for internal use when developing and documenting their software. The DFDs are also intended to be used to demonstrate functionality to their customers (even though no-one has asked for it yet). They have not done any formal threat identification or analysis, apart from a brief on-top-of-the-head analysis of who might be interested in the (mis)use of personal data.

A motivational factor for Organisation C for doing threat modelling is potential customers, who often ask for documentation. Still, in their experience, customers are not interested in the threat modelling results as such, unless they result in concrete mitigation measures that will need to be implemented.

Overall, software security is a top priority in Organisation C and they want it to be an integral part of all their activities. Training, good "security culture" and the use of checklists are considered important. Since they deliver security products, they are dependent on their customers' trust and they hence need to think about security at all times.

2) How and when: In Organisation C it is a single person who works with software development. Despite of being a very small business, they have an established process for software development, but in practice it is up to the developer how it is done. The developer relies a lot on his own experience and likes to follow existing guidance and best-practices, even though he thinks many of the advices are contradictory. He engages a lot in the Norwegian software development community. Threat modelling is done when there are substantial changes in the system, for example when a new feature is created or when they start using data from a new source.

3) Time spent and Definition of Done: Even though Organisation C does not have any formally defined threat identification and analysis activities, they claim they do threat modelling continuously, at all times. According to the developer, security is a natural part of his work and his feeling is that the only additional time he needs for doing threat modelling is the time he spends on the extra documentation. For the same reason, the organisation has no specific "definition of done" of these activities.

In parallel to the threat modelling activities, Organisation C is also implementing an Information Security Management System (ISMS) [19] and they feel this takes a lot of time. Since the ISMS mandates that the organisation performs regular risk assessments, and thereby implicitly also do threat modelling, Organisation C pointed out it is difficult to distinguish between this work and the activities that were the topic of the interview, when it comes to time spent on each of them.

4) Documenting and utilizing the results: In Organisation C, the asset (key features) identification and the results from the PIA are documented in Microsoft Word and PowerPoint. The Data Flow Diagrams are created and documented as mind maps. Since they consider threat modelling to be an ever ongoing activity, they state that its results are continuously taken into the software development process.

5) Challenges and benefits: Organisation C think that finding developers with security knowledge is a challenge. They therefore train all new employees themselves. Their goal is that all employees should have a sense of ownership to the software they deliver and that threat modelling should be something that

they do naturally, not as a formal activity but integrated into they way they think and work.

> *"If you ask a headhunter whether he can get you a "DevOp" with security knowledge, he will just laugh.."*

Organisation C agrees that threat modelling will lead to a more secure product, the main reason being because it increases awareness. However, they do not think that going through all its steps systematically is worth the effort. Threat modelling should be a mindset of the developers, rather than a set of formally defined and documented activities. In their opinion, understanding the value of the assets makes you also understand what mitigation measures need to be implemented, and it is hence not necessary to go through all the steps of, for example, STRIDE.

D. Organisation D

1) What and why: Organisation D is ISO/IEC 27001 certified and has hence implemented a systematic approach for risk management, which includes threat modelling activities. The baseline for their threat modelling are Data Flow Diagrams, which they already have created for all their solutions as part of their technical documentation. They then use Microsoft STRIDE to analyse the solutions. Security is a high priority in Organisation D; they have a dedicated security group, which is lead by a "Security Champion", that has developed guidance documents that are used by the developers. When asked about assets, the interviewees were uncertain whether they had any formal overview (inventory) over their assets, but they claimed that assets are in any case always considered implicitly when they perform the threat modelling activities.

2) How and when: According to the interviewees, Organisation D has an officially formulated strategy that they refer to as "security first". At each software release, or major change, they are required to do a "security check". Even though threat modelling should be done at each such checkpoint, it is not always accomplished; the reason being that releases happen so frequently (about every other week). In practice they therefore do threat modelling only when there are major changes. Further, "Security by Design" was also stated as an important principle in all their software development projects. However, the interviewees admitted, they had not yet implemented this principle in practice, the reason being that this was a fairly new principle in their organisation and all of their ongoing projects had already started when it was introduced.

