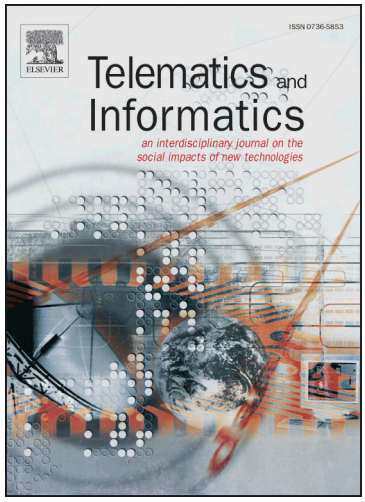

Understanding user privacy expectations : A software developer's perspective

Awanthika R. Senarath, Nalin Asanka Gamagedara Arachchilage

PII: S0736-5853(18)30296-X
DOI: https://doi.org/10.1016/j.tele.2018.05.012
Reference: TELE 1131

To appear in: *Telematics and Informatics*

Please cite this article as: Senarath, A.R., Arachchilage, N.A.G., Understanding user privacy expectations : A software developer's perspective, *Telematics and Informatics* (2018), doi: https://doi.org/10.1016/j.tele.2018.05.012

Understanding user privacy expectations : A software developer's perspective

Abstract

Software developers are trained to develop and design software applications that provide services to users. However, software applications sometimes collect users' data without their knowledge. When applications collect and use users' data without transparency, this leads to user privacy invasions because users do not expect the application to collect and use these information. Therefore, it is important that software developers understand users' privacy expectations when designing applications in order to handle user data transparently in software applications. However, due to the lack of systematic approaches to extract user privacy requirements, developers end up designing applications either based on their assumptions on user privacy expectations, or relating to their own expectations of privacy as a user. Nevertheless, how accurate these perceived privacy expectations are against actual user expectations is not currently known. This research focuses on investigating developers' privacy expectations from a user point of view against users' privacy expectations. We also investigate developers' assumptions on user privacy expectations against actual user privacy expectations. Our findings revealed that developers' assumptions on user privacy expectations are close to their own expectations of privacy from a user point of view and that developers' privacy expectations from a user point of view are significantly different from actual user privacy expectations. With this understanding, we provide recommendations for software developers to understand and acknowledge user expectations on privacy when they design and develop applications.

Keywords: Software Developers, Designing Privacy, Usable Privacy, User

Privacy Expectations, Software Development

1. Introduction

Software developers design and develop software applications to provide services to users such as banking, online-shopping and social networking. However, sometimes these software applications collect data users do not expect the application to collect and save details users do not expect the application to save [1], which may lead to privacy invasions. For example, when mobile applications request permission from users to access their data [2], users are known to accept these permission requests without much consideration to its content, trusting the applications [3]. Therefore, if applications request data users do not expect the applications to collect, it may lead to users disclosing data, such as their location, to the application without their knowledge, which compromises users' privacy [4].

As a solution to this, software developers are expected to consider user privacy expectations and minimize mismatched behaviors of applications that could lead to privacy invasions when they design software applications [5]. In order to minimize mismatches in software application behavior against user expectations, developers either need to design data collection and use in applications close to what users expect, or communicate the differences to the user transparently [6, 5, 7, 8]. For this, it is important that developers understand how user expect the application to behave [9, 8, 4, 10]. However, so far there is no systematic method that instructs how developers should understand user privacy expectations in the software development process [11, 12], which leads to developers designing software applications based on their perception on user privacy expectations. Nevertheless, to date, how accurate this perceived user privacy expectations are from actual user privacy expectations, and how the perceived user privacy expectations affect developers' behaviors when designing software applications are not known [13, 14, 15, 16].

We conduct this research to understand developers' perception on user pri-

vacy expectations and how this perceived privacy expectations affect their software development activities. This information is important to understand why developers design software applications that behave different to the way users expect them to behave. By referring to the existing work that focus on developers' privacy perception, we identified two factors that could affect developers' perception on user privacy expectations. Firstly, developers' own privacy expectations from software applications from a user point of view [14]. Secondly, the assumptions developers make on end user privacy expectations when they embed privacy into software application designs [16]. We focus on these two aspects in this research in order to understand how different developers' perceived user privacy expectations are from actual user privacy expectations. Then by incorporating a simple software design task for developers we observe how these differences in the perceived user privacy expectations affect the way developers embed privacy into software applications.

We incorporate the control group method [17, 18] in our research methodology with two groups of software developers, where one group of developers were instructed to play the role of a user. With the role-play developers we observe developers' own privacy expectations from a user point of view. With the other group of developers (without role-play), we observe the assumptions developers make about user privacy expectations. We also recruit users in order to compare the privacy perceptions of the two groups of software developers against that of actual users. By using this approach we also attempt to observe the potential of using role-play as a possible way to encourage developers to understand user privacy expectations better. Altogether the study had 54 participants, 36 developers (18 role play, 18 without role-play) and 18 users.

The key findings of this study are as follows,

- The privacy expectations of users are significantly different to the privacy expectations of software developers when they were playing the role of a user. Developers are observed to have a reduced level of privacy expectations compared to users when they use software applications and they by

default expected applications to collect more data than users.

- Developers assumptions on user privacy expectations are close to developers own privacy expectations when they were playing the role of the user. Developers' relate to their personal expectations when assuming user privacy expectations when they design software applications.

- Developers' design decisions in designing software applications reflected the difference in the perceived user privacy expectations. When instructed to consider privacy in application designs developers intuitively focus on technical aspects and user privacy expectations are not something developers focus on.

These findings suggest that developers may not be capable of embedding privacy into software applications to a level that is expected by users based on their perception of user privacy expectations. There is a need for better techniques within software development processes to identify user privacy expectations and integrate them into the application designs. Based on our findings we propose recommendations for user privacy requirement elicitation in software application development.

The paper is structured as follows. In the next section we discuss previous work in the area of software developers' privacy behaviors and user privacy expectations. Section 3 explains the research objectives followed by the research methodology section. Section 5 presents our results and in section 6 we extensively discuss the results based on existing knowledge. Finally we present our conclusions and future directions for research.

2. Related Work

Here we discuss previous work on user privacy expectations and developers privacy perception to position our work. We also discuss current privacy methodologies and their take on user privacy expectations in order to establish the knowledge gap, and to define the grounds on which our work stand.

2.1. User Privacy Expectations

There are ample studies that investigate user privacy expectations from software applications and how these expectations are not adequately handled in software application designs. Here we discuss two studies that intrigued our interest due to their unique take on user privacy expectations. Rao et al. [8] investigated the mismatched expectations of application users against the actual behavior of web sites on data collection, sharing and deletion practices. Their study revealed that actual web site behaviors are significantly different to user expectations particularly in data collection and deletion [19, 8]. Similarly, Adams and Sasse [6], identify a "privacy invasion cycle", which shows how users' inability to make accurate assumptions on the behavior of software applications leads to a cycle of privacy invasions. They emphasize the importance of understanding user privacy requirements to protect users, and not just the data in implementing privacy in systems. Both these studies state that rather than understanding the importance of giving users feedback and control in systems, it is important that developers perceive when and why users need these controls. Nevertheless, compared to research that attempt to identify the mismatches in user privacy expectations against actual software behavior [8, 20, 6], knowledge on developers' perception on user privacy expectations and how this perception affects developers' design decisions in software development is currently lacking.

2.2. Developers and their Privacy Perceptions

In a study focusing on how developers make design decisions about user privacy, Ayalon et al. [14] claim that developers' personal privacy expectations significantly affect developers' professional development practices. They suggest that when embedding privacy into software application designs, developers often relate to their own personal privacy expectations. This is known as the implicit effect of unconsciously transferring one's personal expectations into their professional behavior [21].

However, Sheth et al. [20] have identified that developers generally have low concerns towards privacy compared to users. Focusing particularly on data ag-

gregation, distortion, sharing and breaching incidents, they claim that developers are more concerned about technological approaches such as anonymization to mitigate privacy concerns, whereas users are shown to be more concerned about the policies. While it is possible that the incapability of users to grasp technological terms such as distortion, anonymization and aggregation had an impact on these findings, their findings suggests a difference in the focus areas of developers and users when they perceive privacy. Therefore, when developers base their design decisions on their personal privacy expectations, it is possible that they end up with different implementations of privacy in software applications compared to what user expect. However, currently there exist no formal evaluation on how different developers and users are in their perception of privacy expectations and how these differences affect developers' professional activities when they embed privacy into software applications.

2.3. Existing Privacy Methodologies and Developers' Assumptions

Analyzing how developers perceive privacy when they develop software, Wurster et al [22], stress that developers should not be always expected to know the right thing to do. Similarly, Lahlou et al [15], has shown that most developers do not consider privacy to be an important problem in software development. Therefore, it is important that user privacy expectations are identified as a specific requirement when developing software applications.

