Using Situational and Narrative Analysis for Investigating the Messiness of Software Security

Inger Anne Tøndel
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
inger.anne.tondel@ntnu.no

Daniela Soares Cruzes
SINTEF Digital
Trondheim, Norway
daniela.s.cruzes@sintef.no

Martin Gilje Jaatun
SINTEF Digital
Trondheim, Norway
martin.g.jaatun@sintef.no

ABSTRACT

Background: Software engineering work and its context often has characteristics of what in social science is termed 'messy'; it has ephemeral and irregular qualities. This puts high demands on researchers doing inquiry and analysis. *Aims:* This paper aims to show what a combination of situational analysis (SA) and narrative analysis (NA) can bring to qualitative software engineering research, and in particular for situations characterised by mess. *Method*: SA and NA were applied to a case study on software security. *Results:* We found that these analysis methods helped us gain new insights and understandings and a broader perspective of the situation we are studying. Additionally, the methods helped collaboration in the analysis. *Conclusion:* We recommend applying and studying these and similar combinations of analysis approaches further.

CCS CONCEPTS

• **General and reference → Empirical studies**; • **Security and privacy → Software security engineering**.

KEYWORDS

qualitative studies, situational analysis, narrative analysis, software security, agile software development

ACM Reference Format:
Inger Anne Tøndel, Daniela Soares Cruzes, and Martin Gilje Jaatun. 2020. Using Situational and Narrative Analysis for Investigating the Messiness of Software Security. In *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (ESEM '20), October 8–9, 2020, Bari, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3382494.3422162

1 INTRODUCTION

Though new approaches to programming have brought software development more towards the masses, software engineering is still in many cases highly complex both in terms of the software that is developed and the situation in which the development occurs. In this paper we are concerned with understanding these more complex software engineering endeavours, and we use software security as an example of a software engineering practice that is, with current knowledge, difficult to fully analyse and predict.

Software security is certainly a complex topic. Many software security practices are available and many risk factors are known. Still, projects commonly lack the holistic understanding necessary to know with any certainty the best way of action [17]. To secure a system, simply adding security mechanisms like encryption and access control is not enough. One needs to have an overview of the whole system, and based on that be able to identify and address the weak spots. In analysing a system, it is not enough to only consider its technical realization. Empirical studies emphasise the importance of context variables such as the security posture and competence of the development team and of the customer, the relation to and involvement of the customer, the organisational culture and setup, the business case for security and the development approach [16, 18]. There is an interplay between security and insecurity [8], exemplified by the back and forth between detecting and fixing vulnerabilities and the battle between attackers and defenders. Taking care of security is often not a task in itself, but something that arises in proximity to the need, outside of formal responsibility [8]. Additionally, the security work as well as the security of the software product is largely invisible [8, 17] thus it can be hard to pin-point and study directly.

The above description of software security is difficult to put into a tidy process chart. It is better characterised by what Law [10] describes using the term 'mess'. His concern, coming from research within social science, is that though "some things in the world can be made clear and definite" [10] other parts are difficult to capture that way. Law does not offer a clear definition of mess, but rather describes mess using a variety of terms, including "vague, diffuse or unspecific, slippery, emotional, ephemeral, elusive, indistinct, changes like a kaleidoscope, or doesn't really have much pattern at all" [10]. A basic claim is that "simple clear descriptions don't work if what they are describing is not itself very coherent. The very attempt to be clear simply increases the mess" [10]. This does not mean that situations characterised by mess cannot be studied in order to achieve greater understanding. Rather it calls for changes in our ways of doing inquiry. One such change is to allow greater heterogeneity and variation in methods. Another is the need to allow more uncertainty and less coherence in the results, without taking this as necessarily a methodological weakness [10].

In this paper we are concerned with analysis of qualitative data from studying such messy topics as software security. Selecting an analysis strategy is an important part of a qualitative research design [12] and one that can end up influencing what the researchers end up seeing in the collected data [9]. According to Maxwell [12],

ESEM '20, October 8–9, 2020, Bari, Italy

ACM ISBN 978-1-4503-7580-1/20/10…$15.00
https://doi.org/10.1145/3382494.3422162

the main qualitative analysis options can be characterized as either categorizing or connecting strategies. *Categorizing strategies* are commonly understood as coding, where data is arranged into categories to organise the data and aid development of theoretical concepts. *Connecting strategies* are concerned with identifying relationships and thus need to understand the data in context and look at the data more in terms of a coherent whole. In many cases, both are needed and are done in an interplay.