In Organisation D, threat modelling is not only triggered by major changes in their products; it can also be triggered for other reasons. Last year, for example, the organisation performed an additional threat modelling activity when it became publicly known that a security incident had affected one of their competitors operating in the same sector.

When identifying threats, the interviewees said it is preferable if all participants are gathered in the same room. Not only developers attend, but also all other people that are needed to "cover all parts of the application". Regarding tools, they use the Elevation of Privilege card game [8] to identify and model threats. They have also tried the Microsoft Threat Modelling Tool, but found it too cumbersome and time consuming to be used on a regular basis. Also the lack of support for modelling some of the 3rd party components that the organisation uses as part of their services, was considered a barrier for the usage of this particular tool.

The interviewees also pointed out that, apart from the formal threat modelling activities that they trigger at major changes in their software, they continuously think about threats and they always have security in mind when assessing their architectures and their technologies.

3) Time spent and Definition of Done: In Organisation D, the threat modelling sessions usually last 1.5-2 hours. Bigger projects may in theory need more time, but they are usually then split into sub-projects, which are then analysed individually. According to the interviewees, sessions that last longer than two hours are undesirable, since it is difficult for the participants to keep focused for longer periods of time.

> *"Definition of done? You will never be done.."*

The interviewees stated that threat modelling is an activity that will never be completely done; they always find issues that need to be followed up, and their gut feeling is that there is always a need to do threat modelling again afterwards.

4) Documenting and utilizing the results: In Organisation D, the main result from the threat modelling sessions is the "nonconformities" that are discovered in the software. These are then registered as tasks in their issue tracking system and documented in the software backlog. Apart from this, no reports, minutes, or any other type of documentation is produced from the meetings.

5) Challenges and benefits: Organisation D thinks that the most challenging aspect of threat modelling is that most threats are so specific; they often exploit vulnerabilities in other people's code, which one needs to fully understand to know whether the threat is relevant for one's product or not. As an example, the interviewees mentioned the POODLE attack [21], which was a threat that they had identified, but had struggled to understand whether it was affecting them or not. Another challenge pointed out by the interviewees was trying to take the attackers' perspective when analysing threats; what tools and resources do they have? Organisation D had, however, arranged two penetration tests of their infrastructure and services, which they think had helped with this aspect to some degree.

When asked, Organisation D agreed that threat modelling will lead to more secure products and they also said they consider it to be worth the effort. Additional benefits pointed out by the interviewees was that they can more easily demonstrate to customers that they do what they can to deliver a secure product, they avoid incidents and bad publicity, and it could also reduce liability in case of a security breach (c.f. GDPR [20]). Finally, they referred to their ISO/IEC 27001 certification as a major driver for all the security work that they do in the organisation.

V. Observed challenges and best practices

In this section we emphasize and discuss the main challenges and the best practices experienced by the four participating organisations in this study.

A. Challenge: Lack of motivation

In Organisation A, the security champion could at first not explain why they did threat modelling, even though he later stated that it is good for learning purposes and that it eventually may lead to a more secure product. In Organisation B, the lack of motivation amongst some of the developers was illustrated by their tendency to skip it when they have the chance. Organisation C, on the other hand, seemed to be highly motivated, but their motivation was related to having the right "mindset" rather than on going through all the steps of the threat modelling activities. Lack of motivation therefore seems to be common factor appearing in all these three organisations. However, it was not referred to as a challenge at all by any of the interviewees in Organisation D who, in contrast to the other participants in our study, gave the impression that all of their involved employees were motivated to participate in the threat modelling sessions.

B. Challenge: Identifying relevant threats

Identifying relevant threats was a common challenge identified in all the four organisations that we interviewed, even though the reason why differed. In Organisation A, the security champion felt that he lacked experience and that existing guidelines were too shallow. He also considered this a time-consuming activity. In Organisation B, the CISO suspected that relevant threats could be overlooked, because the developers did not understand the value of the information they processed. Organisation C's opinion was that doing threat identification in accordance to an existing method, such as STRIDE, was not worth the effort. Finally, Organisation D often found it hard to know whether, for example, publicly know threats were relevant to them or not.