Nevertheless, in most existing software requirement elicitation methodologies privacy is considered as a sub-category of security requirements. In the instances where privacy requirements are considered as a separate entity, the requirements are mostly elicited either from data protection regulations [23] or from a technical perspective. For example, Kalloniatis et al. [10] propose *Pris*, a systematic methodology to identify privacy requirements for software systems focusing on technical privacy concepts such as anonymity, unlink-ability and un-observability. Similarly, Oetzel et al. [24] propose a systematic approach for assessing privacy in software systems identifying the requirements from data protection regulations. Identifying user privacy expectations and relating them

to software design requirements to minimize mismatched behaviors that may result in privacy vulnerabilities [8, 25, 26, 6] is not therefore given attention in current privacy requirement elicitation methods.

Consequently, most of the existing privacy designing and implementation methodologies, such as the Privacy by Design (PbD) Principles [27], Fair Information Practices (FIP) [28] and Privacy Impact Assessment (PIA) [29] are criticized to be vague and unclear in providing guidance to developers as to how user privacy expectations should be understood [30, 31]. In situations where clear guidance is not available to elicit requirements developers naturally end up deriving requirements based on their assumptions [32]. Therefore, when embedding privacy into software application designs, developers end up assuming user privacy expectations [16, 15]. However, currently there is not enough focus on what affects the assumptions developers make about user privacy expectations and how these assumptions affect the way developers embed privacy into application designs.

In comparison to previous work, in this research our goal is to understand developers' perception of user privacy expectations and how the perceived privacy expectations affect the software design decisions made by developers. Based on the literature analysis we identified two factors that affect developers perception of user privacy expectations. Namely the personal privacy expectations of developers from a user point of view [14] and the assumptions developers make about user privacy expectations [33, 34]. We designed our study to investigate if developers have different privacy expectations when they use software applications compared to users and how different the assumptions developers make about user privacy expectations are against actual users expectations. Our study also investigates how the mismatches in developers perception on user privacy expectations affect developers' decisions when they design software applications.

3. Research Objectives

Based on the definition by Miller [35], in the context of this research we define expectation to be what the user think the application would do. As users behave with a software application according to what they think the application would do we believe this definition fits our research goal. We also briefly investigate on users' desired data collection expectations, corresponding to what they expect the application should do, as opposed to what they think the application would do to observe how users come to cope with application behaviors that do not align with their desired expectations. However, unless otherwise stated, for the remainder of the paper, when we say expectations we mean *what users think the application would do.*

We formed the following research questions (RQ) when designing our experiment.

3.1. RQ1

We first observe the priority developers and users give to privacy when they use software applications. Here we define priority as the importance given to privacy when using a particular software application. Understanding and acknowledging the priority users give for privacy is considered important in embedding transparent privacy into software products [8, 1]. Developers are motivated to embed privacy into the software applications they design based on their perception on how important users would consider privacy when they use the application. Therefore, the level of importance users give for privacy as perceived by developers is important. Because if developers are unable to understand the priority users give for privacy, irrespective of the developers' commitment towards privacy, they may not be able to address user privacy requirements in their software designs to the extent to which users expect. We form the following research question in order to investigate this.

How different is the level of priority given to privacy between developers and users when they use software applications?

Here we investigate the level of priority developers would give to an application if they were to use it. We then investigate software developers' assumption on the priority users would give for privacy for an application. We compare these two against the actual priority level users give for privacy if they were to use the application in order to understand the differences.

3.2. RQ2

Another aspect which often leads to mismatched behaviors in software application is data collection [8]. It has been shown that users are reluctant to disclose their personal information into software applications unless they feel the information is required for the application [36]. Surreptitious data collection is considered a major concern of users when they interact with software applications [8]. Lack of transparency and mismatched expectations for privacy between users and developers could lead to unexpected data collection in software applications which could be considered surreptitious irrespective of the developer's intention.

For example, users could be completely oblivion to the fact that a particular application requires to collect their location and hence would not expect the application to collect their location. However, a developer may assume users to expect the application to collect their location and not include explicit notices to communicate that, which leads to privacy invasions. Therefore, it is important that the differences in the way developers perceive how users would expect an application to collect data is understood. We form the following research question to investigate this in the experiment.

How different user expectations are from the actual data collection practices developers embed into software applications?

Here, we observe how developers' assumptions and their personal expectations on data collection differ from actual user expectations. Then, by assigning a design task for the developers in the study we also observe how the perceived user expectations affect the decisions developers make when they design software applications.

3.3. RQ3

Another important aspect where mismatches could occur in embedding privacy into software applications is the perception on sensitive data. Users' expectations and knowledge as to why certain sensitive information is needed in applications have a major impact on users' feeling of privacy and trust decision making in an application [4]. Furthermore, users expect software applications to have explicit security (encryption, secure storage) for their sensitive data. However, if developers do not understand which data is sensitive to users, and if the data items developers perceive to be sensitive is different to what users consider to be sensitive, this may lead to trust issues and dissatisfaction in users when they use software applications.

Data protection laws [37, 38, 39, 40] also define explicit laws against accessing and storing sensitive data, with different interpretations to as what sensitive data are. For example, the GDPR (European General Data Protection Directive) which is coming into action in 2018 [41], has broadened their definition on sensitive data, including more data items, such as user IP addresses, into the sensitive data category. Similarly, ENISA (European Union Agency for Network and Information Security) guidelines for Privacy and Data Protection by Design [42] also recommends that developers identify which data is important for users privacy when they design software systems and treat them accordingly. Therefore, understanding the data items users consider sensitive to them is important for developers, when they design and develop software applications to acknowledge data as human content [6]. We form the following research question to investigate this.

How different users and developers perceive sensitive data when using/developing software applications?

Here we investigate if developers and users consider same data elements to be sensitive to themselves. We also investigate if developers assumptions on sensitive user data when they design applications is similar to user concerns.

4. Research Methodology

The study was conducted remotely following similar privacy and security research that involve software developers [43]. Previous researchers have also stated that involving software developers for in lab studies are often faced with the difficulty to get professional developers to participate due to higher costs and unavailability [43, 44]. Further it is said that most participants prefer remote participation over in-lab studies [45]. Similar work that use platforms similar to the amazon mechanical turk to recruit participants for remote studies state that if special attention is given to designing the task, remote studies could be used for highly interactive tasks ranging from simple surveys to prototype development [46].

We designed the study based on an application scenario, because user privacy expectations are dependent on the application type [8, 20]. Furthermore, contextual studies are considered better in retrieving reliable and consistent answers from participants compared to general surveys [47]. We used a hypothetical application scenario related to health information as the context for our study for two reasons. Firstly, health information is considered sensitive in data regulations [48, 49]. Secondly, health data if collected and used within legal boundaries are considered highly useful and important [50]. These constraints give developers a strong incentive to collect data within the application, and yet to be mindful of the privacy concerns [49]. Due to higher sensitivity users are also more concerned about their privacy when disclosing health related information. This provides an ideal setup for our experiment [51, 52].

A summary of the application scenario in the experiment is given below,

"*Think of a web-based health-care application that allows remote consultation with medical professionals, general practitioners and specialists, for a payment. Users should be able to browse through a registered list of medical professionals and chat (text/video) with them on their health problems for advice. Doctors and health-care professionals can register on the application to earn by providing their expertise to users. The application is to be freely available on-line*

(desktop/mobile)."

Since the participants did not have a real application to interact with, it was important that their answers were based on the scenario we described. Therefore, we filtered out participants who had never used health based online applications (mobile/web based) before in the first level of participant filtering to minimize the effect of participants' previous experience on their answers.

We had 54 participants in the experiment, recruited through university notice boards and Github. We recruited 12 participants (6 each in role-play and non role-play) from the university through notice board advertisements with a pre condition that the participants have experience in user application development, as previous research suggest the use of students as a proxy for experiments that require developers is valid [53, 43]. We also recruited 25 software developers who have industry experience through crowd sourcing via Github [43, 44]. We recruited Github users who are committers of PhP and Java open-source git repositories, through invitations emails. We selected PhP and java repositories as we were targeting end user application developers. Answers from one developer was removed due to low quality of the answers. 18 users without development experience were recruited through Facebook advertisements. All participants were compensated with a USA $15 Amazon.com gift voucher for their participation.

We gave participants a brief introduction about ourselves and the study content in the invitation advertisement (an invitation email in the case of github participants) and asked them to express their interest if they are willing to participate. Those who expressed interest (developers following the invitation email and users following the Facebook advertisement), received a second email with an instruction form (guiding them through the study content), participant consent form and study information sheet. Participants were asked to read the study information sheet and sign the consent form before participating, giving us the consent to store and analyze their answers. Software developers were asked to generate a random number of 8 digits at the beginning of the study. Participants who had an odd number were asked to think in terms of a user in

the experiment.

Previous work in psychology has shown role-play as an effective tool to encourage and assess psychological and behavioral traits [18], and in this experiment we employed it as a technique to observe software developers' privacy preferences from a user point of view. This approach of applying a variable to a group of participants is known as the control group method in experimental research. It is known to be a successful way of identifying and comparing behavioral changes due to the varied conditions applied in experiments involving humans [17].

We had three participant groups in the experiment.

Software developers playing the role of a user (Experimenting group): Participants in this group were asked to think of themselves as users of the application.