We claim that to do justice to messy topics such as software engineering, there is a need for researchers to emphasize the connecting strategies in their analysis work. Based on the literature on analysis published for the empirical software engineering community, as collected by Molleri et al. [13], we would however claim that the current support on connecting strategies is limited. Most prominent are thematic analysis and grounded theory [13], but these are just a few of the analysis options available [9]. Though the attention these methods receive is merited, additional strategies that e.g. are visual may add to existing methods and improve the fields' ability to take mess into account in analysis. Though the importance of context for software engineering is highly recognized [1, 4, 14], there is a need to move beyond describing context to including context as part of what is studied, as it shapes the whole situation of which software engineering is a part. We here build on Clarke [2] who claims that "*the conditions of the situation are in the situation.* There is no such thing as "context. " [...] everything *in* the situation *both constitutes and affects* most everything else in the situation in some way(s)" [2]. Still, there is a need to dig deep on particular aspects of the situation to achieve the deep understanding that is called for. One such avenue for digging deep is to consider the stories and the sensemaking of key individuals.

The aim of this emerging results paper is to show how Situational Analysis (SA) [3] and Narrative Analysis (NA) [15] can be used in combination when the goal is to understand a situation characterised by mess, with an emphasis on the people involved and their sense making. Together they allow the analysis of whole situations while going deep into the stories of key players. Both SA and NA allow for messiness in that they do not necessarily seek easy and clear answers, while still aiming to provide knowledge about the world. They do so in a form that invites collaboration in analysis and discussion of analysis findings in a community of different experts - something that is often needed in messy topics such as software security where technical, human and organisational factors are all consequential.

This paper is organised as follows. Section 2 gives an introduction to SA and NA before Section 3 shows how these methods can be used in combination on a practical case. Section 4 discusses the experiences so far from applying this analysis approach and points to areas where more understanding is needed. Section 5 concludes the paper.

2 SITUATIONAL ANALYSIS AND NARRATIVE ANALYSIS

Situational analysis (SA) [3] has been claimed to be able to address the messiness of both the empirical world and of conducting research [3]. It uses the "situation of inquiry itself" as the key unit of analysis [3]. The situation is analysed by constructing and working with three different types of visual maps: situational maps, social worlds/arenas maps, and positional maps. These visual representations help get an overview of the full situation and facilitate collaboration in analysis. The maps can be created at an early stage of research to guide the data collection and sampling, and they can be created and/or improved upon in analysis as the study progresses. The method is "especially useful for multi-site research where several different kinds of data are gathered" [3].

Situational maps lay out all the potential elements of a situation, including both human and non-human elements. Two types of situational maps are created; messy situational maps and ordered situational maps. Messy maps are created without any kind of structure, while ordered maps use an organized list of element types [2] as a basis for sorting the elements of a situation. Based on the situational map, the analyst can perform relational analysis, identifying all relations among elements. A main benefit of situational maps is that they work as "holding devices" and "reminder devices", helping analysts to remember the full list of elements and see relations between elements [3]. *Social worlds/arenas maps* interpret the broader situation and "help researchers think about the kinds of collective, organizational, and institutional elements in their projects, too often ignored in qualitative inquiry" [3]. *Positional maps* "lay out the major positions taken, and not taken, in the data vis-à-vis particular axes of variation and difference, focus, and controversy found in the situation of concern" [3]. A main benefit of positional maps it that they untie positions from individuals and groups, thus helping see the discourse in new ways, especially identifying missing/muted/silent positions in the data.

Narrative analysis (NA) [15] is an analysis strategy that uses narratives as the unit of analysis, as opposed to fragmenting the narratives into thematic categories. A narrative can be understood in different ways. In lay terms it can be understood as a story. Narratives are useful when aiming to understand human experience. Due to its focus on details, NA is not suitable for large samples, but should rather be used to analyse selected narratives in close detail.

