"Elderly, with location data, while shopping?" Spotting Privacy Threats Beyond Software: A Quasi-Experimental Study

Tuisku Sarrala
Faculty of Information Technology
University of Jyväskylä, Jyväskylä, Finland
email: tuisku.rad.sarrala@jyu.fi

Tommi Mikkonen
Faculty of Information Technology
University of Jyväskylä, Jyväskylä, Finland
email: tommi.j.mikkonen@jyu.fi

***Abstract*—In software development, privacy has become an increasingly critical aspect due to privacy legislation, the growing complexity of software, and the private nature of many computing systems. However, studies reveal that developers often have security-focused understanding of privacy and expect user privacy needs to align with their own. This can risk regulatory compliance and potentially lead to harm to individuals. In this paper, we present a quasi-experimental study that explores how a card-based privacy threat modeling method using systems thinking elements could help to think about privacy threats on a broader scope and from another person's perspective. Sixty-five software engineering course participants used the same card deck. The experimental group created several scenarios, whereas the control group described their software with the cards. Both reflected against privacy principles. The experimental group's threats had broader and more often social scope, showed consideration for individuals, and were more often context-based. The control group's threats were more security focused and had software artifact focused scope. These findings help to understand how developers' understanding of privacy could be broadened. On a practical level, they have the potential to improve current privacy-by-design tools and methods, ultimately leading to more robust privacy protection in software development.**

Keywords*—*Privacy; privacy impact; software development; card-based modeling; systems thinking; personas; scenarios; process improvement.

I. INTRODUCTION

Today's companies processing personal data cannot avoid addressing privacy, which involves protecting personal information and assessing potential impacts on individuals. Privacy-related legislation enforces the requirement for understanding and mitigating harmful impact, such as the EU General Data Protection Regulation (GPDR) [1] and the forthcoming EU AI Act [2]. A lack of understanding of privacy threats, the source of privacy impact, can eventually expose customers to the risk of subjective and objective harms, and the company to significant financial losses. Consequently, privacy has become an important non-functional property to consider when building and deploying software systems.

Previous studies show that software developers have a narrow security-focused understanding of privacy [3]. During development, software is typically considered as a technical artifact. However, when the software is in use, it becomes a socio-technical system operating in the rich real world with real privacy-vulnerable individuals as its users. Threats arising from this complex context can go unnoticed by developers if their focus is merely on the security of the technical artefact. Therefore, to be successful in mitigating privacy threats in software development, studies suggest developers' understanding of privacy ought to be broadened [3]–[5]. Moreover, improving privacy tooling could help the situation, since developers prefer practical solutions and rely on privacy tools [3]. Furthermore, tools based on traditional reductionist approaches are a poor fit for broadening one's understanding of complex socio-technical issues [6].

Our research seeks to improve the situation, based on the following:

- **Engineering activity**: We target *privacy threat modeling* since it is a practical privacy thinking exercise which developers take part in.
- **Approach**: We apply *systems thinking*, which is an approach that
  - helps to understand *complex* issues, such as today's socio-technical software and privacy;
  - helps to develop one's thinking; and
  - involves techniques suited for broadening developers' understanding of privacy, through offering multiple perspectives, narrative and human focus.
- **Implementation**: We utilise on practical familiar techniques that fit systems thinking approach, namely *personas* and *scenarios* techniques, and *ideation cards*.

In this paper, we present a quasi-experimental study with 65 software engineering course participants during a short course where they developed a piece of software in small teams. We experimented with a card-based privacy threat modeling approach and examined its outcome, hoping to see threats that focus on privacy, consider harm to individuals, and have a broader scope beyond the technical artifact. In particular, we were interested in how a method with systems thinking features compares to a method with traditional features in privacy threat discovery in terms of identified threats.

"Systems thinking" analyzes complex situations by examining interactions and dependencies among system components [7]. The study revealed that with the systems thinking based method the identified privacy threats were wider in scope, less security-focused, more contextual and human-focused. Our results suggest that moving attention from the technical artifact to the wider context and directing it to the users' potential pri-

 ISBN: 978-1-68558-098-8

vacy vulnerabilities and interaction with the artifact appear to be key factors in improving privacy threat modeling outcome, leading to an enhanced understanding of privacy as a whole.

The rest of this paper is structured as follows. In Section 2, we present the background for the work. In Section 3, we describe our research approach and method. In Section 4, we provide the results of the study. In Section 5, we present a discussion regarding the key findings. In Section 6, we draw some final conclusions.

II. Background

A. Privacy and Privacy Threats

Privacy is a multifaceted concept that has many definitions, such as the right to be let alone [8] or one's control over their data [9]. A particular characteristic of privacy is that it touches people through the lack of it, resulting in harm which we call the *privacy impact*. Hence, it has been argued that rather than asking what privacy is, we should focus on the negative impacts and harms to address privacy [10]. Along similar lines, Daniel J. Solove, an influential privacy researcher, offers a taxonomy of privacy violations [11]. Privacy legislation too recognises the harms and impacts viewpoint. For example, the GDPR [1] mandates the anticipation of negative impacts arising from personal data processing to ensure the protection of individuals. In this article, we take the harms and impacts focused approach to privacy, and define *privacy impact* as any negative impact to individuals arising from the processing of personal data, in line with the GDPR. Similarly, we define *privacy threat* as something that has the capability of causing privacy impact to an individual.

Engineering is dominated by an approach that considers privacy as pre-definable through the application of compliance requirements and privacy principles [12]. Privacy-related legislation dictates the generic approach to privacy threat identification. Privacy impact and data protection impact assessment methods and templates from authorities [13] are commonly followed. Specifically for technical audience, there are privacy engineering methods for privacy threat and impact identification [14][15]. Typically, extensive modelling or description of the target is required and then compliance requirements, privacy principles or privacy goals are iterated against it. However, many threats can arise from the rich real world context where the system operates, and the traditional narrow focus on pre-defined privacy issues and the artifact model can leave these threats unnoticed.