Software developers (Control group): These participants were asked to follow their normal behavior as an application developer (under normal conditions).

Users WITHOUT Application Development background (Benchmarking group): They were asked to consider themselves as users of the application.

4.1. Questionnaire

We designed the questionnaire following the mixed method research where we use both qualitative and quantitative components as appropriate to derive the results [54, 55]. The primary investigation and result derivation is done using quantitative analysis and qualitative questions were used to further understand and reflect on the quantitative results. We tested our questions extensively in a pilot study through the feedback from four participants (two developers and two users) known to the first author, and not related to the research, to ensure that the questions conveyed the expected meaning. The questionnaire was fine tuned based on their feedback and the final version used for the study is available in the appendix A and B.

The wording of the questions were carefully designed as we wanted all three participant groups to answer similar questions with a different approach according to their role play [56]. Both role-play and non role-play developer groups were exposed to the same questionnaire, worded differently to suite the role play. For example, if we asked the non-role-play developers "what do you think users would expect from this application", we asked the role-play developers "If you use this application, what would you expect from this application?". Users had the same questions worded in a simple way "what would you expect from this application?". We eliminated engineering jargon from all questions in the user version of the questionnaire, acknowledging the lack of understanding of technical terms in users without software development background. Developers' version of the questionnaire had more questions focusing on their development practices and they had a design task in the middle of the questionnaire.

We used a design activity with pre-task and post-task questions for developers in order to investigate their perception on user privacy and their behavior when designing applications while minimizing the limitations due to memorability and recalling capacity of participants of their usual actions [57]. Many studies that focus on software developers' privacy engagement [20, 58, 14] commonly employ online surveys and interviews investigating developers past experience. However, in such approaches even if the developers claim they consider privacy as a very important aspect [59], in reality, either willingly, or driven by design requirements, they may decide to collect user data without much concern towards user privacy. Therefore, the answers may be overshadowed by unconscious privacy paradox. In order to minimize this effect in our study, we first ask the developers about their perceived user privacy expectations. Then, we ask them to design an application with instructions to consider privacy. Following the design task, we ask them how they decided to collect data in their designs and how their perceived user privacy expectations affected their design decisions. This way, rather than asking what they think they would do, or what they think they did in the past, participants are asked about what they just did. Furthermore, by analyzing the designs, we could also observe whether the

developers perception of user privacy requirements they stated earlier on the questionnaire were reflected in their designs.

4.1.1. Pre-design task questions

Before priming the participants to consider privacy in the study we first asked all participants their general expectations in using the application. With this we aimed to observe how users and developers pay attention to privacy in using a potentially privacy invasive software application. With the non role-play developers we aimed to observe whether they recognize privacy as a user expectation for this particular application context.

Then, to evaluate the first research question we used a 10 point Likert scale. Questions that use Likert scales are considered as an effective form to directly investigate preferences and priorities [60]. Since this was more about the level of priority, rather than an agree/disagree situation, we used an even numbered 10 point Likert scale[61]. The Likert scale requested the users and role-play developers to mark the level of priority they would give to their privacy expectations for the application scenario described, with 10 being the highest level of priority and 0 being the lowest. Non role-play developers were asked to mark their assumption on the priority users would give for privacy when using this application.

After this developers were asked to perform a design activity. Users continued to answer the questions in the post task questionnaire.

4.1.2. Application Design Activity

The design task investigate the conscious attention developers paid in embedding privacy into their designs. Developers were asked to perform sketches designing (use case diagram, class diagram, data-base diagram, list of data items to collect and information flow diagram) the application scenario and share the drawings or scanned copies of their sketches. Participants were given the freedom to use hand sketches or software tools they are comfortable with. Information flow diagrams are taken as the way to understand where and how

privacy should be embedded in software development activities [62, 1]. Therefore, the task given here encouraged the participants to think of the complete application design and related privacy aspects. We provided examples for each of the diagrams requested (data-base sketches, information flow diagrams) in the instruction guide to help developers.

Once the developers completed the design task, they were asked to include the 8 digit random number they generated in their sketches and continue with the questionnaire. At the end of the questionnaire they were asked to email us their design sketches. The questionnaire submissions and designs were anonymized before saving and the 8 digit number was used as the identifier to map the sketches with the answers of the developers.

4.1.3. Post-design task questions

Following the design activity, to observe data collection expectations of users against the decisions made by developers we asked the participants (developers and users) to select the data items they used/ expected the application to use from a list of data elements. We decided to use a list because our pilot study participants claimed that having this question as an open ended question is not appropriate, as it would require them to recall all data items they used in the design and state them. Furthermore, we observed that this resulted in inconsistent answers which may not help us to evaluate the data collection decisions. Therefore, we generated a list of data through a pre-investigation. We asked 13 graduate students with software development experience, what data they would collect for the application context we defined. However, in order to give the participants freedom to select data they prefer we also provided space for the participants to state any data element they decided to collect for the application, which was not available in the list.

Finally, in order to gain better insights in to the behavioral aspects we also included simple and direct open ended questions [63, 13]. We made use of the threat avoidance model by Liang et al. [64] in designing the open ended questions, on the basis that users, when exposed to potential threats in using soft-

ware applications would go through a behavioral process for threat avoidance. Developers in the role play group, were expected to demonstrate behaviors for threat avoidance as well as threat coping, as they were expected to be capable of making use of their technical knowledge to cope with the privacy risks they may face as a user in their attempts to design the application.

Developers spent about 3 hours in the design task and 1 hour in the on-line questionnaire, where as users spent an average of 15 minutes to go through the application scenario and 1 hour answering the questionnaire. The complete study design (participant recruitment, experiment design and the questionnaire) was approved by the university research ethic committee responsible for the ethical conduction of research studies that involve humans.

4.1.4. Participants

The following tables (table 2 and 3) depict the age and gender distribution of the participants.

Table 1: Participant Profile

Table 2: Age Distribution

Age	Developers	Users
18-23 years	4	0
24-30 years	25	13
30-34 years	4	3
35-45 years	3	2

Table 3: Gender Distribution

Gender	Developers	Users
Male	28	15
Female	7	3
Prefer not to say	1	0

All the participants without an application development background had at least a bachelor's degree and spent more than 4 hours on the Internet on a daily basis. Following tables (table 5 and 6) give the professional profile and hours of coding per week for the participants with development experience

4.2. Data Analysis

We used the T test (for independent samples) to evaluate the first research question. T test was applied to the answers received for the semantic questions

Table 4: Developers' Demographic Profile

Table 5: Profession

Professional Status	No. of devs
Software employee	21
Freelancer	2
Student	12
Company Owner	1

Table 6: Weekly Hours of coding

Hours	No. of devs
<5 hrs	11
5 to 10 hrs	3
10 to 20 hrs	5
20 to 40 hrs	11
40 hrs <	6

on Likert scale considering two participant groups at a time. We evaluated users against role-play developers, users against non-role-play developers and role-play developers against non-role-play developers.

The second and the third research questions were evaluated using ANOVA [65]. We applied Two factor ANOVA without replication to evaluate the difference against user expectation and developer practice (role-play and non role-play) on data collection across and within the three participant groups. We considered sensitive and not-so sensitive data items, defined according to data protection regulatory definitions [37, 38] separately in this analysis.

We then performed thematic analysis on the descriptive answers received for the open ended questions. We followed the coding approach [13] to identify reasons for the behaviors observed [66]. We did not receive long answers for the open ended questions. The longest answer we received was three sentences. Therefore, the coding process was simple and straight forward. One coder first went through each and every answer and generated simple codes summarizing the answers. With these codes we generated a coding scheme in Nvivo. We then recruited another coder and generated another coding scheme. Both coding schemes were similar and interchangeable. Hence, the first coder then coded all answers using the coding scheme on Nvivo [67]. This process is known as grounded approach in qualitative data analysis.

5. Results

Figure 2 below captures the key findings of our study and demonstrates the relationship among developers personal privacy expectations as users, their assumptions on user privacy expectations and their behaviors when they designed the software applications. We indicate a significant relationship with the hard arrows and an observed relationship with the dashed arrows.

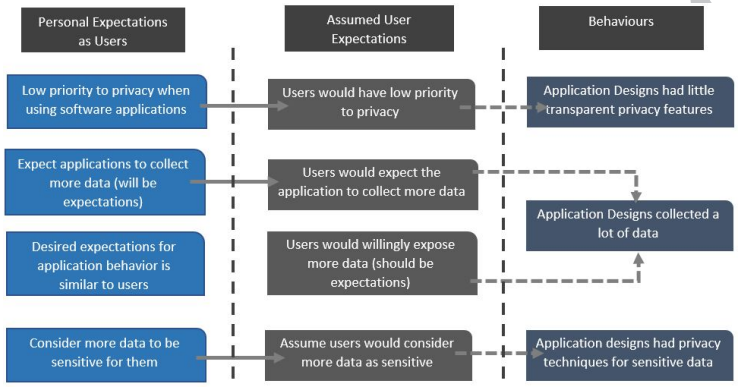

Figure 1: Relationship among developers personal privacy expectations as users, their assumptions on user privacy expectations and their behaviors. The hard arrows indicate a significant relationship and the dashed arrows indicate an observed relationship

We could observe that developers assumptions on user privacy expectations relate to their personal privacy expectations as users. For example, compared to users developers expected applications to collect more data and they assumed users to be comfortable sharing more data into the application. Developers also assumed that users would have lower priority towards privacy when using the application. We could observe effects of these assumptions in their designs. However, interestingly, developers and users had similar expectations on desired data collection (ideal behavior) in software applications. Nevertheless, this had no effect on the behavior of the developers. In the next sections we discuss the present our results in detail and answer the research questions.