NA can be done in varying ways; it can focus on "what" is said, "how" it is said, "to whom" it is said, "when" it is said and/or "for what purpose" it is said. *Thematic analysis* focuses on the content of the narrative, the "what." It often uses prior theory to interpret the case. *Structural analysis* looks at how narratives are organised or "put together," e.g. by identifying sequences and structural parts in a story and seeing how these recur across stories. *Dialogic or performance analysis* is concerned primarily with the production and performance of the narrative, including its purpose, who it is directed towards, when and why. With this type of analysis approach a lot of emphasis is put on the context in which the narrative is constructed, and the context is actively used to interpret the narrative. This includes the role of the investigator in the creation and analysis of the narrative.

3 EXPERIENCES FROM A CASE STUDY

To explore the combined use of SA and NA, we use data from one of our own case studies. This case study was descriptive in nature, with the aim to improve understanding of security requirements work in agile software development projects. One development project was selected as case. The data collection activities in this case study spanned one year and nine months and was done by one researcher. It consisted of active participation in initial meetings to elicit security concerns for this project, observation of meetings throughout the project where security was a main topic, access to security requirements documentation at various stages in the project, interviews with a security champion, product owner and technical product owner about two thirds out in the project (recorded and transcribed) and regular talks with the security officer of the company. The case study thus resulted in collection of varied types of data. As the interviews was inspired in part by the use of war stories [11] to extract data, the collected data included stories about how individual security requirements was handled in the project.

The analysis examples presented below are not based on a full analysis of the data from this case. Thus, any analysis examples should be considered early versions to be extended upon and improved in future analysis of the data collected from the case.

The situational maps have been used to get and keep an overview of all the elements that could potentially be studied in the situation under study. Figure 1 shows an early version of an ordered situational map. The ordered situational map was created based on two rounds of creating a messy situational map and based on categories used by Clarke [2] and Fosket [6]. Using these previously established categories was useful to broaden our thinking and identify more elements of the situation. In this ordered map the categories were however modified to fit the situation under study. We experienced that the messy and the ordered process helped us see different aspects of the situation, and in the end quite a lot of elements were identified (as shown in Figure 1). This illustrates well the complexity and messiness of the situation under study. Additionally, it fosters awareness of which elements we cover in our study, and which we do not cover.

Based on the situational map, the next step in SA is to do relational analysis. In this step every element in the map is considered in relation to every other element, drawing arrows and exploring relationships. We have experimented with this step by looking at the relations of two of the elements we identified: the security champion and the product owner. From our experimentation we see that relational analysis can help us see more clearly how these two roles operate in different ways. We also believe it can help us see what elements are central to the situation. However, the high number of identified elements at this point makes doing a full relational analysis a big effort. Others have dealt with this challenge e.g. by selecting a focus of analysis and doing relational analysis related to that focus only [7].

