Towards Identification of Privacy Requirements with Systems Thinking

Tuisku Sarrala[1(✉)], Tommi Mikkonen[1], Anh Nguyen Duc[2], and Pekka Abrahamsson[1]

[1] University of Jyväskylä, PO Box 35, 40014 Jyväskylä, Finland
tuisku.rad.sarrala@student.jyu.fi,
{tommi.j.mikkonen,pekka.abrahamsson}@jyu.fi
[2] Department of Business and IT, University of South-Eastern Norway, PO Box 4, 3199 Borre, Norway
Anh.Nguyen.duc@usn.no

Abstract. Implementing privacy as software functions is required by privacy regulation. Achieving this requires shared understanding between business process owners and software engineers, who implement it. Current literature reveals a major gap between privacy requirements and how engineers interpret privacy. Furthermore, as today's sociotechnical systems are increasingly complex and ever-evolving, unknown privacy issues can emerge from them as a side-effect. Understanding privacy and identifying privacy threats are pre-requisites for deciding on and implementing the right functionality in software. However, current methods for privacy threat identification do not cover all aspects of privacy, suit complex sociotechnical systems or requirements engineering, or support engineers forming a mental model of privacy. We claim that this situation can be improved by applying a systems thinking approach to privacy threat identification. In this paper, we elaborate the problem and propose a research agenda that will help close the gap between privacy requirements and technical software functionality.

Keywords: Privacy by design · Privacy engineering · Privacy threat modelling · Privacy mental model · Systems thinking

1 Introduction

Privacy is a common concern for today's businesses as almost all businesses have to deal with personal data at some scale. Although privacy is a public value, in many contexts it is not a matter of choice to implement privacy in software, but a necessity imposed by laws and regulation (e.g. [1]). How to satisfy legal requirements when designing software (software being the technical manifestation of the business processes that handle personal data) is a nontrivial question for both research and practice. Today's business processes are complex, software is complex, and business process owners and software engineers lack shared

B. Shishkov (Ed.): BMSD 2022, LNBIP 453, pp. 249–258, 2022.
https://doi.org/10.1007/978-3-031-11510-3_16

understanding and cooperation around operationalising privacy in the technical software functionalities. Both as a cause and an effect of the struggles, privacy is often seen as an afterthought in software development projects [2].

Not only business process owners need to understand privacy in relation to the business processes, but also software engineers need to share this understanding to design, implement, and maintain privacy-related functions in software systems [3]. Changes to the software can create privacy threats as a side effect. Engineers need to be able to understand and identify emergent privacy threats at their end and involve the business when necessary. This paper focuses on the engineers. It has been shown that there is a major gap between privacy requirements and how software engineers' perceive and interpret privacy [4]. This issue has received little research attention.

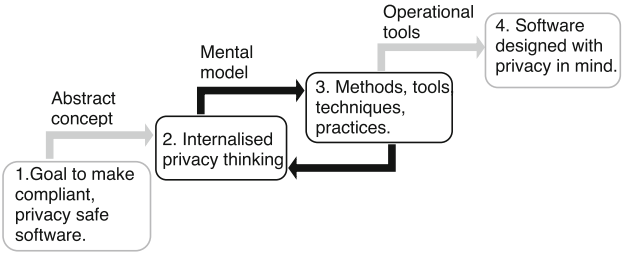

Fig. 1. A gap (shown in black) in the development and deployment of privacy-by-design for software design. The figure has been contextualised from method adoption framework of [5]

The gap between the development and deployment of privacy-by-design for software design is illustrated through Fig. 1. The figure depicts the necessary components of successful development and deployment of privacy-by-design for software design.