B. Developers' Understanding of Privacy

Understanding privacy is imperative for software developers, for several reasons. The developers are required to (1) make privacy-safe and ethical design decisions, (2) spot and escalate privacy issues that they observe from their deeply technical viewpoint, and (3) collaborate effectively with business owners and company legal experts. These objectives consolidate the developers' contribution to the company achieving its main goal, legal compliance.

Recent research [4][5] points out that supporting developers' privacy understanding and practical privacy work is an under-researched area that requires attention. Security dominates when it comes to privacy in software development, but privacy is much more than security. Security is prominent in privacy research related to developers, developers understanding, and privacy tools for developers [3]–[5]. Previous research [3][16] shows that developers use security vocabulary to discuss privacy, which limits their perception of privacy to external threats, such as a hacker gaining access to personal data.

There are further aspects narrowing down developers' views. Developers have a practical rather than theoretical understanding of privacy, which does not match the policy makers' view [3][16]. Developers' understanding is built through practical work and online communities, as well as by observing the (at times questionable) privacy practices of big tech companies [4][17][18]. User privacy appears to be considered through the developer's own privacy persona [17][19], whose privacy needs may vastly differ from the needs of the individuals that become the users [19][20].

Due to the lack of skills in implementing privacy in practice, developers rely heavily on privacy tools and methods to carry out the necessary tasks [4]. Simultaneously, effective use of these tools is hampered due to developers still lacking a mental model of privacy. Training alone appears to be insufficient in bridging this gap, due to the lack of its practical relevance [4].

C. Systems thinking

"Systems thinking" is a conceptual approach used to understand and analyze complex situations, problems, or phenomena by focusing on the interactions and interdependencies among various components or elements within a larger system [7][21]. "System" could be comprised of physical entities, processes, people, organizations, or even abstract concepts. Arnold and Wade [7] propose the following definition for the purpose of systems thinking: "*Systems thinking is a set of synergistic analytic skills used to improve the capability of identifying and understanding systems, predicting their behaviors, and devising modifications to them in order to produce desired effects. These skills work together as a system.*". In practice, systems thinking commonly involves taking a holistic view, examining the dynamic interconnections of the different elements and observing the behaviour of the system that arises from the interconnections and the system's structure (i.e., behaviour that cannot be seen by examining the parts in isolation) [21]. Specific techniques include probing from different angles, multiple perspectives and narrative techniques [21].

The need for systems thinking skills for dealing with complex socio-technical systems and problem situations is recognised in literature [6][22]. Many of today's software systems can be seen as social systems [23]. They are embedded in their environment and constantly evolving [24]. This makes them complex and their boundaries blurred. Privacy is a complex social-technical issue, and more complex and ever-evolving software has more potential to create dangerous combinations for privacy. Both generic and engineering approaches

 ISBN: 978-1-68558-098-8

for identifying privacy threats commonly rely on a detailed description and analysis of all the parts of the target. This traditional reductionist engineering approach can be heavy and poorly matched to today's complex software systems and privacy threats [6]. Where traditional approaches may struggle, systems thinking is suited for understanding complex human of problem situations [6].

D. Understanding Users and Their Privacy Vulnerabilities

Personas is a technique to model actual users as fictitious personas so that the software design can better fit their needs and expectations. Personas are commonly generated based on focus groups, interviews and workshops [25]. In the case of privacy personas, the unveiled privacy preferences of the users are used for privacy persona creation [26][27]. Understanding gained through personas can be enriched and strengthened by the use of scenarios [28]. *Scenarios* are a general non-standardised way of creating narratives around user activities. Personas based on users' preferences may not, however, help to identify privacy threats towards them, since their source may not be known to the user. The users may not be aware of their privacy vulnerabilities nor understand how threats may arise through their interaction with the software. Modeling personas with a variety of privacy vulnerabilities, rather than privacy preferences, could address this. This approach would be similar to the suggestion to model personas with various disabilities [29].

E. Card-based Implementation

Card-based design tools are virtual or physical cards commonly used for various design, planning, brainstorming, and collaborative activities. The cards typically represent individual pieces of information, concepts, tasks, or elements, and can be arranged, grouped, and manipulated by the users. A review of card-based design tools [30] shows that the most worthwhile outcomes appear to be produced by cards that stimulate creativity, facilitate early user participation, and summarise design or good practice guidelines. The review called for independent scientific trials since the cards had been evaluated mostly by their developers. Recent studies with privacy-related cards list accessibility and potential for communicating complex ideas as their benefits, and well as how cards intertwine with practice rather than separate privacy to be considered in separate forms [18]. Cards can enhance understanding at an introductory level, bring practical value and engage participants with the topic [17][31]. Weaknesses include overloading user with information, topic oversimplification, and being difficult to use, apply and update [30].

Next, studies with setups similar to ours are discussed. In contrast to our study, the identified privacy threats were neither studied nor reported in detail. Rather, the studies focused on the process of using the cards, so their findings will be more relevant for our future studies. We were interested in card-based methods that have a threat discovery element and scoped out methods that see privacy as pre-definable [12], such as compliance and privacy principle checklists.

1) Security and Privacy Threat Discovery Cards: The method [32] has some elements that are present in systems thinking approach, namely multiple perspectives and combining cards to create new viewpoints. The focus is on security threats by an attacker, but impact on humans is considered. The deck has four suits: human impact, adversary's motivations, adversary's resources and adversary's methods. Cards can be added. The eight human impact cards cover impacts on a wide scope: emotional, financial, physical and societal wellbeing, relationships, security of personal data, the biosphere and 'unusual impacts'. Different activities with the cards are suggested, such as combining, sorting, considering the unusual, and risk assessment.