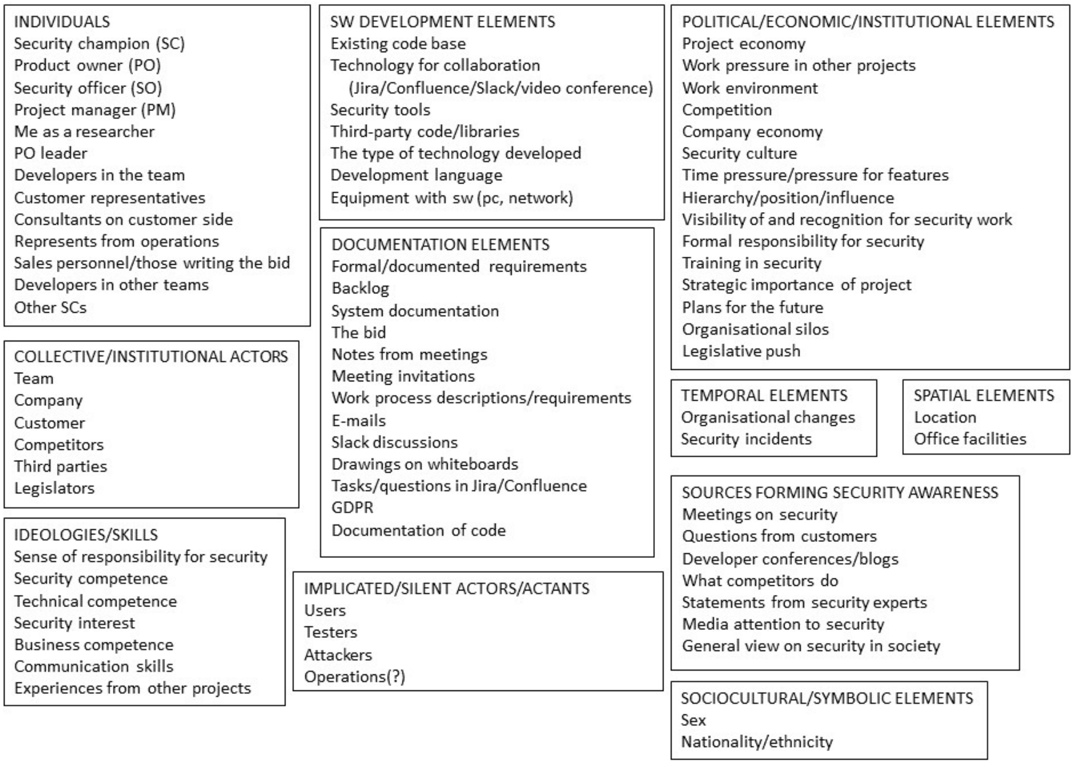
Figure 2 shows an early versions of a social worlds/arenas map. The centre of the map is the topic of research, in this case making security priorities in a project either in an indirect or in a structured way. In a social worlds/arenas map, the lines around the social worlds/arenas are dotted as the boundaries of a social world/arena is quite porous; individuals commonly go in and out of social worlds many times during a day [3]. The size of the circles represents their relative importance related to the common concern in the middle. Those worlds/arenas overlapping with the common concern represent a direct involvement. Overlap of social worlds/arenas represent overlap in the form of individuals being part of both worlds/arenas.

Working on a social worlds/arenas map can prompt researchers to identify what social groups are central to the situation compared to others, and question why that may be the case. Additionally, one may look at how the common concern is shaped by the social worlds or consider how the common concern is understood differently in the different social words [3]. We have, as an initial step and inspired by Fosket [6], looked at the role each of the social worlds play related to the common concern, and highlighted their contribution and what is at stake for them. Other questions may be asked based on the social worlds/arenas map at later stages, e.g. how security requirements and security priorities are understood, made and used in different social worlds.

Our exploration of the contributions of the different social worlds and what is at stake for them made us aware that having a lot at stake does not necessarily mean more involvement. The developers, project management and security resources in the development company have a large impact on the security priorities, but little at stake compared to other actors. The customer, the users or the development company's reputation will be more at stake if there is a security breach. Even operations that must respond to the incident will likely have more at stake than many of the developers involved. What is at stake for many of the key players is likely to be more along the lines of professional integrity and being recognised for achievements.

Narratives in the data material can be used to explore this further and dig deeper into the way key individuals, e.g. the security champion, view their role in the security work. To illustrate we use the following narrative that came as a response to a question intended to prompt narratives about how security requirements (in a broad sense) were handled in this project. Note that the narrative has been translated and shortened for presentation purposes.

> Researcher (R): If you think about one important security requirement in this project, what made you identify it and how was it handled?
> Security Champion (SC): [...] I do not remember if this was mentioned specifically in the requirements from the customer. It was more something that we had thought about ourselves based on the requirements from the customer. [. . .] The customer had mentioned some things that could be done related to this [provides example]. Based on this we have thought that some of the things the customer has suggested are neither easy to do nor helps that much with security. [. . .] Based on our understanding [...] the easiest solution is encryption. [...]
> R: So, you got requirements from the customer that were maybe not that meaningful for you but found another way to solve this. Was this something that you initiated, or did it come from others, do you remember how it came up?
> SC: I do not know if I want to claim I initiated. I have been a driving force. And in those parts of the project that I have coded on, then I have implemented and initiated it. But it is also something I have discussed with the other developers and that they have agreed on and they could suggest solutions to improve what I suggested. [...] For this particular requirement it was something I thought about, partly because I have wanted to do it in previous projects and thought [based on customer requirements in this project] that this was an opportunity to go through with it. Then I brought it up with the other developers, and those I talked with agreed.
> R: You did this on slack?
> SC: Yes
> R: Then you just agreed to do it like this. What happens then?