Privacy as a high-level concept and a goal (Step 1) has been widely adopted by organisations and can be assumed to be well known for engineers [4]. Operational methods, tools and techniques exist (Step 3), such as privacy engineering methods and privacy standards. However, the "internalised privacy thinking" step (Step 2) between goal and operation is poorly supported, which hinders the use of provided operational tools. In addition, current operational tools do not support well the forming a mental model of privacy. Hence, the learning loop between 2 and 3 that develops privacy practice is no well supported. As a result of the gap, engineers proceed directly to the provided operational tools, without the help of a mental model of what privacy means in the context. Engineers end up having to interpret privacy requirements while lacking the skills to do so [4]. Having a mental model is essential for new practices [6], like for engineers to effectively operationalise privacy-safe business processes in the software design. When the gap is present, there is a risk of mismatch between what was intended and what was built. Thus, the goal represented on Step 4 is not reached.

To close this gap, we need to focus on a critical and difficult privacy-by-design task for engineers, that includes collaboration with business and understanding of privacy in the context: the identification and addressing of *systemic* [9] privacy threats. This activity is a pre-requisite for risk-based privacy-by-design: deciding on and implementing the right technical software functionality. With systemic privacy threats, we mean threats that arise from the interplay of the software's business purposes, technology in use, and people who it touches, without forgetting its wider context [1,20]. The ever-evolving unbounded nature of software makes this task even harder, since engineers to need be able to understand what emergent, unknown privacy threats may arise from the software system's interacting and ever-changing aspects [7].

So far this activity, identifying systemic privacy threats, has not been well supported in the described context. Current methods and tools either take a reductionist approach that does not suit complex software contexts; only target either engineers or business; omit aspects of privacy threats such as the technology or impact to people; focus on direct compliance requirements do not consider threats of systemic nature at all; or do not support forming a mental model of privacy.

We look to systems thinking to address the described gap, focusing on the task and tools for privacy threat modelling. Systems thinking is aimed at understanding complex targets such as today's ever-evolving unbounded software systems [8]. It commonly promotes focus on the whole rather than parts, the dynamic behaviour of the system, relationships and interconnections, and how system behaviours (such as threats) arise from the system's structure. Systems thinking commonly utilises conceptual modelling. Complexity of the scenario (in one's mind, in order to work with it) is reduced by conceptual modelling [8]. This in turn builds a mental model for the observer, an internal representation of the real world, and improves their overall ability to deal with the complex scenario [10,11].

The rest of the paper is organised as follows. Section 2 considers the concepts of privacy and privacy threats as well as complexity and systems thinking. Section 3 discusses existing approaches. Section 4 presents the research agenda. Finally, towards the end of the paper, Sect. 5 draws some final conclusions.

2 Background

Privacy as a Value and a Requirement. Turning abstract elements like values or a goals into system functionality is a known struggle [12]. Public values such as privacy have made their way into non-functional requirements (NFRs), but it is argued [13] that they should be treated differently since they are essentially values, not requirements. An important point with public values as NFRs is that they are cross-cutting concerns that should cover all parts of the design. Shishkov and Mendling propose a metamodel [13] in which value consideration targets business process models, based on which software functionality can be

specified. Therefore, ideally, public values are operationalised directly into functional solutions rather than separately gathered through requirements engineering process and scattered in relevant places.

For some values, the pressure to include them comes from the public, but some are legislated for. Next, we discuss the regulatory issues in the EU, but it is noted that similar legal frameworks exist elsewhere in the world. In the EU, commonly agreed public values are written in the EU Charter of Fundamental Rights. Two of the rights, data protection and privacy, have been particularly prominent in the area of software development in recent times. Data protection is known through its implementation as the General Data Protection Regulation (GDPR) [1]. This paper considers in particular the values of data protection and privacy through the requirements of the GDPR. The term privacy is used.

Privacy as a value differs from others and deserves particular attention in the context of software design due to several reasons:

- Privacy directly and concretely relates to software design, when personal data is used in software.
- Lack of privacy has direct human impact; it can result in real harms to people, even death [20].
- Privacy and especially privacy threats are wide complex concepts, challenging to understand in the context of today's complex sociotechnical software.
- There are known challenges implementing privacy in software functionality [4].
- Unlike many values, privacy is not a choice but a legal requirement, with accountability requirement and potential sanctions for non-compliance [1].