2) An Ideation Card Study "Playing the legal card": The study was carried out by Luger et al. [31]. The systems thinking element of multiple user perspectives is present but the card usage was very linear. The target scenario was created specifically for the study. The card deck contained cards covering four GDPR requirements (data breach notification, use of consent, right to be forgotten and privacy by design), cards providing context (a description of a system), cards providing user groups (e.g., older people, ex-offenders, women of all cultures and faiths), and cards with system constraint descriptions. System architects and programmers took part along with HCI and research oriented players. The players drew one card of each category for discussion at five minute intervals and then discussed all of them for 15 minutes. The user cards reportedly had a significant effect on the system design. The groups saw the users through a stereotype, but these stereotypes highlighted several privacy issues.

3) An Ideation Card Study: The study by Tang et al. [17] elaborated the "Playing the legal card" study and involved teams of undergraduate students completing an industry-sponsored software development project. All the projects were different. The deck was similar to the "Playing the legal card" deck. From systems thinking perspective, the elements were the same. The cards included personas with qualities such as age, mental health, language, country, gender spectrum and physical health. The teams drew user, constraint and regulation cards from the deck, discarding cards which they felt were not applicable to their software. Teams could draw more cards, time permitting. The teams discussed each card for five minutes, and then all the drawn cards together. A week later, the teams provided a list of changes to be made in their projects based on the session. The teams struggled to understand the privacy concepts on the cards and rarely were able to generalise the concept to the team's context. Nevertheless, the authors suggest that privacy ideation cards are a promising pedagogical tool and should be used in student projects to help students learn about privacy.

III. Research Approach

A. Research Question and Experiment Design

To investigate how the problem situation could be improved, we set up a quasi-experiment. We selected privacy threat modeling as the practical engineering activity and implemented

 ISBN: 978-1-68558-098-8

it as an experimental card-based tool with added systems thinking features. To control the experiment, we implemented a similar tool but with traditional features instead. The research question we sought to answer was:

RQ: How does a method with systems thinking features compare to a method with traditional features in privacy threat discovery in terms of identified threats?

In the experimental version, the systems thinking elements were the following:

- Multiple perspectives: looking at the situation through the persona cards' perspectives
- Narrative technique: creation of scenarios with the personas and explaining these to others
- Exploring interconnections and system behaviour arising from them: creation of scenarios from different elements and observing what privacy threats they may generate
- System's blurred boundary, context, environment: The modeling was not bounded by the software artifact boundaries or centred to that. The artifact was not modeled but was to be kept in mind.

Our reasoning for including personas was that they would:

- add a social dimension and thus broaden the scope where threats can appear (not bounded to the technical artifact);
- reveal impacts – privacy is easier to understand through its impacts, than through the abstract privacy concepts;
- illuminate in depth why privacy matters, since they amplify privacy threat effects for the particular persona due to their special vulnerabilities [31]; and
- offer an alternative for the participants reflecting against their own personas.

In the control version, the traditional features were the following:

- The modeling steps: the target is first described and then a check through privacy principles is carried out
- Focus: the technical artifact in the centre
- Pre-defined and checklist-based approach to privacy: the target is compared to privacy principles.

Although not a traditional feature but a compromise, we opted to ask the control group to describe their target with the given cards rather than words or diagrams, so that the cards available to both teams would be the same.

For the visual design and user instructions, we took into account the recommendations from the other card based studies. The cards were designed to be aesthetically pleasing and the threat modeling was organised as fun game, with short descriptions and no jargon, as recommended [17]. Along the recommendations, we provided an information session about privacy before the exercise and provided the teams all the cards as a reference, rather than restricting participants to drawn cards. In the experimental game, the user scenario was designed to be considered before the privacy principles. This is in line with "Playing the Legal Card" [31] where the participants saw the user and technology cards forming one inseparable whole, and ranked them higher in importance than the privacy regulation cards. Furthermore, the experimental game was designed to move focus from the technical artifact to threat scenarios, which is supported by a machine learning ethics cards study [33] stating that,"focus should be less on technology and more on consequences and implications". In another ethics focused card based study [18], it was observed that participants 'clustering' cards together enabled more nuanced discussions and communicating about complex threats. The experimental teams 'cluster' cards together into scenarios.

To narrow down the exercise, the discovered threats were not required to be a risk assessed. Free threat brainstorming was encouraged on the basis that a larger number of threats, less criticism on the ideas, allowing unusual ideas, and building on the ideas of others would produce more quality threats [34] and therefore be more valuable for the risk assessment process that would normally follow. The experiment focus was on initiating broader privacy thinking, rather than a final plausible threat listing. The participants were explained that in an industry setting, the threats would be the raw material for a risk assessment process, where their quality, likelihood and impact would be weighed, but that would not be part of this exercise.

B. Participant Selection and Experiment Setting

The experiment was conducted during a five-week remotely taught (online) software engineering course. The course was open both to persons already in the industry as well as to current students at master's level. Sixty-five participants gave research consent. The participants responded to a pre-course survey that asked how confident they were in any programming language and how many years of work experience in software engineering or development they had. The participants' work experience varied from none to over 10 years. Twenty participants had no relevant work experience; 19 had less than 1 year; 20 had 1-5 years; and 6 had over 6 years.