INDIVIDUALS
Security champion (SC)
Product owner (PO)
Security officer (SO)
Project manager (PM)
Me as a researcher
PO leader
Developers in the team
Customer representatives
Consultants on customer side
Represents from operations
Sales personnel/those writing the bid
Developers in other teams
Other SCs

COLLECTIVE/INSTITUTIONAL ACTORS
Team
Company
Customer
Competitors
Third parties
Legislators

IDEOLOGIES/SKILLS
Sense of responsibility for security
Security competence
Technical competence
Security interest
Business competence
Communication skills
Experiences from other projects

SW DEVELOPMENT ELEMENTS
Existing code base
Technology for collaboration
(Jira/Confluence/Slack/video conference)
Security tools
Third-party code/libraries
The type of technology developed
Development language
Equipment with sw (pc, network)

DOCUMENTATION ELEMENTS
Formal/documented requirements
Backlog
System documentation
The bid
Notes from meetings
Meeting invitations
Work process descriptions/requirements
E-mails
Slack discussions
Drawings on whiteboards
Tasks/questions in Jira/Confluence
GDPR
Documentation of code

IMPLICATED/SILENT ACTORS/ACTANTS
Users
Testers
Attackers
Operations(?)

POLITICAL/ECONOMIC/INSTITUTIONAL ELEMENTS
Project economy
Work pressure in other projects
Work environment
Competition
Company economy
Security culture
Time pressure/pressure for features
Hierarchy/position/influence
Visibility of and recognition for security work
Formal responsibility for security
Training in security
Strategic importance of project
Plans for the future
Organisational silos
Legislative push

TEMPORAL ELEMENTS
Organisational changes
Security incidents

SPATIAL ELEMENTS
Location
Office facilities

SOURCES FORMING SECURITY AWARENESS
Meetings on security
Questions from customers
Developer conferences/blogs
What competitors do
Statements from security experts
Media attention to security
General view on security in society

SOCIOCULTURAL/SYMBOLIC ELEMENTS
Sex
Nationality/ethnicity

Figure 1: Example ordered situational map

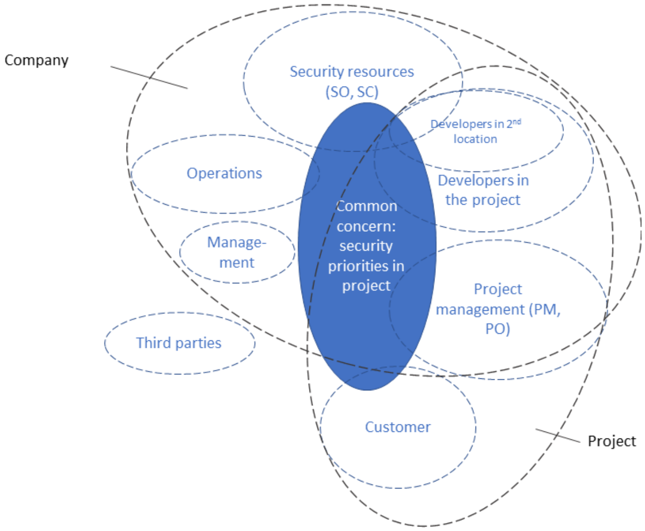

Company
Security resources (SO, SC)
Developers in 2nd location
Operations
Developers in the project
Common concern: security priorities in project
Manage-ment
Project management (PM, PO)
Third parties
Customer
Project