Attempts have been made to turn the GDPR into a list of NFRs which then are turned into functionality by engineers [14]. This approach of gathering privacy NFRs from legal and adding it to the requirements list is far from ideal. Since the GDPR requirements are cross-cutting, they must be evident everywhere in the software's design and because of privacy being essentially a value, it should ideally be built in the business processes. This paper is concerned of the process of arriving at functionality that have privacy requirements built in. This implements the idea of data protection by design and is in line with the observations of Shiskov and Mendling [13].

The GDPR contains only some clear requirements but also the requirements to do data protection by design (DPbD) and to consider impacts to people. The clear requirements are simpler to incorporate in business process models and are commonplace seen as NFRs. For example, tracking of sensitive data, controlling data transfers outside of the EU and ensuring conditions for consent. By nature, they are already more "technical", and can be excused to be listed as the NFR. However, the DPbD and impact assessment are processes that are meant to produce privacy-aware functionality for the software, and are difficult to satisfy without an understanding of privacy in both business and technical viewpoints and a way to assess very complex situations. This is where privacy threat modelling is essential.

Going back to public values, to operationalise a value, one needs to identify possible threats to it. In this case, these are privacy harms. The GDPR requires these harms to be understood very widely in the DPbD and impact assessment. They may be physical, material or non-material damage to people as a result of processing personal data [1,20]. Being subject to unethical data processing is one. It is expected that they arise from interplay of different elements in the situation, meaning that they are systemic threats. Because of this wide concept of a privacy threat extending far outside of the business processes, they are especially difficult to identify. Understanding of the threats and impacts arising from the personal data utilising business processes as whole should result in an understanding of what particular business processes should be varied for privacy and how. They can then be expressed as a privacy-safe business processes variants, that can then be turned directly into functionality. However, since privacy issues have a clear technology aspect and business processes manifest as software, these privacy-by-design activities cannot take place in the business side alone. Engineers and their understanding is essential.

Software Complexity and Systems Thinking. Today's software systems are complex. Since they increasingly revolve around people, they should essentially be viewed as social systems [16]. This raises their complexity and means that they have no clear boundary.

From the viewpoint of Lehman's SPE-classification of computer programs with respect to their evolution, these are E-type systems: constantly evolving, embedded in their environment and aiming to satisfy their users' varying needs [15]. An underpinning idea in systems thinking is that such systems are best approached through holistic synthesis—aiming to understand the whole—instead of reductionist analysis, trying to understand their parts in isolation and aggregating the results [16]. The approach to understand them should be flexible to match their complexity. Ashby's law of requisite variety means that if the target has high variety, such as complex ever-evolving software, the variety of the intervention has to match it [17,18].

Systems thinking [8] is a way to approach complexity, commonly through focusing on the whole rather than parts, dynamic interconnections of the parts and behaviour of the whole arising from that as well as the system's structure. Iterative approaches are common and the problem situation is often probed by different techniques and from different angles, for example multiple cause diagramming, rich pictures, systems maps, and multiple-perspective techniques [9]. Sense-making, understanding and learning have an important role in systems thinking [9]. Systems thinking approach and developing one's own thinking go hand in hand, which is an area widely researched by Senge [19].

3 Existing Approaches for Privacy Modelling

Our main interest is on tools that consider privacy, impact to people from the processing of personal data and tools that are placed in the requirements engineering context. Privacy threats arise from the interplay of business purposes

and technical aspects as well as people and the wider context, so ideally all of these, and the interplay aspect, are present. Applications of systems thinking to threat and impact assessments are also relevant.

The GDPR [1] includes the *data protection impact assessment* requirement, which includes the requirement to understand systemic privacy threats. It does not elaborate how exactly to uncover the privacy threats. Although technical experts are recommended to take part, it is a generic, not requirements engineering process. Data protection authorities have published versions of the process but the privacy threat identification stage lacks detailed methodology [20]. Impact assessments in general promote learning about the situation [21].