5.1. Expectations, developers vs users

When we first asked the participants about their expectations in using the application role-play developers mentioned key words such as, "speed". "efficiency" and "instant service". Non-role-play developers used key words such as "efficiency", "convenience", "cheap" and "remote service" and users used key words such as, "lower cost", "easy", "fast","avoid unnecessary expenses" and "instant service".

However, surprisingly, 8 out of of the 18 (44.4%) role-play developers also mentioned "security" and "confidentiality" as expectations, which were not observed in the answers of the users. None of the developers in the non role-play group assumed confidentiality or privacy as a user requirement either. We find this observation interesting. This suggests that developers when using software applications consider privacy as a requirement even if they are not consciously encouraged to consider privacy. This may be because developers unconsciously combine their experience as a software developer into their expectations when they use software applications, because they know how software applications are designed and how the applications may behave.

However, only one out of the 18 (0.05%) users mentioned that s/he would not use the application due to the sensitivity of data that would be required by the application. This suggests that privacy is not a requirement that generally comes to users' mind when they are asked about their expectations from a software application. Nevertheless, our next result indicate that if asked directly, users have high privacy expectations when using software applications.

5.2. How different is the level of priority given to privacy between developers and users when they use software applications?

The following graph (Figure 01) shows the answers we received from the three participant groups. The level of priority users gave for privacy had a mean = 9.56 (st.d= 1.042). Role play developers' level of priority for privacy when using the application had a lower mean = 8.333 (st.d= 1.81497). Non role-play developer group's assumption on the priority users would give for privacy

had the same mean with a different standard deviation, mean = 8.333 (st.d= 1.029).

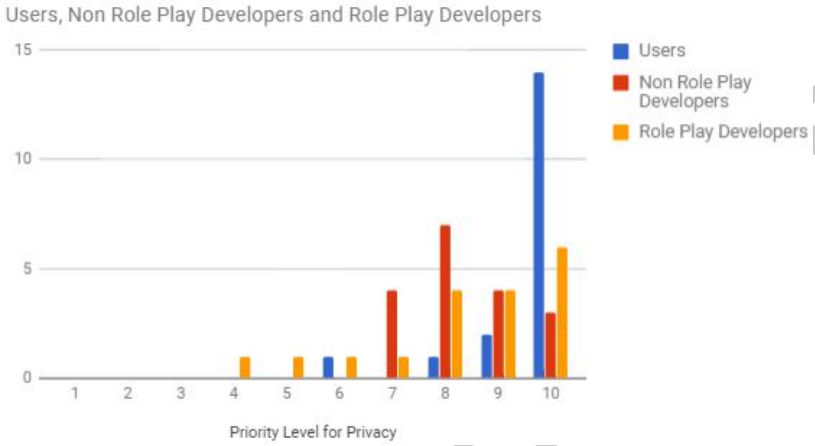

Figure 2: privacy expectations of users vs developers in the three groups

There was a significant difference in the level of privacy expectation of the developers who were performing role-play against users ($p = 0.0197 < 0.05$). This suggests that application developers give less priority to privacy expectations compared to users when using software applications. Developers assumptions on how important privacy would be for users in using this application was also significantly different to that of users ($p = 0.00059 < 0.001$). However, developers' assumptions were similar to what developers in role-play expressed as their own priority if they were to use the application ($p = 0.5 > 0.05$). This suggests that developers may relate to their own expectations as a user when they assume user privacy expectations.

5.3. How different user expectations are from the actual data collection practices developers embed into software applications?

When it comes to data collection expectations, we focus on both types of expectations defined by Miller [35]. The *should be* expectations, which corresponds to what the user desire the ideal setup to be as well as the *will be* expectations,

which corresponds to what the user accept as the reality, irrespective of their ideology. For example, here we ask what data users would willingly expose to the application, which corresponds to their *should be* expectations or the desired data collection. That is, what they believe the application should collect in ideal behavior. Then we ask what data they think the application would collect, which corresponds to their *will be* expectations. Because, how they think the application would behave, may or may not be similar to their ideal expectation. However, sometimes even if it is against their ideal expectations, users may expose data to an application due to previous experience with similar applications and their general knowledge on the current status of data practices in software applications. This corresponds to the gap from *should be* expectations to *will be* expectations. Focusing on this, we also ask the participants what data they would unwillingly expose to the application

Table 7 captures the responses we received. For ease of presentation we have summarized data into categories such as contact information where the participants gave answers such as address, telephone no. etc. Personal information implies name, marital status, age and behavioral data implies their hobbies, food they eat, smoking/drinking habits and exercise routines.

We observed that the desired application behavior of role-play developers (*should be* expectations) was somewhat close to that of users. For example, similar to users, role-play developers stated that they did not desire the application to collect their personal information, financial information and contact details. This was quite different from what non role-play developers assumed. They only assumed that users would not be willing to expose their behavioral data into the application. Such mismatches in developer assumptions against user expectations on desired application behavior may result in user frustration which leads to users' dissatisfaction with the application. This may lead to users discontinuing to use the application, or continuing to use it with dissatisfaction due to the benefits the application gives or unavailability of other options.

However, assumptions of developers on users' *will be* expectations that are different from the actual expectations of users may result in privacy invasions.

Table 7: Mismatched Expectations on Data Collection

Participant Group	**Willingly Expose as a User/ Users would Willingly Expose**	**Unwillingly Expose as a User/ Users would Unwillingly Expose/**	**Expect the Application would Collect/ Users would expect the Application would Collect**
Users	Personal Information, Medical History	Contact Information, Educational, Occupational or Financial and Insurance Information, Location, Browsing History and Past Interactions with the application	Browsing History, Health issues and Conditions, Behavioral data, Location, Insurance Information
Role-play developers as users	Medical background, Demographic Data, Location	Personal Information, Financial Information, Payment Details, Contact Details	Personal Information, Medical background, Contact Details, Payment Details
Non-Role-Play developers	Demographics, Medical History, Identification Data, Financial Data, User Feedback, Location Data, Past Interactions with the Application, Payment Data, Photograph	Behavioral data	Personal Information, Medical History, Identification Data, Financial Data, User Feedback, Location Data, Contact Details

For example, our results indicate that developers assume users would expect the application to collect their contact details (such as the mobile number, email address etc), which is different to users' *will be* expectations. However, based on their assumptions, developers may design the application to collect users' contact details without explicit notifications, and this could lead to invading user privacy, because users do not expect the application to collect these data from them. Users are known to interact with software applications based on how they think the application would behave [8]. Therefore, the consequences of mismatched *will be* expectations are critical in terms of privacy. Accordingly, in the next section when we evaluate how the data collection expectations differ among the three participant groups, we only considered the *will be* expectations and disregarded their ideal expectations.

Following the design task we asked the participants the data items they decided/expected the application to collect. We first evaluated the data non role-play developers decided to collect for the particular application against the selection of the users as to what they thought the application would collect. Then, we evaluated the data elements role-play developers decided to collect against that of non role-play developers. Table 8 depicts the results from the participants.

Table 8: Data Element List

Data Element	devs as devs	devs as users	users
Name	7	9	10
Age	10	11	11
Birth-date	4	10	6
Gender	12	11	12
Marital Status	5	4	7
Sicknesses over the last 2 yrs	6	9	9
Sicknesses over the last 5 yrs	1	0	0
Sicknesses over the last 10 yrs	5	2	0

Occupation	6	4	0
Education Level	1	2	0
Drinking smoking habit	9	6	0
Blood type	6	8	0
MRI/ECG Reports	7	4	0
Blood/urine reports	7	4	0
Address	4	4	4
Most Preferred Hospital	3	0	1
Most Preferred Doctor	3	0	0
Hobbies	2	4	2
Exercise routine	5	7	9
Previous Medicines Prescribed	8	6	0
Race	3	4	5
Nationality	3	7	5
Sexual Preferences (LGBT)	3	3	2
Critical Health Conditions	10	11	10
Health Conditions in the family	7	4	9
Food Preferences	5	6	6
Heart Rate (Regularly)	1	4	0
Weight/Height	6	9	10
Current Medicines	6	6	0
Photograph	3	0	6
Sleeping Hours	5	6	7
Discussions with Doctors	4	3	4
Patient's Feedback on Doctors	8	6	8

When evaluating the results we categorized data into sensitive data and not so sensitive data based on the definitions in the USA FTC [38] and the

Australian Privacy Act [37]. Table 9 shows the two data groups (sensitive data and not so sensitive data) after the classification.