Figure 2: Example social worlds/arenas map

SC: Not much more happened other than we did it.
R: Then it is you developers that in a way can decide to go for this solution, or do you need to involve the customer?
SC: No, in this case it was just developers. And that is something I have thought about that some of the problems with that, as we talked about previously, the security priority is lower in the middle of the project, is that this is something that gives good security but it is probably only developers that know it has been done. It is not documented anywhere [. . .]
R: When did this happen in the project?
SC: For the part I am involved in it happened early [. . .] Do not remember exactly, but it took some time before we made them do the same in the other part of the system.
R: But still, first half of the project then.
SC: Yes. And then we have discussed much later in the project that what we made encryption for was all data at rest. Later we have remembered [other place] that is not encrypted by default. So, we have discovered later that we did a good job with the parts we thought about, but there are parts we have not thought about. This is something we have not implemented yet.
R: Then you discovered a bit late, not so easy to fix. Maybe expensive?
SC: Then there is prioritizing. We need to fix bugs. Adding a lot of encryption can introduce new bugs. Not sure, we may find time to add it in parts of the system, but uncertain.

Moving back to the social worlds/arenas map, one can see that this narrative from the security champion gives insight into the experience of one key individual in this project, and this individual's participation in two social worlds: the developers and the security resources. As a security resource, the security champion shows ongoing attention to security throughout different states of in-/security. The timeline includes seeing this issue in previous projects, getting the opportunity in this project, trying to spread the security solution to other parts of the system, and finding out the limitations and considering what to do about it. Still, the influence of the security champion has its limits. It is strongest in the part of the project where this person is an active contributor to the coding, whereas it is more of a challenge (though not impossible) to bring the same security solutions to other parts of the project.

Although this narrative shows the importance of individuals such as this security champion, security as a shared responsibility is important in the narrative, although primarily among developers. The security champion, though it seems this person did the initiation of considering this security requirement, wants to show the team effort to discuss and arrive at a decision regarding what to do about security. Additionally, the narrative grapples with the responsibilities of the customer vs. the responsibilities of the developers and illustrates how they are part of different social worlds. The customer is responsible for providing the requirements, but the customer and the developers have different competencies and thus different abilities to identify security needs and solutions.

Potential influences that have been found important in previous studies (e.g. the "Business case for security" and "Organizational culture and setup" [18]) is completely missing from the narrative. The role of the product owner, management, budget and time pressure is likely to have an impact, but they are not part of this narrative (except from a small hint regarding time towards the end). The narrative additionally highlights the challenge of security becoming invisible and, in particular, undocumented; nobody except the developers knows it is there. Viewing the narrative in relation to the situation as a whole can bring out questions like, in this case, what does the lack of references to the product owner convey? Being able to ask questions about what is missing is an avenue for understanding the situation in a deeper way, in addition to just looking at what is explicitly included in the data material [3, 15].

Positional maps help flash out possible positions and discourses that actors may have. To get at potential axes for a positional map, we started by identifying potential positions and discourses related to security. Topics from the narrative, such as the responsibility for security and the motivation for security were examples where there were different positions that could be taken and where we explored positional maps (not included for space limitations). The process of coming up with potential axes and experimenting with positional maps was challenging but felt rewarding in that it made us more aware of the many ways people approach security and talk about security. Reflecting on discourses related to responsibility and motivation, in addition to awareness of what is "at stake" for different social worlds (ref. the social worlds/arenas map), made us consider questions like "Is the main message communicated related to security able to motivate those that are not already on board?" Additionally, it makes one able to see what positions are present and not present in the data.

4 DISCUSSION

Our experiences from using a combination of Situational and Narrative Analysis (SA and NA) on this case were positive. We found that the strengths and weaknesses of SA and NA largely complemented each other. The maps of SA provided tools for thinking about the situation as a whole, without demanding a "clean" depiction of reality. Without the overview given by SA, we believe it would have been more difficult to identify questions related to the importance of what is at stake for different types of actors. To explore these questions there is however a need for both overview and depth. NA supported the analysis in going deeper on individuals' sensemaking and understand better the potential role of individuals and their social worlds. Used in combination, SA and NA helped us identify what we perceived as an absence of central actors in key narratives. We see a great potential for using the maps of SA to inform NA, and also to use NA as an input to SA.

Partly, the benefits with using SA and NA were related to the messy properties of software security engineering, e.g. the coming and going of security attention, the dispersed responsibilities, and its invisibility [8]. Narratives and maps were two ways of making security and the work associated with it visible and easier to study. The maps and the narrative allowed the messiness of the situation to shine through and become clearer, something that is important because this messiness is part of what we want to understand deeper. The increased awareness of the many possible viewpoints and narratives about security opened our eyes to interesting avenues for further inquiry, and it made the limitations and contributions of our own study clearer.