C. Challenge: Threat modelling is time-consuming

This was reported as a challenge by the first three interviewed organisations. In Organisation A, they found it hard to find time to do these activities, in addition to all their other obligations. Organisation B had a "worst case" example where a lot of time had been spent on these activities. Finally, as mentioned above, Organisation C considered in particular the threat identification and analysis part to be a waste of time. Time was not perceived as a challenge in organisation D though; the reason for this is most likely that they always have already existing DFDs to be used as a baseline and that the amount of documentation that they produce is minimal.

D. Challenge: Knowing the "definition of done"

None of the four organisations had a clear definition of when they were "done". Organisation A stopped when they felt they had spent "sufficient time" (without being able to explain what this was). Organisation B had a clear definition of when the *risk assessment* was done, but not of the individual threat modelling activities that they did during this assessment. Organisation C, on the other hand, claimed they will never be done; their approach is to think about threats constantly, at all times, a mindset that they seem to share with Organisation D.

E. Best practice: Involving the developers

A common success factor, identified in all four of the interviewed organisations, is the involvement of developers in the threat modelling activities. This seems to be a successful approach, regardless of whether the developers were only contributing to the activities (as in Organisation A), or if they did most of the work themselves (as in Organisation B, C and D). However, the initial step of identifying assets were not done by developers in any of these organisations; the common opinion appeared to be that this is an exercise for people higher up the "food chain", who have the business perspective and understand the value of the organisation's assets.

F. Best practice: Using checklists and clearly defined processes and routines

All four organisations advocated the use of checklists, clearly defined processes, routines and the like. Organisation A needed it because they felt they lacked experience to identify and assess all relevant threats themselves. Organisation B said this was particularly useful to their newly employed developers. Organisation C also advocated checklists, however without further specifying why. Organisation D used guidance documents that had been developed by their security group.

G. Best practice: Triggering the threat modelling activities

All four organisations use an agile software development process with short iterations and they all pointed out the benefit of doing threat modelling activities at regular time intervals.

VI. Threats to validity

Interviews have been the main source of data collection in our study. Myers and Newman [14] mention the artificiality of qualitative interviews as a potential challenge, where one interrogates a stranger that does not know or trust you. They state that the lack of trust may cause the interviewee to withhold information that could be of value to the study. As an attempt to mitigate potential trust issues we highlighted the anonymization of all the data collected during our study during the interviews. Myers and Newman also mention the possibility for interviewees to construct knowledge to appear knowledgeable and rational. We tried to mitigate this challenge by giving interviewees enough time to answer questions and by carrying out the interviews as a dialogue rather than as an examination.

Internal validity concerns the data analysis and whether inferences are being made without sufficient evidence present [12], [13]. The interpretation of information depends largely on the analysts background. All members of our research team share a similar background in software security, but differ in experience. Choosing an inductive qualitative

research approach was a way of reducing bias in our results, as we did not aim at proving a specific theory, rather starting our information collection with open minds.

External validity deals with the issue of whether findings from one study can be generalized to other cases [12], [13]. We have interviewed four quite different organisations (although all of them are using an agile software development methodology) and we still found that there are several similarities between them in how they do threat modelling and what challenges they face. This may indicate that the findings may be transferable to other organisations that resemble the case context for this study. However, it may still be reasonable to replicate the study on similar cases in order to collect stronger support for our conclusions.

VII. Concluding remarks

We have studied how threat modelling is performed in practice by four different organisations, which all employ agile software development practices. We examined what activities they do and why, how they do it and when, how much time they spend and how they defined when they are done, how they document and utilize the results, and what challenges and benefits they associate with the threat modelling activities. The findings show that, even though they all consider threat modelling to lead to a more secure product, they all struggle with practical aspects of the established theory. In particular, lack of motivation amongst the employees, the task of identifying relevant threats, the time spent on the threat modelling activities and the difficulty to know the "definition of done" were prominent in most of the cases. On the bright side, our study also revealed that many things worked really well. In particular involving the developers in the threat modelling activities, using checklists and clearly defined processes and routines, and triggering the threat modelling activities at regular time intervals were standing out as best practices that all the four organisations had (at least partly) adopted with good results.