Table 9: sensitive data and not so sensitive data

sensitive data	not so sensitive data
sicknesses in the last 2 yrs, sicknesses in the last 5 yrs, sicknesses in the last 10 yrs, blood type, MRI/ECG/CTC reports, blood/urine reports, previous medicines prescribed, race, nationality, sexual preferences (LGBT), prevailing health conditions, health conditions running in the family, daily heart rate, discussions with doctors in the application, photograph and medicines taken currently	Name, Age, Birth-date, Gender, Marital Status, Occupation, Education Level, Drinking smoking habit, Address, Most Preferred Hospital, Most Preferred Doctor, Hobbies, Exercise routine, Food Preferences, Heart Rate (Regularly), Weight/Height, Sleeping Hours, Discussions with Doctors and Patient's Feedback on Doctors

Interestingly, there was no significant difference in the collection expectations of users against actual developer practices when it comes to not so sensitive data (developers, $p = 0.89 > 0.5$, role-play developers $p = 0.79 > 0.5$). However, when it comes to sensitive data, there was a significant difference in what users expect the application to collect, against what developers actually decide to collect (developers, $F = 6.15$, $F\ crit = 4.60$, $p = 0.02 < 0.05$, role-play developers $F = 6.79$, $F\ crit = 4.60$, $p = 0.02 < 0.05$). More importantly, role-play had no effect on this difference and developers from both groups demonstrated a significant difference with their data collection practices against user expectations. Furthermore, the data collection decisions (sensitive and not so sensitive data together) made by developers were strongly dependent on the developer and hence non consistent (Cronbach's alpha for consistency among the answers = 0.109). This suggests that the data collection practices of software applications

are dependent on the developer and the same application being developed by different developers with the same specification, without explicit instructions on the data items to be collected, would be inconsistent in data collection.

5.4. How different users and developers perceive sensitive data when using/developing software applications?

Our last research question focused on what users and developers perceive as sensitive information when they use and/or develop an application respectively. For this, users and role-play developers were asked to select data they considered to be sensitive for them, and non role-play developers were asked to select data items they assume to be sensitive for users. For this particular context we defined sensitivity to be the impact of loss of data. Table 10 shows the results.

We observed that none of the users considered age, gender and birth date to be sensitive data where as developers considered these data elements to be sensitive to the user. Interestingly, users did not consider their credit card data to be sensitive to them where as developers considered it to be so. Furthermore, compared to the data elements identified as sensitive by data protection regulations, users did not consider nationality or race to be sensitive to them and only one user considered personally identifiable data such as name and address to be sensitive. Most users (12 out of 18) considered health conditions and medicines taken to be sensitive information.

Interestingly, developers identified all the data items identified in the regulations and some more to be sensitive to the user. The data elements users considered to be sensitive was a subset of the data developers in both role-play and non-role-play groups considered to be sensitive. This implies that developers consider more data elements to be sensitive than what users actually consider to be sensitive. This may be because developers are more aware of the privacy risks that exist in software applications and hence have a better understanding on the impact of loss of data, whereas users due to lack of knowledge and experience, do not consider some data items such as their name and address to be

Table 10: Identifying Sensitive Data

Data Element	devs as devs	devs as users	users
Previous and current health conditions	12	7	12
name	7	3	1
address	8	1	1
credit card	5	2	1
medicines taken (current and previous)	7	3	10
Family health conditions	4	0	1
habits (drink/smoke)	2	3	7
Height/weight	3	1	1
pulse/heart rate	1	1	1
nationality/race	4	1	0
LGBT preferences	3	3	1
Occupation/income level	1	0	1
Age	7	5	0
Gender	5	6	0
Blood Group	3	0	0
Birth date	3	2	0
Test Reports	0	1	0

sensitive to them. An ANOVA test across the three groups on all data elements demonstrated a significant difference across the three groups ($F = 4.13$, $F\ crit = 1.97$, $p = 0.003 < 0.005$).

Nevertheless, it should be noted that there was a disparity among developers on sensitive data elements, and their decision on sensitive data was not consistent. Developers varied significantly among themselves when they were asked to decide whether a particular data item would be sensitive to a user or not. Users were more consistent in their selection as almost all of them identified health conditions, medicines taken and habits to be sensitive. Among developers majority of them agreed only on the sensitivity of medical conditions. An ANOVA test among the individual developers on the identified data elements demonstrated a significant difference ($F = 6.03$, $F\ crit = 3.29$, $p = 0.006 < 0.01$).

5.5. Developers' behaviors

Interestingly, there was no significant effect of role-play in the design decisions made by the participants (ANOVA $F = 1.71$, $F\ crit = 6.61$, $p = 0.25$).

When observing developers' behaviors we particularly focused on how the participants had used privacy techniques such as anonymization, pseudonimization, data aggregation, data separation, encryption and data expiry in storage. Most developers had integrated these techniques when storing the data they considered sensitive for users (Aggregation = 23, Separation = 25, Anonymization = 20, Pseudonimization = 11, Data Expiry (Storage period) = 18, Encryption = 22, out of 36). Developers were fairly consistent in embedding these privacy techniques into their designs (Cronbach's alpha = 0.801). Since developers were allowed to go back and forth between the questionnaire and the designs, it is possible that developers made changes to their designs after answering the question on sensitive data.

However, there were two participants who claimed that database storage of user data without encryption cannot have any effect on user privacy. One participant had claimed that anonymous storage would hinder the benefits they

could gain by processing data. Most of those who did not use these techniques claimed that either they did not know the concept or that they did not think of it at the time of designing. For example, one participants said,

"I did not think of aggregation, although this is a good strategy for ensuring anonymity of data when, say, being transferred to a data warehouse".

This suggests that developers require explicit instructions to embed privacy into their application designs. General explicit instructions may not be sufficient in guiding them to decide on which privacy techniques they should consider and why. Consequently, developers had little concern towards user privacy expectations in their designs. None of the participants had integrated any privacy notifications into their designs. When instructed to consider privacy in their designs it was apparent that developers intuitively focused on technical aspects and not user privacy expectations. We could not observe any implicit effects of the role play developers performed in their designs. None of the role play participants displayed concerns on their personal preferences on desired application behavior or *will be* expectations in their designs. Overall,This suggests that explicit instructions are required to encourage developers to consider mismatched expectations of users against the application behaviors when they design software applications.

However, since developers had used privacy techniques on data inconsistently (for example, encryption was available for some sensitive data, and not for some) and due to lack of accepted formal methods to evaluate the privacy embedded in a system design, we are unable to claim any significant relationship between developers perceived user privacy expectations against their design decisions. Therefore, as demonstrated in Figure 2, here we only claim observed relationships.

5.6. Effectiveness of Role-Play

Finally, to understand the effectiveness of the role-play approach we asked developers in the role-play and non role-play groups about using their skills and expertise as a developer to improve the application.

Role play developer suggested implementing a methodology for doctors to retrieve sensitive information on demand to reduce the time spent on consultation and incorporate machine learning to suggest common sicknesses. Non role-play developers suggested recording video during consultation in the background to identify mental health issues, introducing machine learning to predict common illnesses, storing consultation records to help future consultations and making use of users' browsing history to see their current health situations.

Despite the observed differences in the privacy expectations of role play developers from actual user expectations, this result suggests that developers may make an effort to utilize their knowledge in implementing privacy preserving technologies in the application better when they are playing the role of a user, as role-play developers were surprisingly privacy concerned. Role play developers demonstrated an interest to make use of their technical knowledge to cope with potential threats that would affect user privacy. They also did not suggest potentially privacy invasive technologies such as accessing user browsing history or consultation recordings. The study evaluating conscious effort from the participants towards privacy may have had an effect on this. However, if that is the case, the effect would be similar on both role-play and non-role play developers and hence could be disregarded.

Therefore, even though it is not 100% effective in enabling developers to understand user privacy expectations, we may still consider the role play approach as a potential way to nudge developers towards user privacy concerns. This is further highlighted by the skewness measurement of the priority given for privacy by the three participant groups. Users were highly skewed towards highest level of priority (skewness 2.80) and developers performing role-play had a skewness of 1.15 which is still significant, whereas developers who did not perform role-play had a skewness of -0.32. While the level of privacy embedded into the design while playing the role of a user may not quite be the same as what a user would expect, the results suggest that the role play approach may encourage developers to make an effort to embed privacy into the applications they develop better.

To further explore how developers own expectations of privacy interact with the way they embed privacy into software applications, we asked role-play developers how possible they believe it is to address their own privacy needs as users if they developed this application. 6 out of the 18 role-play developers said it is not possible to address their privacy expectations.

"*as a user I won't prefer app saving details, but I know [it] is a primary requirement here*" P18.

"*as developers we give priority to client business requirements as oppose to user requirements in scenarios where the two does not get along*" P13.

and,

"*as a developer, when I design applications I tend to keep [my] privacy concerns in the back seat because my own privacy interests sometimes conflict with client requirements*" P03.

All of these 6 participants believed that according to their experience, their own privacy requirements and business and client requirements do not get along.