Analysis efforts using SA and NA do not aim at generalisation. Instead, SA and NA offer concrete ways of presenting rich descriptions of the situation studied, something that will help readers assess the relevance of the study and its result to their context. With the situational maps, SA brings what is commonly viewed as context into the study from the beginning, with the aim to include this as part of what is studied. There is however a blurred line between just describing context and including context in the study, as is the aim of SA. This points to a main challenge: the lack of a stopping criteria for the analysis. The aim to capture and study the whole situation can be overwhelming and is generally not possible. Taking a broad view of the situation brings to the forefront all the additional relations one could choose to dig into. One should expect to make several tries on the maps, and ideally the maps should be refined throughout data collection and analysis in order to reflect the deeper understanding achieved. In practice there is however a need to stop somewhere, and guidance in making such decisions is needed. Such guidance could e.g. be to continue describing the situation until one does not see more things to describe (saturation), but in reality this may not always be possible due to limited resources. Then the maps of SA can be used to clarify the limitations of the study.

It is important to note that SA and NA and their combination is only one of many ways to approach analysis of qualitative data. It is out of the scope of this paper to make a comparison of SA and NA with other analysis techniques applied in the software engineering field, though we would welcome more contributions on helping qualitative researchers choose analysis approaches, e.g. as is offered by Langley [9] for process research. It is well established in the empirical software engineering field that the research method chosen matters, and there are support in deciding upon a research method [5]. The analysis approach selected can however further influence what is seen in the data, and should be a conscious choice [9, 10]. We experienced that drawing maps and looking at narratives as a whole made us ask different questions of the data material than what we probably would have done with coding alone or from looking at patterns across data sets. Thus NA and SA complement other analysis approaches commonly suggested in the field, such as grounded theory and thematic analysis [13]. In addition, SA may be combined with other analysis methods than NA, e.g. when narratives are not collected. Note however that to gain benefits from the combination of methods, there is a need to combine methods that complement each other in similar ways as SA and NA does.

By presenting our initial efforts and experiences with SA and NA we aim to make the combination of SA and NA more accessible to the software engineering research community. Additionally, our aim is to bring the notion of 'mess' [10] to the community in order to bring additional metaphors that can help us see different aspects of our field of study and motivate more variety in the methods selected. We expect that there are other fruitful combinations of methods that could be useful in other projects, and would in general encourage more experimentation in adopting analysis methods from other fields, and in trying out more combinations of analysis methods.

As for limitations of the preliminary results presented in this paper, we would like to mention that the analysis presented in this paper has mainly been done by one researcher. In this particular project, we wanted a way to discuss intermediate analysis results among experts without all experts having the ability to invest a large amount of time and effort on analysis and both SA and NA helped in that respect. The maps and the narrative has been discussed with other researcher to inform new iterations. The analysis was mainly done with the reliance on pen and paper that is recommended for SA [3], and this worked well. In other projects with deeper collaboration on analysis there will likely be a need for tool support. More work is needed to understand the potential implications of adding tool support and make recommendations.

For researchers wanting to use SA and NA in their analysis efforts, we would point them to Clarke et al. [3] for SA and Riessman [15] for NA. Both resources provide a practical introduction to the analysis method and include or point to example studies that have used these types of methods, allowing readers to see examples of how output from SA and NA can be reported in e.g. papers.

5 CONCLUSION

This emerging results paper introduces the combination of Situational Analysis (SA) and Narrative Analysis (NA) as an analysis approach suitable for studies that aim to understand software engineering situations that are characterised by mess, and where individuals' sense making is part of what one wants to study. SA and NA are applied to a study of software security. In that study we experienced that SA and NA were a good match with our case. This is likely because we study one case in depth, thus being concerned with understanding one situation, and this case has characteristics of being messy. We had collected narratives from key actors through semi-structured interviews, making NA an option. One strength of SA is its ability to help researchers work with different types of data, and this is a need that we had. The combination of SA and NA helped open up for new insights and ideas for further inquiry, it allowed taking a broader perspective of the situation while digging deep on the stories of key individuals, and it supported collaboration and interdisciplinarity in analysis.