Value-sensitive design includes techniques that aim to incorporate public values in software designs [22], for example Security and Privacy Threat Discovery Cards [23]. These tools and guidelines consider privacy along other impacts in systemic manner, taking the wider context into account. The techniques offered under the value-sensitive design brand are not particularly attached to the requirements engineering context. *Technology assessment* has a wider scope and is used for example in medical and new innovative technology contexts [24]. Recent topic of artificial intelligence ethics has inspired *ethics tools* such as ECCOLA [25], which is a method for incorporating ethics in requirements engineering. ECCOLA involves the listed aspects; however, its privacy considerations are not developed far enough.

Various operational tools and methods exist for *privacy engineering*, such as PRIAM [26], the "design science approach" [27], LINDDUN [28] and Elevation of privacy cards [29]. Wider impact to people is not considered. These methods commonly take a reductionist approach involving techniques such as detailed mapping, making them inflexible and resource heavy for complex targets [21], although the last two allow for a lighter application as well. Reductionist approaches by their nature are not equipped to uncover systemic privacy threats, but rather address the more straightforward compliance requirements. Also, they lack the benefits of learning and mental model development that systems thinking approach has. Systems thinking has been widely used for identifying *systemic threats and solutions to global problems*, such as climate change [30], and for public policy development.

The gap between privacy requirements and engineers' understanding of privacy may be addressed by training and education. However, our focus is on the practical task and learning through doing and through cooperation. Therefore, pure privacy training and education is not in the scope of this paper.

4 Research Agenda

Based on the above background, research is needed to improve operationalising privacy requirements in technical software functionality. We argue that a systems thinking approach should be explored as a possible way forward. We believe that the targeting privacy threat modelling task would improve deciding on and implementing the right functionality for privacy and result in ever-improving privacy mental model for the users, engineers especially.

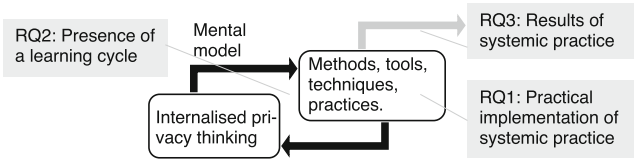

Fig. 2. Research questions illustrated against the identified gap

The agenda aims at answering to the following research questions, which have been illustrated in Fig. 2 against the identified gap between privacy requirements and implementing privacy in software:

- **RQ1:** *Through what kind of practical implementation could systems thinking approach be included in the privacy threat identification processes?* RQ1 targets the practical implementation. It addresses the practical need of engineers and benefits them, and for the academic community, widens the understanding of how systems thinking approach could be brought into this setting. Answering RQ1 would help to understand what features of system thinking approach produce desirable effects.
- **RQ2:** *How does using systems thinking approach in the privacy threat identification process impact on the forming of a mental model of privacy in engineers?* RQ2 contributes to RQ1 by checking that the implementation does what it is envisioned to: builds a mental model. Should strong learning effect be evident, that knowledge could be applied and further researched in the areas of privacy awareness and training. Even without evidence of learning, RQ3 would let us find out how well the tool would support the task of operationalising privacy in software functionality.
- **RQ3:** *What is the impact on arriving at software functionality for privacy in terms of efficiency, effectiveness and efficacy, if a systems thinking approach is used in process of uncovering systemic privacy threats?* RQ3 contributes to RQ1 by checking that the output is meaningful privacy threat information for deciding on software functionality. RQ3 would let us find out how well the tool would support the task of operationalising privacy in software functionality.

To answer RQ1, the practical implementation of systems thinking, we plan to carry out a literature review to describe and explore current practices in operationalising privacy in software design; those in particular that involve privacy threat identification or systems thinking approach. We will also review systems thinking approaches to identify suitable features for inclusion in privacy threat identification. With the gathered understanding, we plan to insert systems thinking approach in privacy threat identification in the requirements engineering setting and test its effects. In practice, this means creating a tool with systemic features and testing and developing it in an iterative manner using action learning approach.