The main course assignment was to develop a piece of working software in teams of 3-5 participants. Participants were arranged in 16 teams, which were split to experimental and control group, as shown in Table I. The majority's experience in each team is in bold. It was not disclosed to the participants whether they belonged to the experimental or control group. The split was based on the confidence scores and then the experience scores, making the groups equal and avoiding variance within teams, as far as practicable.

The developed software was to be an online auction system, where users could sell and buy goods by bidding. The required features included email registration, user authentication, seller and buyer interfaces, system operator functions, and currency conversion. The course had an industry sponsor, and the team who delivered the best solution was promised a low-value prize.

In week two, all the participants were given a 30-minute basic lecture about privacy. In week four, the participants were given a 15-minute lecture focusing on privacy threats, and introducing the privacy threat modeling game that was created for the experiment. At the end of the lecture, the participants were instructed to play the privacy threat game within their

 ISBN: 978-1-68558-098-8

TABLE I
CONFIDENCE LEVEL AND YEARS OF WORK EXPERIENCE IN PROGRAMMING.

Experimental Team	Confidence 0-10	Work experience in yrs 0-10+
Team E1	1.8-3	**0**, 1-5
Team E2	3.6-4	**<1**
Team E3	4.1-5	0, **<1**, 1-5
Team E4	6	0, **<1**, 1-5
Team E5	6.9-7	**<1**, 1-5
Team E6	7-8	0, **1-5**, 6-10
Team E7	7-8.5	0, **<1**, 1-5
Team E8	9	1-5, **6-10**, 10+
Control Team	Confidence 0-10	Work experience
Team C1	0-1	**0**, <1, 6-10
Team C2	4	0, 1-5
Team C3	5-5.5	**0**, <1, **1-5**
Team C4	5.5-6	0, <1
Team C5	6-6.9	**0**, 1-5
Team C6	7.5-8	**0**, <1, 10+
Team C7	4.5-8.5	<1, **1-5**
Team C8	9-10	<1, **1-5**

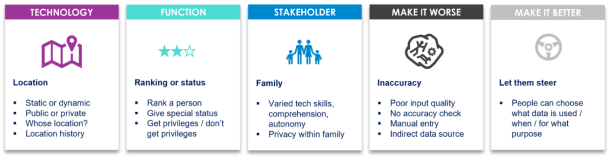

Figure 1. Examples of each card.

teams at their chosen time, with the aim of identifying privacy threats relating to the software they were designing. It was not disclosed to the participants that there were two different versions of the game.

C. Experimental and Control Game Implementation

The experimental card deck design was similar to the deck in the ”Playing the legal card” [31] study. The cards depicted personas, technological context and privacy principles. Both games included the same cards, examples shown in Figure 1. There was a difference in usage of the software aspect cards and the game board, as shown in Figure 2.

The five categories of the cards were:

- Software aspect cards, describing the following:
  - Technology: 10 technologies that may be utilised in software (chat bot, office software, AI/machine learning, sensor, wearable, mobile phone, website, wireless, photos and video, location)
  - Function: 9 software functions (marketing, profile, ranking or status, security, shopping, social, access and identification, customer service, incidents and accidents)
  - Stakeholder: 10 personas that the stakeholders could be (elderly, family, influencer, knowledge worker, person with a past, child/teen, contractor, temporary staff, very important person, visitor)
- 12 “Make it worse” cards describing things that may weaken privacy in software, like anti-privacy principles

Figure 2. Experimental teams’ board A and control teams’ board B.

(1) Familiarise yourself with the game board. You can add your own cards by using the blank ones, if the ones provided are not enough.
Experimental teams: “Your scenario” area is used to represent different scenarios relating to the team software with one blue, one green and one purple card.
Control teams: “Your software” area is used to represent different aspects of the team software. From the blue, green and purple cards, choose all the technologies, functions and stakeholders that could represent your software now or in the future. Place them in the purple, green and blue boxes below (”Your software”).

(2) Play the game. The cards are meant to give you ideas, rather than restrict you. So don’t be bound by the exact things written on them.
Experimental teams: First make a scenario that could relate to your software, now or in the future. Pick one purple, green and blue card and place them in the middle under ”Your scenario”.
Both teams: Split into two roles: Baddies vs Goodies. Play takes place in turns. The Baddies’ aim is to come up with privacy problems with the (**E**: software scenario)/(**C**: software). Baddies have the “Make it worse” cards to help them with bad ideas. The Goodies’ aim is to mitigate baddies’ ideas. Goodies have the “Make it better” cards to help them to mitigate the bad ideas. The Baddies go first. Baddies choose one card to worsen the software scenario and describe verbally a privacy problem that would happen, relating it to the software in a believable way. Then it is Goodies’ turn to mitigate it, with the help of one or more ”Make it better” cards. The aim is that the Goodies can mitigate all the privacy problems that the Baddies come up with, and keep the software safe for people to use. After each round, write down the privacy problem and its mitigation in the ”Privacy problems catalogue” below the game board.
For the next round, swap roles and re-use all cards as you like! (**E**: You can change the scenario. Mix and match to make new scenarios) Come up with as many problems as you can. Play for 30-45 mins.

Figure 3. Instructions as given to the teams.

 ISBN: 978-1-68558-098-8

(more is more, inaccuracy, kept forever, vague purposes, identity and access, revealed, not available, no secrets, don't tell them, take advantage, sensitive data, combination)

- 12 "Make it better" cards describing things that may strengthen privacy in software, like privacy principles (use and reuse controlled, fair and ethical, minimise, expiry date, identity and access, let them steer, tell them about it, fresh and accurate, basic data, data segregation, availability, it's confidential)

Each card listed privacy vulnerabilities or related considerations about the topic or persona, to help the participant to think of the topic from the privacy angle. The cards were not specifically tailored for the target, but generic in their nature. All the card categories included also one "invent your own" card.