Interestingly, 4 out of the 18 role-play participants said that it is possible to address user privacy needs, however, they said that it would be time consuming and expensive due to which it is not practical. Further, they mentioned that developers' voice is not given much attention in solving conflicts on user privacy requirements in the design phase and hence, this should be focused at the management level. The rest of the participants (8 out of 18) said that they are not sure.

6. Discussion

6.1. Perceived Expectations vs Reality in Software Development

Overall our findings indicated a significant difference between the perceived user privacy expectations by developers against actual user privacy expectations. The differences we observed in developers' perceived privacy expectations from users had an impact on the way they designed the applications. Previous

research has identified that the lack of knowledge and understanding on information privacy in developers act as a deterrent when developers embed privacy into the software applications [13]. Our findings indicate that in addition to the knowledge on information privacy, developers also need to understand the user expectations on privacy for successfully embedding privacy into software applications. However, for this developers first need to acknowledge the differences between their perception against actual user expectations. For example, the prejudice developers have, where they believe they know what users want from an application in terms of privacy [16] can have detrimental consequences on application designs where developers' perceived privacy expectations are observed to be significantly different from users.

In the current setup of software development, developers are expected to be responsible in embedding privacy into the applications they design [34, 16, 68, 6]. Most privacy technologies and conceptual guidelines such as PbD [27], PIA [29] and FIP [28] are heavily dependent on the capability of the developer to make the right choice at the right time. For example, these methodologies guide developers to use privacy technologies in embedding user privacy requirements such as non-tracking Internet searching, dis-joint storage of information and aggregated data storage to avoid personal information tracking [69] into software applications. Onion Routing, Crowds, Gaps are some of the commonly used privacy technologies in engineering environments [69] that focus on enabling user privacy requirements through privacy concepts such as anonymity, untraceability and un-observability of users from their internet actions. However, these methodologies, or rather tools, solely focus on system implementation aspects [70]. They are dependent on the developers' perception of user privacy requirements in order to decide why, how and when to use them in a design [69]. However, since our findings indicate developers' perceived user expectations to be very different from that of users, developers may required to be systematically guided to explicitly elicit user privacy requirements in order to effectively use these privacy technologies in their software designs.

6.2. Enabling developers to understand user privacy expectations

Overall, the results suggest that developers are aware of the existence of privacy vulnerabilities in software applications. This was also evident from their lowered privacy expectations from software applications. Consequently, it is important that developers realize their personal privacy expectations are lower than that of users, in order to avoid these lowered expectations affecting their development activities. Embedding privacy education into the curriculum in IT education [71] could be one long term approach to develop the mindset required by developers to understand and acknowledge user privacy requirements from an early stage itself. Values and knowledge embedded in the early stage of career may assist them when they grow as developers and hence improve the privacy knowledge and interest within the community of software developers. However, in order to address the need to enable developers to successfully address user privacy expectations in their software application designs, it is important that developers are systematically guided to understand and embed privacy within the development practices itself [72]. This is emphasized in our findings where developers claim that they have little room and authority to embed privacy into the softwares in their tasks. Management attention together with developer education on privacy matters may assist the development of privacy aware software applications.

Nevertheless, so far, no systematic methodology has been proposed to define how developers should attempt to identify user privacy expectations [13, 11, 12]. Our findings suggest that "Role-play as a user" may nudge software application developers to be more concerned about privacy in designing and developing software applications. This would encourage developers to think from a user point of view, which was shown to have favorable effects on considering privacy when developing software applications. However, as developers own expectations of privacy are observe to be significantly different from user expectations, more influential strategies are needed to guide developers to understand the differences in user expectations compared to their own.

Interaction with users and formally extracting user privacy expectations

through surveys and user studies may be required especially when designing sensitive applications similar to the scenario used in our study. Implicit methods that may encourage developers to assume user privacy expectations should not be encouraged due to the obvious reason of misunderstanding user expectations. Participatory design, cognitive walk through [73] and task action grammar [74] are some existing methods that enable developers to extract mismatched user expectations when designing applications through direct interaction with users. Interactive prototypes is another approach that is used to directly engage users to avoid false assumptions of developers affecting software application designs [75]. However, as our results indicate that users do not express privacy as a requirement unless explicitly asked, it is important that privacy is considered as an explicit conscious component in these methodologies when extracting user expectations.

7. Limitations

This study focus on two factors that affect developers perception on user privacy expectations, which are developers' personal privacy expectations and developers' assumptions on user privacy expectations. We, by no means rule out the possibility of the existence of other factors that could affect developers perception of end user privacy expectations, which are currently not known. However, the focus of our study design is explicitly on these two factors and we encourage future research to identify more factors, if any, that may affect the perception of software developers on user privacy expectations. Furthermore, in forming the research questions we focus on the priority given to privacy when using software applications, data collection expectations and expectations on sensitive data. We accept that there are many aspects of data (deletion, saving, processing) that affect transparency in software applications. However, our focus here is limited to the above three areas due to the lack of focus in previous work given to these areas in particular. For a complete analysis, it is important that the differences in perceived and actual user privacy expectations

are observed in the complete data processing chain in software applications. We encourage future studies to focus on these areas to broaden our findings and see how our findings stand on these aspects.

We chose a remote evaluation for this study because it allowed us to recruit a geographically diverse sample [44]. Although we are aware that a remote evaluation gives us less control over the study environment, we still believe using a remote evaluation had a minimal effect on the main findings of this research because we were interested on the participants thought process according to their role play. The questionnaire was carefully designed to elicit details about their thought process and the difference in their behaviors, and we have discussed any impact due to remote investigation in the results wherever applicable. Further, allowing the developers to participate in the study in an environment they are comfortable with [44] supported their original behavior in contrast to a controlled environment. As we carefully evaluated the results against the designs submitted by the participants we were able to validate their answers. We found that the answers were descriptive and valid. For example, the data items developers said they decided to collect were accurate against their designs except for few false positives where developers had not mentioned all data they collected. Therefore, we suggest that for similar studies that observe participant behavior, remote studies that are carefully designed can be considered appropriate.

As a result of having a remote study outside of an organizational setting of actual software development, the study assigned an artificial task for the developers. We kept the task simple and straight forward to ensure that the participants would not get bored, which would affect their participation with genuine interest, and yet complete to ensure that all the steps in an application design would be thought of within the task. However, we accept that an artificial task may not initiate the exact same enthusiasm or rather, commitment in developers towards the task compared to their professional commitment to their work. However, since we are using the control group method by keeping the factors other than the investigation parameters to be similar for other groups was sufficient in this experiment setup. Furthermore, since privacy is a sensitive

topic for software development companies that deal with user information, it is unlikely that they would allow researchers to observe their developers when they make decisions to collect user information for an application they are developing within the organization. Therefore, the task based questionnaire approach we employed here was the closest we could get to observe software developers original behavior in consciously considering privacy in their development tasks to elicit their perception of user privacy expectations and data collection, and we consider this approach to be appropriate for deriving recommendations.

8. Conclusion

This study attempts to understand developers' perception on user privacy requirements when they design software applications. We used three participant groups in the study with software developers and users. Our findings indicate significant differences in the level of priority given for privacy among users and developers when they use software applications. The study also show that user expectations on data practices on software applications are significantly different from what developers assume users would want. Interestingly, developers' assumptions on user privacy expectations were close to their own privacy expectations as users. Furthermore, the study also revealed that the behaviors of developers when designing software applications was influenced by their assumptions on user privacy expectations. However, we observed that without explicit instructions developers do not consider user privacy expectations when embedding privacy into their designs.

Based on our findings we propose recommendations for user privacy requirement elicitation for software application development. We suggest embedding the importance of understanding user privacy requirements into the curriculum in IT education and making developers understand the importance of interacting with actual users through systematically guided methods to extract user privacy requirements when developing software. Our findings would be important for organizations and policy makers to gain an understanding as to how

they should implement and enforce organizational practices on software developers. Further research on how the observed differences could be addressed in organizational software development practices and how developer behavior in organizations software development relate to these findings would help to see how our findings stand in those setups. How often developers assume user privacy requirements, and how software requirement elicitation methods should incorporate user privacy requirements would be interesting avenues to continue our initiation.

References

[1] R. Kang, L. Dabbish, N. Fruchter, S. Kiesler, my data just goes everywhere:âĂİ user mental models of the internet and implications for privacy and security, in: Symposium on Usable Privacy and Security (SOUPS), USENIX Association Berkeley, CA, 2015, pp. 39–52.

[2] T. Vidas, N. Christin, L. Cranor, Curbing android permission creep, in: Proceedings of the Web, Vol. 2, 2011, pp. 91–96.

[3] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, D. Wagner, Android permissions: User attention, comprehension, and behavior, in: Proceedings of the Eighth Symposium on Usable Privacy and Security, ACM, 2012, p. 3.

[4] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, J. Zhang, Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 501–510.

[5] S. Lederer, J. I. Hong, A. K. Dey, J. A. Landay, Personal privacy through understanding and action: five pitfalls for designers, Personal and Ubiquitous Computing 8 (6) (2004) 440–454.

[6] A. Adams, M. A. Sasse, Privacy in multimedia communications: Protecting users, not just data, in: People and Computers XVâĂŤInteraction without Frontiers, Springer, 2001, pp. 49–64.