To answer RQ2, the presence of a learning cycle, we plan to run quasi-experimental studies of engineers using the tool against traditional, non-systemic

approaches. We will use observational method, surveys, interviews and analysis of the quality of the requirements elicitation outputs to gather data of the learning. The aim is to evaluate the tool's learning value to engineers in their requirements engineering task, and to explain the relationship between systems thinking approach and privacy mental model forming in this setting.

To answer RQ3, the output of systems thinking practice, we plan to run quasi-experimental studies as in RQ2, to describe and explain how taking a systems thinking approach impacts the results that the tool produces. Mainly qualitative data in the form of privacy threats and requirements will be gathered and analysed. The effectiveness, efficiency and efficacy of the tool will be evaluated.

In general, an action learning approach will be taken to begin answering the research questions. Action learning approach suits the practical aim of helping engineers. Action learning suits research that is about system thinking approach; ideally this results in double loop learning, where learning happens both about the target and the learning itself, and the approach is modified on the way to better respond to the changing situation. In this case, we could see an action learning cycle taking the research forward about how to include action learning (in the form of systems approach) into privacy threat modelling practices.

5 Conclusion

This paper was framed around the wider challenge of turning the NFRs and cross-cutting concern for privacy into technical functionality in software. We traced this to the lack of shared understanding and cooperation between business and engineering, and focused on the major gap in engineers' understanding of privacy. Privacy threat modelling was identified as a concrete activity through which overall improvements could be made.

We made a distinction between straightforward privacy requirements and systemic privacy threats. The identification of systemic privacy threats is essential for deciding and implementing the right functionality in software. Recognising also the increasing complexity of software, we argued that applying systems thinking approach to privacy threat modelling would bring improvement. Systems thinking is well suited at understanding complex situations such as privacy issues in today's ever-evolving software, improving one's mental model in the process. We expect that improvement at the engineers' end will benefit business process owners alike.

Our future research plans include investigating practical tools for privacy threat modelling, the presence of a learning cycle while using them, and the practical outcomes for requirements engineering. Therefore, our contribution comprises of (i) a presentation of a critical problem in business and software design to address and (ii) a tentative solution with a formulated research agenda to address the problem.

Acknowledgements. This research was partially funded by Business Finland under ITEA 18033 Mad@Work.

References

1. Regulation (EU) 2016/679 General Data Protection Regulation (GDPR). 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. Official Journal Of The European Union L119/1 (2016). http://eur-lex.europa.eu/legal-content/FI/TXT/?uri=CELEX%253A32016R0679
2. Kostova, B., Gürses, S.,Troncoso, C.: Privacy engineering meets software engineering. On the challenges of engineering privacy by design. ArXiv:2007.08613 [cs], 16 July 2020. http://arxiv.org/abs/2007.08613
3. Sinnhofer, A.D., Oppermann, F.J., Potzmader, K., Orthacker, C., Steger, C., Kreiner, C.: Increasing the visibility of requirements based on combined variability management. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 203–220. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_13
4. Hadar, I., et al.: Privacy by designers: software developers' privacy mindset. Empir. Softw. Eng.. **23**, 259–289 (2018)
5. Ebert, C., Abrahamsson, P., Oza, N.: Lean software development. IEEE Comput. Archit. Lett. **29**, 22–25 (2012)
6. Senge, P.M.: Mental models. Plann. Rev. **20**(2), 4–44 (1992). https://doi.org/10.1108/eb054349
7. Anthonysamy, P., Rashid, A., Chitchyan, R., Lancaster, S.: Privacy requirements: present & future. In: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS) (2017). https://ieeexplore.ieee.org/document/7961663
8. Arnold, R., Wade, J.: A definition of systems thinking: a systems approach. Proc. Comput. Sci. **44**, 669–678 (2015)
9. Monat, J., Gannon, T.: What is systems thinking? A review of selected literature plus recommendations. Am. J. Syst. Sci. **59**, 11–26 (2015). http://resources21.org/cl/files/project264_5674/Overv
10. Richardson, G., Andersen, D., Maxwell, T., Stewart, T.: Foundations of mental model research. In: Proceedings of the 1994 International System Dynamics Conference, pp. 181–192 (1994)
11. Jones, N., Ross, H., Lynam, T., Perez, P., Leitch, A.: Mental models: an interdisciplinary synthesis of theory and methods. Ecol. Soc. (2011). https://www.jstor.org/stable/26268859
12. Chung, L., Nixon, B., Yu, E., Mylopoulos, J.: Non-functional Requirements in Software Engineering. Springer, Heidelberg (2012). https://doi.org/10.1007/978-1-4615-5269-7
13. Shishkov, B., Mendling, J.: Business process variability and public values. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 401–411. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_31
14. Miri, M., Foomany, F.H., Mohammed, N.: Complying with GDPR: an agile case study. ISACA J. **2**, 1–7 (2018)
15. Lehman, M.: Program evolution. Inf. Process. Manag. **20**, 19–36 (1984)
16. Ackoff, R.: Systems thinking and thinking systems. Syst. Dyn. Rev. **10**, 175–188 (1994)
17. Ashby, W.: Requisite variety and its implications for the control of complex systems. Cybernetica **1**, 83–99 (1958). http://pcp.vub.ac.be/Books/AshbyReqVar.pdf