Written instructions to the teams are shown in Figure 3. The experimental teams were instructed to consider three software aspect cards of their choice at a time (1 technology, 1 function, 1 stakeholder) to come up with threats. The control teams had an additional step in the beginning, to choose all the software aspect cards that related to their software and lay them out on the game board. After that, they could start freely identifying threats.

The game was delivered through a Miro board[1]. A separate password-protected board was created for each team. The boards included a gaming area, cards movable with a mouse, a table where to record the identified threats and mitigations, and written instructions on how and why to play the game. Participants were instructed to record their online gaming session (showing a shared screen, with no participant video required). Everyone participated remotely. There was no facilitator. Using a game format instead of workshop format allowed participants to play without a facilitator, thus minimising external influences. A previous study [17] had listed strict time limits as a limiting factor for higher level cognitive processes. In our study, the teams could ultimately decide themselves how long they would play, but 30-45 minutes was instructed.

IV. Results

Once the teams had carried out the privacy threat exercise, the resulting privacy threat catalogues were collected from each team for analysis. The threats that the teams identified were categorised from different viewpoints to reveal differences in the number of threats discovered by the experimental and control groups based on threat type, scope, contextuality and the description for the harmed party, as well as the overall number of threats. In addition, the time taken to do the exercise was noted.

The threat analysis was done by coding the threat descriptions by the main author. For the initial coding the group codes were hidden and the threats were mixed, after which a sample of approx. 20% was reviewed by a researcher outside of this project. The final codings were adjusted based on the reviewer comments. Codings which had no room for interpretation were not validated, such as merely highlighting the word used to describe the harmed party.

[1]miro.com

TABLE II
Type of threats discovered.

Type	Experimental teams	Control teams
Privacy	31	21
Security	4	16
Other	8	6
Total	43	43

TABLE III
Scope of the threats.

Scope	Experimental teams	Control teams
Software	15	**26**
Malicious	5	**14**
Social	**21**	3
Society	**2**	0

A. Number of Threats and Time Taken

It was found that teams in both experimental and control groups reported similar number of threats, between 4 and 7, approximately 5 each. Coincidentally, both groups' total was 43 threats. Among both experimental and control groups, the more experienced teams found fewer threats than the less experienced (around 4.4 against 6). No instructions on how many threats should be identified had been given to the teams, but the privacy threat catalogue template included three numbered rows and a help text how to add more.

The teams were instructed to play for 30-45 minutes, and they actually played for 28-65 minutes based on the lengths of the video recordings. On average per team, those in the experimental group played for 42:23 in mm:ss (5:39:07 in total, hh:mm:ss), whereas those in the control group played for 38:56 (5:11:27 in total). On average per team, those in the experimental group played approximately 3.5 minutes longer than those in the control group.

B. Types of Threats

The threats were categorised under three different types of threats:

- Privacy threats: Threat relates to how the person's personal data is used or exposed, or how their private life is exposed. Example: "The system collects data without telling the user and uses the data for other purposes." (Team C3)

TABLE IV
Descriptions of the harmed party.

-	Experimental teams	Control teams
None	12	**18**
Neutral	12	**25**
Persona	**19**	0

 ISBN: 978-1-68558-098-8

- Security threats: Threat is a security issue without a distinct privacy element. Example: ”Database credentials reveals from public GitLab repository and gives full access to database.” (Team E8)
- Other threats: Threat is about harm to a person, but does not relate to privacy or personal data use or exposure. Example: ”Shopping website can have secretly extra fees hidden from customers.” (Team E3)

Table II shows that experimental group uncovered a higher number of privacy threats than the control group.

C. Contextuality of Threats

The contextuality of the threats was analysed as either:

- Pre-definable: Threat could have existed on a generic checklist and the context does not matter much. Example: ”Sensitive and unnecessary data is collected from users.” (Team C8)
- Context-based: The threat is such as it would not have existed on a generic checklist, but it arises from the context. Example: ”Financial status of the user can be identified though his purchase history.” (Team C3)

Control group found 24 pre-definable threats and 19 context-based threats. Experimental group found 14 pre-definable threats and 29 context-based threats.

D. Scope of Threats

The scope of each threat was categorised, from narrow to wide scope:

- Software: Threat description is limited to the scope of software, where something is wrong with the software and it can be fixed there. Example: ”Transactional data is never removed, regardless of success or date of the auction” (Team C6)
- Malicious party: Threat materialises through a malicious party, a greedy company or an attacker, internal or external. Example: ”If forms are not controlled enough, user can input malicious data on input fields such as SQL injection and destroy or steal user data from database.” (Team C4)
- Social: Threat materialises through how people interact with the software and touches people’s social sphere. Example: ”An elderly user inputs wrong payment data (account number, telephone number, address).” (Team E8)
- Society: Threat touches the society. ”The app could collect excess amounts of GPS data from user’s phone, the user could be e.g., a member of the parliament. Threat to national security.” (Team E3)

Table III shows that most of the experimental group’s threats were on the social scope. The control group uncovered threats on a narrower scope, with most of their threats being on the software and malicious scope. Only three social threats and no society threats were identified by the control group.