As further implications for the empirical software engineering community, we conclude from our preliminary work that more experience is needed in order to provide guidelines for when and how to apply SA and NA, and when other approaches would be more useful. Future work includes applying these methods in analysis efforts of different types to collect additional experiences to use as a basis for guidelines for this field.

ACKNOWLEDGMENTS

This work was supported by the SoS-Agile: Science of Security in Agile Software Development project, funded by the Research Council of Norway (grant number 247678, https://www.sintef.no/en/digital/sos-agile/). Thanks to Prof. Guttorm Sindre and colleagues at "Fabrikken" for commenting on this paper at various stages. Thanks to the study participants for participating in this research.

REFERENCES

[1] Deepika Badampudi, Claes Wohlin, and Tony Gorschek. 2019. Contextualizing research evidence through knowledge translation in software engineering. In *Proceedings of the Evaluation and Assessment on Software Engineering*. 306–311.
[2] Adele E Clarke. 2016. From Grounded Theory to Situational Analysis: What's New? Why? How? In *Situational analysis in practice: Mapping research with grounded theory*, Adele E Clarke, Carrie Friese, and Rachel Washburn (Eds.). Routledge, 84–118.
[3] Adele E Clarke, Carrie Friese, and Rachel Washburn. 2016. *Situational analysis in practice: Mapping research with grounded theory.* Routledge.
[4] Tore Dybå, Dag IK Sjøberg, and Daniela S Cruzes. 2012. What works for whom, where, when, and why? On the role of context in empirical software engineering. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement.* 19–28.
[5] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer, 285–311.
[6] Jennifer R Fosket. 2016. Situating knowledge. In *Situational analysis in practice. Mapping research with grounded theory*, Adele E Clarke, Carrie Friese, and Rachel Washburn (Eds.). Routledge, 195–215.
[7] Marilou Gagnon, Jean Daniel Jacob, and Dave Holmes. 2016. Governing through (in) security: a critical analysis of a fear-based public health campaign. In *Situational analysis in practice: Mapping research with grounded theory*, Adele E Clarke, Carrie Friese, and Rachel Washburn (Eds.). Routledge, 270–284.
[8] Laura Kocksch, Matthias Korn, Andreas Poller, and Susann Wagenknecht. 2018. Caring for IT Security: Accountabilities, Moralities, and Oscillations in IT Security Practices. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–20.
[9] Ann Langley. 1999. Strategies for theorizing from process data. *Academy of Management review* 24, 4 (1999), 691–710.
[10] John Law. 2004. *After method: Mess in social science research.* Routledge.
[11] Wayne G Lutters and Carolyn B Seaman. 2007. Revealing actual documentation usage in software maintenance through war stories. *Information and Software Technology* 49, 6 (2007), 576–587.
[12] Joseph A Maxwell. 2012. *Qualitative research design: An interactive approach.* Vol. 41. Sage publications.
[13] Jefferson Seide Molléri, Kai Petersen, and Emilia Mendes. 2019. CERSE - Catalog for empirical research in software engineering: A systematic mapping study. *Information and Software Technology* 105 (2019), 117–149.
[14] Kai Petersen and Claes Wohlin. 2009. Context in industrial software engineering research. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement.* IEEE, 401–404.
[15] Catherine Kohler Riessman. 2008. *Narrative methods for the human sciences.* Sage.
[16] Evenynke Terpstra, Maya Daneva, and Chong Wang. 2017. Agile Practitioners' Understanding of Security Requirements: Insights from a Grounded Theory Analysis. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW).* IEEE, 439–442.
[17] Inger Anne Tøndel, Daniela Soares Cruzes, and Martin Gilje Jaatun. 2020. Achieving" Good Enough" Software Security: The Role of Objectivity. In *Proceedings of the Evaluation and Assessment in Software Engineering.* 360–365.
[18] Inger Anne Tøndel and Martin Gilje Jaatun. 2020. Towards a Conceptual Framework for Security Requirements Work in Agile Software Development. *International Journal of Systems and Software Security and Protection (IJSSSP)* 11, 1 (2020), 33–62.