18. Braithwaite, J., Braithwaite, J., Wears, R., Hollnagel, E.: Resilient Health Care. Volume 3, Reconciling Work-as-Imagined and Work-as-Done. CRC Press (2016). https://www.finna.fi/Record/jamk.993205274806251
19. Senge, P., Sterman, J.: Systems thinking and organizational learning: acting locally and thinking globally in the organization of the future. Eur. J. Oper. Res. **59**, 137–150 (1992)
20. Privacy Impact Assessment PIA Knowledge Base (2018). https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-3-en-knowledgebases.pdf
21. Raab, C.: Information privacy, impact assessment, and the place of ethics. Comput. Law Secur. Rev. **37**, 105404 (2020)
22. Hendry, D.: Designing Tech Policy: Instructional Case Studies for Technologists and Policymakers. UW Tech Policy Lab (2020)
23. Denning, T., Friedman, B., Kohno, T.: Security and privacy threat discovery cards. University of Washington (2013). http://securitycards.cs.washington.edu/assets/security-cards-deck-with-croplines.pdf
24. Nemoto, E., Issaoui, R., Korbee, D., Jaroudi, I., Fournier, G.: How to measure the impacts of shared automated electric vehicles on urban mobility. Transp. Res. Part D: Transp. Environ. **93**, 102766 (2021). https://www.sciencedirect.com/science/article/pii/S1361920921000705
25. Vakkuri, V., Kemell, K., Abrahamsson, P.: ECCOLA - a method for implementing ethically aligned AI systems. In: Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020, pp. 195–204 (2020)
26. De, S., Métayer, D.: PRIAM: A Privacy Risk Analysis Methodology. Springer, Heidelberg (2016). http://link.springer.com/10.1007/978-3-319-47072-615
27. Oetzel, M., Spiekermann, S.: A systematic methodology for privacy impact assessments: a design science approach. Eur. J. Inf. Syst. **23**, 126–150 (2014). https://www.tandfonline.com/doi/full/10.1057/ejis.2013.18. ISBN 1476-9344
28. Yskout, K., Heyman, T., Landuyt, D., Sion, L., Wuyts, K., Joosen, W.: Threat modeling: from infancy to maturity. In: Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2020, pp. 9–12 (2020)
29. F-Secure Elevation of Privacy, Privacy Cards for Software Developers (2018). https://github.com/F-Secure/elevation-of-privacy. Issue: 1.1, vol. 2021
30. Li, H., Wang, X., Zhao, X., Qi, Y.: Understanding systemic risk induced by climate change. Adv. Clim. Change Res. **12**, 384–394 (2021). https://www.sciencedirect.com/science/article/pii/S1674927821000782