E. Descriptions of the Harmed Party

The words used by the experimental and control groups to describe the harmed party were as follows:

- Experimental teams: Child, family, elderly, famous person, influencer, knowledge worker, member of parliament, teenager, VIP, seller, customer, person, ’they’, user, (or: no description)
- Control teams: Buyer, customer, person/people, someone, user, (or: no description)

The description of the harmed party for each threat was categorised as follows:

- No description: The threat description did not describe the harmed party in any way. Example: ”Some page contains forgotten debug lines that reveal too much data.” (Team C3)
- Neutral description: The harmed party was described as user, seller, buyer, customer, person/people, someone/they/who. Example: ”Without authentication and with shared credentials user would see other user’s info.” (Team C4)
- Persona description: The harmed party has a persona. Example: ”Customer service worker is obsessed with a famous individual, which happens to contact our customer service. Now he/she learns lots about his/her target of obsession!” (Team E4)

Table IV shows that that the control group’s descriptions were limited to neutral descriptions of buyer, customer, person or people, someone and user. Experimental group used persona descriptions from the cards as well as descriptions that appear to be inspired by the cards (famous person, member of parliament) in addition to neutral descriptions. Both groups had threats where they had not described the harmed party at all. The neutral descriptions were invented by the individual teams. Within the experimental group, team E1 did not use any personas, and in contrast, team E4 only used personas. The rest of the experimental teams used a mix of personas, neutral descriptions and no descriptions.

The control group were instructed to choose the relevant cards in the beginning. Thus, their game boards were analysed to ascertain to what extent that had narrowed their selection of stakeholder cards. One control team had picked 9 out of the 10 stakeholder cards, five teams had picked 4-6 stakeholder cards, and two teams had picked 1-2 cards and supplemented their selections with 5-6 invented neutral stakeholders (such as seller, buyer and product owner). The experimental group chose relevant cards as the game went on. Based on the average number of threats found, the teams in the experimental group had picked 1-5 different stakeholder cards during the game.

V. Discussion

In this study, we set out to find ways to broaden developers’ understanding of privacy beyond security and bring more focus to the harm to individuals via improving the threat modeling process. The research question to be addressed was:

 ISBN: 978-1-68558-098-8

RQ: How does a method with systems thinking features compare to a method with traditional features in privacy threat discovery in terms of identified threats?

The findings to the question were that the experimental group's threats had broader and more often social scope, were more often context-based and described the harmed party more often in a personal way. The control group's threats were mostly security-focused, their scope was the software artifact and the harmed party was described in a non-personal way.

The control group's results were in line with existing research regarding developers' understanding of privacy. The experimental group's results showed a positive result, pointing to that systems thinking features may improve the situation and is a promising direction of research. The findings could be used to inform the design of privacy threat modeling and privacy impact assessment methods for developers as well as privacy education.

A. Why Did the Same Card Deck Yield Different Results?

1) More Material to Consider: The experimental group used the software aspect cards to create several scenarios and the control group used them in a static way, for describing the software. The experimental group's changing scenarios introduced new additional viewpoints for every round, which means that they had more new material to consider than the control group. Having more material did not result in higher number of threats identified, but it may have contributed to the wider scope and contextuality of the threats. The experimental teams played approximately 3.5 minutes longer each. Therefore the experimental group was slightly slower, but not considerably, taking into account the extra scenario creating.

2) Scenarios Before Privacy Principles: The control group relied on the privacy principle cards for identifying threats, which may have led them to describe their threats more often in a pre-definable way, stating what is written on the card. The learning value of cards for understanding privacy *concepts* (privacy principle cards) is not well supported [17]. The experimental group had to be already thinking of threats when constructing the scenarios before applying the privacy principle cards. The threat scenario creation stage likely led to the threats being not pre-defined, but unique.

3) Mixing and Matching: Connected to scenario creation, mixing and matching cards may have contributed to the experimental group threats having a wider scope and more contextual, since mixing and matching is a different sense-making activity to concept generalisation. Whilst this study did not analyse the interaction with the cards, the instructions were that the experimental teams should mix and match cards, whereas control teams were instructed to pick a card (privacy principle / anti-priciple). The other card-based studies reported on the varying usage of the cards, such as sorting, grouping and so on, but not on the effects of this. It is likely that the card sorting was done in an attempt to increase understanding of the cards and possibly for getting more ideas.

4) Personas for Social Threats: Each scenario had a stakeholder card depicting a persona. This meant that the experimental group was first focused on the persona's privacy story, rather than the privacy concepts. This likely led the experimental group to use the personas in their threat descriptions. In "Playing the legal card" [31], the persona cards had a major effect, causing the players to see threats through their individual circumstances. Similarly in our study the experimental teams described nearly half of their threats through the personas and centred their threats to them. The control group did not describe any personas, probably since their focus was foremost on the static, described software artifact. This in turn may explain the very low number of social threats for the control group. In contrast, the experimental group's scenarios were inherently social since they always involved a persona, and the scenarios were natural interactions rather than happening inside the software artifact.

B. What May Have Affected the Results?

1) Time and Available Threats To Be Found: The combined effect of time, potential threats to be found, and the participants' effort, motivation and privacy-related experience, is difficult to establish. Neither the time for the task nor the threats to be identified were controlled. It is not known how many potential threats there were to be found in each software, which were all slightly different. This made the task more realistic but less controlled. Furthermore, this study was interested in non-pre-defined contextual threats in particular.

2) Controls for Persona Use: Whilst personas appear to have improved the experimental group's threats in our study, persona use comes with challenges [25] that may have affected this experiment. The experiment did not include detailed instructions about how to consider the persona cards, so it is possible that the participants did not know how to apply them against their software. For example, it was not explicitly stated that the personas depicted in the stakeholder cards had privacy vulnerabilities, although the bullet points under each hinted that way. It is possible that the experimental group relied on the personas too much, since all but one team used them in their threat descriptions. Due to the story-telling nature of scenarios and the personas not being directly representative of the software's users from the viewpoint of its functionality (buyers, sellers, etc.), it is possible that threat scenarios became a stories of their own, rather than tightly relating to their software. The control group did not use any of the personas given on the stakeholder cards in their threat descriptions although six out of the eight teams had selected them to represent their software. Again this could be due to the personas appearing unrepresentative. In addition, the control group was not challenged on their stakeholder card selections after they had selected them in the beginning.