[7] J. R. Reidenberg, N. C. Russell, A. J. Callen, S. Qasir, T. B. Norton, Privacy harms and the effectiveness of the notice and choice framework, ISJLP 11 (2015) 485.

[8] A. Rao, F. Schaub, N. Sadeh, A. Acquisti, R. Kang, Expecting the unexpected: Understanding mismatched privacy expectations online, in: Symposium on Usable Privacy and Security (SOUPS), 2016.

[9] K. M. Ramokapane, A. Rashid, J. M. Such, âĂIJi feel stupid i canâĂŹt delete...âĂİ: A study of usersâĂŹ cloud deletion practices and coping strategies, in: USENIX Symposium on Usable Privacy and Security (SOUPS), 2017, pp. 241–256.

[10] C. Kalloniatis, E. Kavakli, S. Gritzalis, Addressing privacy requirements in system design: the pris method, Requirements Engineering 13 (3) (2008) 241–255.

[11] I. Omoronyia, L. Cavallaro, M. Salehie, L. Pasquale, B. Nuseibeh, Engineering adaptive privacy: on the role of privacy awareness requirements, in: Proceedings of the 2013 International Conference on Software Engineering, IEEE Press, 2013, pp. 632–641.

[12] K. Thomas, A. K. Bandara, B. A. Price, B. Nuseibeh, Distilling privacy requirements for mobile applications, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 871–882.

[13] I. Hadar, T. Hasson, O. Ayalon, E. Toch, M. Birnhack, S. Sherman, A. Balissa, Privacy by designers: software developers privacy mindset, Empirical Software Engineering (2017) 1–31.

[14] O. Ayalon, E. Toch, I. Hadar, M. Birnhack, How developers make design decisions about users' privacy: The place of professional communities and organizational climate, in: Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, ACM, 2017, pp. 135–138.

[15] S. Lahlou, M. Langheinrich, C. Röcker, Privacy and trust issues with invisible computers, Communications of the ACM 48 (3) (2005) 59–60.

[16] D. D. Caputo, S. L. Pfleeger, M. A. Sasse, P. Ammann, J. Offutt, L. Deng, Barriers to usable security? three organizational case studies, IEEE Security & Privacy 14 (5) (2016) 22–32.

[17] M. M. Pithon, Importance of the control group in scientific research, Dental press journal of orthodontics 18 (6) (2013) 13–14.

[18] A. S. Bellack, M. Hersen, D. Lamparski, Role-play tests for assessing social skills: Are they valid? are they useful?, Journal of Consulting and Clinical Psychology 47 (2) (1979) 335.

[19] J. Gluck, F. Schaub, A. Friedman, H. Habib, N. Sadeh, L. F. Cranor, Y. Agarwal, How short is too short? implications of length and framing on the effectiveness of privacy notices, in: Symposium on Usable Privacy and Security (SOUPS), 2016.

[20] S. Sheth, G. Kaiser, 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, ACM, 2014, pp. 859–870.

[21] S. G. Barsade, L. Ramarajan, D. Westen, Implicit affect in organizations, Research in organizational behavior 29 (2009) 135–162.

[22] G. Wurster, P. C. van Oorschot, The developer is the enemy, in: Proceedings of the 2008 workshop on New security paradigms, ACM, 2009, pp. 89–97.

[23] P. Guarda, N. Zannone, Towards the development of privacy-aware systems, Information and Software Technology 51 (2) (2009) 337–350.

[24] M. C. Oetzel, S. Spiekermann, A systematic methodology for privacy impact assessments: a design science approach, European Journal of Information Systems 23 (2) (2014) 126–150.

[25] A. Rao, F. Schaub, N. Sadeh, What do they know about me? contents and concerns of online behavioral profiles, PASSAT 2014.

[26] A. Acquisti, L. Brandimarte, G. Loewenstein, Privacy and human behavior in the age of information, Science 347 (6221) (2015) 509–514.

[27] Cavoukian, pbd, http://ewh.ieee.org/conf/sge/2012/p9-2012-08-29%20-%20IEEE%20(Final), accessed: 2016-09-22 (2013).

[28] D. J. Solove, M. Rotenberg, P. M. Schwartz, Information privacy law, 2010, 4th edition.

[29] R. Clarke, Privacy impact assessment: Its origins and development, Computer law & security review 25 (2) (2009) 123–135.

[30] S. Gürses, C. Troncoso, C. Diaz, Engineering privacy by design, Computers, Privacy & Data Protection 14 (3).

[31] S. Spiekermann, The challenges of privacy by design, Communications of the ACM 55 (7) (2012) 38–40.

[32] C. Yang, P. Liang, P. Avgeriou, Assumptions and their management in software development: A systematic mapping study, Information and Software Technology.

[33] S. S. Shapiro, Privacy risk analysis based on system control structures: Adapting system-theoretic process analysis for privacy engineering, in: Security and Privacy Workshops (SPW), 2016 IEEE, IEEE, 2016, pp. 17–24.

[34] Y. Liu, K. P. Gummadi, B. Krishnamurthy, A. Mislove, Analyzing facebook privacy settings: user expectations vs. reality, in: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, ACM, 2011, pp. 61–70.

[35] J. A. Miller, Studying satisfaction, modifying models, eliciting expectations, posing problems, and making meaningful measurements, Concep-

tualization and measurement of consumer satisfaction and dissatisfaction (1977) 72–91.

[36] B. Krishnamurthy, C. E. Wills, On the leakage of personally identifiable information via online social networks, in: Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 7–12.

[37] Australian Data Privacy Law, http://www.legislation.vic.gov.au/domino/web_notes/ldms/pubstatbook.nsf/f932b66241ecf1b7ca256e92000e23be/05CC92B3F8CB6A6BCA257D4700209220/$FILE/14-060aa%20authorised.pdf, accessed: 2017-06-23 (2016).

[38] Data protection in the United States: overview, https://uk.practicallaw.thomsonreuters.com/6-502-0467?transitionType=Default&contextData=(sc.Default)&firstPage=true&bhcp=1f, accessed: 2017-06-30 (2016).

[39] European Union data protection, http://ec.europa.eu/justice/data-protection/data-collection, accessed: 2017-06-23 (2016).

[40] G. Steinke, Data privacy approaches from us and eu perspectives, Telematics and Informatics 19 (2) (2002) 193–200.

[41] W. G. Voss, European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting, The Business Lawyer 72 (1) (2017) 221–233.

[42] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, S. Schiffner, Privacy and data protection by design-from policy to engineering, European Union Agency for Network and Information Security.

[43] D. Wermke, M. Mazurek, Security developer studies with github users: Exploring a convenience sample, in: Symposium on Usable Privacy and Security (SOUPS), 2017.

[44] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. L. Mazurek, C. Stransky, Comparing the usability of cryptographic apis, in: Proceedings of the 38th IEEE Symposium on Security and Privacy, 2017.

[45] A. Brush, M. Ames, J. Davis, A comparison of synchronous remote and local usability studies for an expert interface, in: CHI'04 Extended Abstracts on Human Factors in Computing Systems, ACM, 2004, pp. 1179–1182.

[46] A. Kittur, E. H. Chi, B. Suh, Crowdsourcing user studies with mechanical turk, in: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, 2008, pp. 453–456.

[47] F. J. Fowler, Improving survey questions: Design and evaluation, Vol. 38, Sage, 1995.

[48] J. K. Kumekawa, Health information privacy protection: crisis or common sense, Online Journal of Issues in Nursing 6 (3).

[49] M. Hussain, A. Zaidan, B. Zidan, S. Iqbal, M. Ahmed, O. Albahri, A. Albahri, Conceptual framework for the security of mobile health applications on android platform, Telematics and Informatics.

[50] Reuters, Sep 24, 2014 privacy officials worldwide press google about glass, https://bits.blogs.nytimes.com/2013/06/19/privacy-officials-worldwide-press-google-about-glass/?mcubz=0, accessed: 2017-05-25 (2014).

[51] B. Kaplan, Selling health data: de-identification, privacy, and speech, Cambridge Quarterly of Healthcare Ethics 24 (3) (2015) 256–271.

[52] T. Glenn, S. Monteith, Privacy in the digital world: medical and health data outside of hipaa protections, Current psychiatry reports 16 (11) (2014) 494.

[53] I. Salman, A. T. Misirli, N. Juristo, Are students representatives of professionals in software engineering experiments?, in: Proceedings of the 37th

International Conference on Software Engineering-Volume 1, IEEE Press, 2015, pp. 666–676.

[54] A. Naiakshina, A. Danilova, C. Tiefenau, M. Herzog, S. Dechand, M. Smith, Why do developers get password storage wrong? a qualitative usability study, ACM Conference on Computer and Communications Security (CCS)(2017), (To appear).

[55] A. J. Onwuegbuzie, N. L. Leech, On becoming a pragmatic researcher: The importance of combining quantitative and qualitative research methodologies, International journal of social research methodology 8 (5) (2005) 375–387.

[56] W. A. Belson, The design and understanding of survey questions, Gower Aldershot, 1981.