Our study is limited to four organisations, and it would be useful to compare our results with previous similar studies, such as [4], [10], [22], in order to see whether our four cases would support or undermine their results. Such a study is in fact already in the pipeline for our research team during the coming months.

Acknowledgment

This work was supported by the SoS-Agile project: Science of Security in Agile Software Development, funded by the Research Council of Norway (grant number 247678).

References

[1] M. G. Jaatun, K. Bernsmed, D. S. Cruzes, and I. A. Tøndel, "Threat modeling in agile software development," in *Exploring Security in Software Architecture and Design*, M. Felderer and R. Scandariato, Eds. IGI Global, 2019.

[2] A. Shostack, *Threat Modeling: Designing for Security*. Wiley, 2014.

[3] B. Schneier, "Attack trees," *Dr. Dobbs journal*, vol. 24, no. 12, pp. 21–29, 1999.

[4] D. S. Cruzes, M. G. Jaatun, K. Bernsmed, and I. A. Tøndel, "Challenges and experiences with applying Microsoft threat modeling in agile development projects," in *Proc. 25th Australasian Software Engineering Conference (ASWEC)*, Adelaide, Australia, Nov. 2018.

[5] M. Howard and S. Lipner, *The Security Development Lifecycle*. Microsoft Press, 2006.

[6] G. McGraw, *Software Security: Building Security In*. Addison-Wesley, 2006.

[7] A. Shostack, "Experiences threat modeling at Microsoft," in *MODSEC@ MoDELS*, 2008.

[8] ——, "Elevation of privilege: Drawing developers into threat modeling," in *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[9] H. Assal and S. Chiasson, "Security in the software development lifecycle," in *Fourteenth Symposium on Usable Privacy and Security (SOUPS) 2018)*, 2018, pp. 281–296.

[10] R. Galvez and S. Gurses, "The odyssey: Modeling privacy threats in a brave new world," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2018, pp. 87–94.

[11] B. J. Oates, *Researching Information Systems and Computing*. Sage Publications Limited, 2005.

[12] R. K. Yin, *Case Study Research Design and Methods*, 4th ed. SAGE Publications, 2009.

[13] C. Robson, *Real world research*, 3rd ed. John Wiley & Sons Ltd., 2011.

[14] M. Myers and M. Newman, "The qualitative interview in IS research: Examining the craft," *Information and Organization*, vol. 17, pp. 2–26, 12 2007.

[15] C. Cassell and G. Symon, *Essential Guide to Qualitative Methods in Organizational Research*. Sage Publications Limited, 2004.

[16] SAFEcode, "Software security takes a champion," 2019. [Online]. Available: http://safecode.org/wp-content/uploads/2019/02/Security-Champions-2019-.pdf

[17] D. R. Thomas, "A general inductive approach for analyzing qualitative evaluation data," *American journal of evaluation*, vol. 27, no. 2, pp. 237–246, 2006.

[18] OWASP Foundation, "OWASP top 10 - 2013," 2013. [Online]. Available: https://www.owasp.org

[19] "ISO/IEC 27001:2013 Information technology – Security techniques – Information security management systems – Requirements," Tech. Rep., 2013.

[20] EU, "REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, no. 119, 2016.

[21] B. Møller, T. Duong, and K. Kotowicz, "This POODLE Bites: Exploiting The SSL 3.0 Fallback," 2014. [Online]. Available: https://www.openssl.org/~bodo/ssl-poodle.pdf

[22] K. Tuma, G. Calikli, and R. Scandariato, "Threat analysis of software systems: A systematic literature review," *Journal of Systems and Software*, vol. 144, 06 2018.