3) Controls for Participants: The participants' programming confidence and software engineering related work experience was varied. Having variety is natural in the industry and being in a group somewhat helped to balance the variety. Participants were arranged in groups based on experience and

 ISBN: 978-1-68558-098-8

confidence, so that the difference between the results of high and low confidence and experience could be also compared. No information about participants' understanding of privacy was collected beforehand. The effects of this was mitigated by delivering all participants the same lectures about privacy. No training on the delivery platform, Miro, was given, but it was excepted that due to its simplicity, the possible learning curve would not be too steep. Participants were instructed to familiarise themselves with the platform functionality before beginning the exercise. Since the exercise was completed only once, the teams could not enjoy the benefits of learning the platform and the tool. The participants' physical environment was not controlled but all individuals were remote.

C. Threats to External Validity

1) Presence of Complexity and Systems Thinking: One of the drivers for the research was using systems thinking for understanding complex systems. Although privacy and socio-technical systems are complex phenomena and the results may be generalizable to those, the teams' modeling targets were not complex from a technical viewpoint. Hence, the results' applicability to technically complex systems remains open. Traditional approaches may work well with simple systems [6], but so far as our control method can be considered "traditional", it was outperformed by the experimental method from our research perspective. Systems thinking approach uses various techniques, some of which can be found in other settings too but here they were applied from the systems thinking perspective. For example, the personas technique is commonly used to model the actual users, whereas here it was used to bring in multiple perspectives and probe the issues. However, it would be worth exploring whether the general approach or the specific techniques made the difference, or perhaps their interplay.

2) Realistic Control Method: Instead of using an existing traditional method for the control groups, the control method was specifically designed for the experiment, making it somewhat artificial and simplified. This was a compromise to increase control of the experiment, but it can lower the generalizability of the results. The control group version was designed based on the traditional way of identifying privacy threats, where the teams built a representation of the software (their selection of relevant cards) and then examined it against the privacy principles and anti-principles. The control group's 'traditional' results indicate that the control version design was successful and provided appropriate control for the experiment.

3) Plausibility of the Threats: Due to the threat scenario building encouraging story-telling, there is a chance that the experimental group threats came out as far fetched stories about the personas and were not so closely related to the software. This is not a major concern since in this study we were interested in what can evoke broader and human centred privacy thinking, rather than focusing on the threats' quality from the impact and likelihood assessment viewpoint. It is also possible that any implausible threat scenarios can be modified to plausible ones in the risk assessment stage.

4) Generalizability to Developers and Industry Setting: The experiment was not carried out with software developers in an industry setting. Two thirds of the participants had no or very little relevant work experience, while the remaining third been in the industry for at least one year. When looking at the teams, all but two teams had participants with at least one year of industry experience, which helps to increase the generalizability of the results. Other aspects that made the set-up more realistic were that the course had an industry sponsor acting as the client, who evaluated and commented on the final pieces of software at the end of the course, and the experiment was embedded as a natural element in the software development process.

D. Directions for Future Work

To address the threats to validity and limitations discussed above, we plan to validate the findings in an industrial environment with a more complex target and extend the analysis to the plausibility of the threats. Secondly, we plan to analyse the session recordings so that comparisons to the other card-based studies can be made, which concentrated on the participants' interaction with the cards. To investigate the participants' understanding of privacy, the session recordings could be analysed for cognitive processes, as done in study by Tang et al. [17].

In terms of the experimental method, in addition to reviewing and refining all of the cards, the persona cards and related guidance should be developed further. Each persona's privacy vulnerabilities should be stated more clearly and users should be instructed clearer on their usage for reflection.

VI. Conclusion

In this paper, we were motivated by the potentially harmful combination of the impact that developers can have on user privacy and their limited security-focused understanding of this subject. In response, we designed a quasi-experiment that targeted the privacy threat modeling activity and investigated how an experimental method with systems thinking features compared to a traditional-style method in terms of identified threats.

The threats identified within the experimental group prominently considered wider contextual factors and human interactions, which equals to a positive result showing a broader view of privacy. The control group, employing the traditional-style method, generated more security-focused threats, aligning with the prevailing norms. We attribute the experimental group's result to the shift of focus from the software artifact and privacy principles to the human interaction with the software beyond its technical boundaries. The shift was achieved with the use of personas and scenarios with a systems thinking approach. These techniques can be inserted in privacy tools and methods to improve current practice and ultimately help to produce more privacy safe software. Following these results, we plan to gain additional insights by analysing the session

 ISBN: 978-1-68558-098-8

recordings, refine the cards and the user guidance accordingly, and finally validate the results in the industry.

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," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj [Retrieved: October, 2023]

[2] Proposal for Regulation (EU) COM/2021/206 final, "Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts." 2021.

[3] I. Hadar et al., "Privacy by designers: software developers' privacy mindset," *Empirical Software Engineering*, vol. 23, no. 1, pp. 259–289, 2 2018.

[4] M. Tahaei, "The Developer Factor in Software Privacy," Ph.D. dissertation, The University of Edinburgh, 2021.

[5] M. Peixoto et al., "The perspective of Brazilian software developers on data privacy," *Journal of Systems and Software*, vol. 195, p. 111523, 1 2023.

[6] P. F. Katina, C. B. Keating, and R. M. Jaradat, "System requirements engineering in complex situations," *Requirements Engineering*, vol. 19, no. 1, pp. 45–62, 2014.

[7] R. D. Arnold and J. P. Wade, "A definition of systems thinking: A systems approach," in *Procedia Computer Science*, vol. 44, no. C. Elsevier, 2015, pp. 669–678.

[8] S. D. Warren and L. D. Brandeis, "The Right to Privacy," *Harvard law review*, vol. 4, no. 5, pp. 193–220, 1890.

[9] A. F. Westin, *Privacy and freedom*. New York: Atheneum, 1970.

[10] W. Hartzog, "What is Privacy? That's the Wrong Question," *U. Chi. L. Rev.*, vol. 88, p. 1677, 2021.

[11] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 477–564, 2006.

[12] R. Y. Wong and D. K. Mulligan, "Bringing Design to the Privacy Table," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 5 2019, pp. 1–17.

[13] Information Commissioner's Office, "Sample DPIA template," 2018. [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-impact-assessments/ [Retrieved: October, 2023]

[14] M. C. Oetzel and S. Spiekermann, "A systematic methodology for privacy impact assessments: a design science approach," *European Journal of Information Systems*, vol. 23, no. 2, pp. 126–150, 3 2014.

[15] K. Wuyts, L. Sion, and W. Joosen, "LINDDUN GO: A Lightweight Approach to Privacy Threat Modeling," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 9 2020, pp. 302–309.

[16] M. Peixoto et al., "On Understanding How Developers Perceive and Interpret Privacy Requirements Research Preview," in *Requirements Engineering: Foundation for Software Quality*, ser. Lecture Notes in Computer Science, N. Madhavji, L. Pasquale, A. Ferrari, and S. Gnesi, Eds., vol. 12045. Cham: Springer International Publishing, 2020, pp. 116–123.

[17] Y. Tang, M. L. Brockman, and S. Patil, "Promoting Privacy Considerations in Real-World Projects in Capstone Courses with Ideation Cards," *ACM Transactions on Computing Education*, vol. 21, no. 4, p. 34, 12 2021.

[18] L. D. Urquhart and P. J. Craigon, "The Moral-IT Deck: a tool for ethics by design," *Journal of Responsible Innovation*, vol. 8, no. 1, pp. 94–126, 2021.

[19] A. R. Senarath and N. A. G. Arachchilage, "Understanding user privacy expectations: A software developer's perspective," *Telematics and Informatics*, vol. 35, no. 7, pp. 1845–1862, 10 2018.

[20] S. Sheth, G. Kaiser, and W. Maalej, "Us and Them: A Study of Privacy Requirements across North America, Asia, and Europe," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: Association for Computing Machinery, 2014, pp. 859–870.

[21] J. P. Monat and T. F. Gannon, "What is Systems Thinking? A Review of Selected Literature Plus Recommendations," *American Journal of Systems Science*, vol. 2015, no. 1, pp. 11–26, 2015.

[22] K. E. Dugan, E. A. Mosyjowski, S. R. Daly, and L. R. Lattuca, "Systems thinking assessments in engineering: A systematic literature review," *Systems Research and Behavioral Science*, vol. 39, no. 4, pp. 840–866, 7 2022.

[23] R. L. Ackoff, "Systems thinking and thinking systems," *System Dynamics Review*, vol. 10, no. 2-3, pp. 175–188, 1994.

[24] M. M. Lehman, "Program evolution," *Information Processing & Management*, vol. 20, no. 1-2, pp. 19–36, 1 1984.

[25] D. Karolita, J. McIntosh, T. Kanij, J. Grundy, and H. O. Obie, "Use of personas in Requirements Engineering: A systematic mapping study," *Information and Software Technology*, vol. 162, p. 107264, 10 2023.

[26] E. Kim, J. K. Yoon, J. Kwon, T. Liaw, and A. M. Agogino, "From innocent irene to parental patrick: Framing user characteristics and personas to design for cybersecurity," in *Proceedings of the International Conference on Engineering Design, ICED*, vol. 2019-August. Cambridge University Press, 2019, pp. 1773–1782.

[27] M. Rudolph, S. Polst, and J. Doerr, "Enabling users to specify correct privacy requirements," in *Requirements Engineering: Foundation for Software Quality: 25th International Working Conference, REFSQ 2019, Essen, Germany, March 18–21, 2019, Proceedings 25*, 2019, pp. 39–54.

[28] J. Salminen, K. Wenyun Guan, S. G. Jung, and B. Jansen, "Use Cases for Design Personas: A Systematic Review and New Frontiers," in *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, 4 2022, pp. 1–21.

[29] J. T. Nganji and S. H. Nggada, "Disability-Aware Software Engineering for Improved System Accessibility and Usability," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 47–62, 7 2011.

[30] R. Roy and J. P. Warren, "Card-based design tools: A review and analysis of 155 card decks for designers and designing," *Design Studies*, vol. 63, pp. 125–154, 7 2019.

[31] E. Luger, L. Urquhart, T. Rodden, and M. Golembewski, "Playing the legal card: Using ideation cards to raise data protection issues within the design process," in *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2015-April. Association for Computing Machinery, 4 2015, pp. 457–466.

[32] T. Denning, B. Friedman, and T. Kohno, "The Security Cards," 2013. [Online]. Available: http://securitycards.cs.washington.edu/index.html [Retrieved: October, 2023]

[33] K. E. K. Bilstrup, M. H. Kaspersen, and M. G. Petersen, "Staging reflections on ethical dilemmas in machine learning: A card-based design workshop for high school students," in *DIS 2020 - Proceedings of the 2020 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, Inc, 7 2020, pp. 1211–1222.

[34] J. E. Danes, J. Lindsey-Mullikin, and K. Lertwachara, "The sequential order and quality of ideas in electronic brainstorming," *International Journal of Information Management*, vol. 53, p. 102126, 8 2020.

 ISBN: 978-1-68558-098-8