[57] Y. Acar, S. Fahl, M. L. Mazurek, You are not your developer, either: A research agenda for usable security and privacy research beyond end users, in: IEEE Cyber Security Development Conference, (IEEE Secdev), IEEE, 2016.

[58] R. Balebako, A. Marsh, J. Lin, J. I. Hong, L. F. Cranor, The privacy and security behaviors of smartphone app developers.

[59] S. Kokolakis, Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon, Computers & Security 64 (2017) 122–134.

[60] R. L. Rosnow, R. Rosenthal, Beginning behavioral research: A conceptual primer, Prentice-Hall, Inc, 1996.

[61] G. Albaum, The likert scale revisited: an alternate version, Journal of the Market Research Society 39 (2) (1997) 331–332.

[62] K. Wuyts, R. Scandariato, W. Joosen, Lind (d) un privacy threat tree catalog.

[63] B. Skeggs, S. Yuill, The methodology of a multi-model project examining how facebook infrastructures social relations, Information, Communication & Society 19 (10) (2016) 1356–1372.

[64] H. Liang, Y. Xue, Avoidance of information technology threats: a theoretical perspective, MIS quarterly (2009) 71–90.

[65] B. G. Tabachnick, L. S. Fidell, Experimental designs using ANOVA, Thomson/Brooks/Cole, 2007.

[66] A. Strauss, J. Corbin, et al., Basics of qualitative research, Vol. 15, Newbury Park, CA: Sage, 1990.

[67] J. Saldaña, The coding manual for qualitative researchers, Sage, 2015.

[68] M. Howard, S. Lipner, The security development lifecycle, Vol. 8, Microsoft Press Redmond, 2006.

[69] S. Gritzalis, E. Kavakli, C. Kalloniatis, P. Loucopoulos, S. Gritzalis, Incorporating privacy requirements into the system design process: the pris conceptual framework, Internet research 16 (2) (2006) 140–158.

[70] J.-H. Hoepman, Privacy design strategies, in: IFIP International Information Security Conference, Springer, 2014, pp. 446–459.

[71] S. Peltsverger, G. Zheng, Enhancing privacy education with a technical emphasis in it curriculum, Journal of Information Technology Education: Innovations in Practice 15 (1) (2015) 1–17.

[72] A. Senarath, N. A. Arachchilage, J. Slay, Designing privacy for you: A user centric approach for privacy, in: 19th International Conference on Human-Computer Interaction, Springer, 2017.

[73] N. S. Good, A. Krekelberg, Usability and privacy: a study of kazaa p2p file-sharing, in: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2003, pp. 137–144.

[74] F. Dabek, J. J. Caban, A grammar-based approach for modeling user interactions and generating suggestions during the data exploration process, IEEE transactions on visualization and computer graphics 23 (1) (2017) 41–50.

[75] A. Girgensohn, D. F. Redmiles, F. M. Shipman, Agent-based support for communication between developers and users in software design, in: Knowledge-Based Software Engineering Conference, 1994. Proceedings., Ninth, IEEE, 1994, pp. 22–29.

Appendix A. Questionnaire for Developers

Please generate a six digit random number before you proceed. Write the random number you generated below and click the option below according to the random number you generated.

option 1 : Random number generated is even. - Consider yourself as a software developer and attempt the design task below. Then proceed to the questionnaire

option 2 : Random number generated is odd. - Consider yourself as a user of the following application context and attempt the design task. Then proceed to the questionnaire

> Imagine that you are assigned to design a web-based health-care application that allows remote consultation with medical professionals, general practitioners and specialists, for a payment. Users should be able to browse through a registered list of medical professionals and chat (text/video) with them on their health problems for advice. Doctors and health-care professionals can register on the application to earn by providing their expertise to users. The application is to be freely available on-line (desktop/mobile). You may consider advertising and data sharing with third parties such as insurance providers and hospitals as potential avenues to increase the profits gained from the application.

Appendix A.1. questionnaire for role play developers

1. How would you describe your expectations if you were to use the application above?

2. Consider yourself as a user of the application context above. If you use the application of this nature, what level of privacy do you expect from the application on a scale of 1 to 10, with 1 being the lowest?

3. As a user what data would you willingly expose to this application as a user?

4. As a user what data do would you expose to the application even if you are unwilling as a user?

5. As a user what data do you expect the application to collect (irrespective of whether or not you are comfortable disclosing them)?

Conduct a high level stakeholder analysis, prepare a list of data collected, draw the information flow diagram and a high level data base sketch for the application described above. Your designs should consider privacy by following the principles listed below. Make use of the additional documentation provided at then end to read and understand the principles if you are not aware of them.

6. If you are asked to improve the application beyond the features described, what features would you like to add to it, given your expertise as a developer?

7. How possible is it to consider your own privacy expectations when you design a software application?

8. Please select the data items you decided to collect from a user from the list below if you were designing the application.

9. Please select the data items you consider to be sensitive from the list below.

10. Did you consider the following concepts in processing these sensitive data in your designs ? (Aggregation, Anonymization, Data.Expiry, Encryption, Pseudonimization, Separation)

Appendix A.2. questionnaire for non-role play developers

1. How would you describe a user's expectations in using the application above?

2. As a developer what level of privacy do you expect from the application on a scale of 1 to 10, with 1 being the lowest?

3. What sort of data do you think users would willingly expose to this application?

4. What sort of data do you think users would expose to the application even if they are unwilling?

5. What sort of data do you think users would expect the application to collect (irrespective of whether or not they are comfortable disclosing them)?

Conduct a high level stakeholder analysis, prepare a list of data collected, draw the information flow diagram and a high level data base sketch for the application described above. Your designs should consider privacy by following the principles listed below. Make use of the additional documentation provided at then end to read and understand the principles if you are not aware of them.

6. If you are asked to improve the application beyond the features described, what features would you like to add to it, given your expertise as a developer?

7. Please select the data items you decided to collect from a user from the list below.

8. Please select the data items you consider to be sensitive for users from the list below.

9. Did you consider the following concepts in processing these sensitive data in your designs ? (Aggregation, Anonymization, Data.Expiry, Encryption, Pseudonimization, Separation)

Appendix B. Questionnaire for users

Read and understand the application scenario given below. Consider you are to use the application and answer the questions below.

> Imagine that you are considering using a web-based health-care application that allows remote consultation with medical professionals, general practitioners and specialists, for a payment. In this application you will be able to browse through a registered list of medical professionals and chat (text/video) with them on their health problems for advice. Doctors and health-care professionals can register on the application to earn by providing their expertise to you. The application is freely available on-line (desktop/mobile).

1. How would you describe a your expectations if you were to use the application above?

2. What level of privacy do you expect from the application on a scale of 1 to 10, with 1 being the lowest?

3. Please select the data items you think the application would collect from a user from the list below.

4. Please select the data items you consider to be sensitive from the list below.

5. What sort of data do you think you would willingly expose to this application?

6. What sort of data do you think you would expose to the application even if you are unwilling?

7. What sort of data do you think you would expect the application to collect (irrespective of whether or not you are comfortable disclosing them)?

Appendix C. Participant Invitation Email

Dear X,

We are researchers in Computer Science & Engineering at the University X. We are looking for application developers to participate in a research study on Developers Attitudes towards Privacy by X. Looking at your contribution to the <X project> we would like to invite you to take part in our research study.

We appreciate the participation of experienced developers like you in research studies that focus on privacy and security aspects of software engineering to enable secure and privacy-aware software development through our research.

This study involves a simple application design task and a questionnaire on your experience in the industry environment as an application developer. You will be asked to design a software application (high-level stakeholder analysis, prepare a list of data collected, draw the information flow diagram and a high-level database sketch) by hand or any preferred software. The complete study is expected to take approximately 3 hours (2 hrs - design task, 40 mins - questionnaire). However, as the study is divided into two separate tasks, we believe it will be fun and interesting and would give you a break from your other activities.

If you're willing and available, please reply to this email! We will absolutely work around your schedule, and you can do the study remotely at any time that suits you. It is important that you don't feel coerced to take part in this study in any way. Just because we have emailed you does not mean that we, or anyone else, expect you to take part in the study, and no penalty will be incurred if you choose not to do so. If you decide to participate we would share our results with you. Further, as a gratitude for your participation, we will be giving you an Amazon gift voucher worth of 15 USD.

If you have any questions, please feel free to contact me replying to this email

Please feel free to forward this email or the advertisement attached herewith among your colleagues who are suitable for this study. That would be really helpful for us. Also, because we care about not spamming people, if you don't respond to this email (or one follow-up after you express your interest) or if you tell us you are not interested, we will not email you again or distribute your email anywhere.

Thanks and Regards X

Appendix D. Email for Participants with Study Information

Dear X,

Thank you very much for expressing your interest to participate in our research study on "Developers Attitudes towards Privacy" by X.

Attached herewith are the participation consent form (which gives us as the researchers to store and analyse your anonymized data), study information sheet (which explains who we are, what we do and the background of this study) and the study instruction form (which will guide you step by step through the study) as seperate pdf files. We have appendices attached in the study instruction form that would give you all the information you need on the technical aspects required in the tasks. First, read and understand the study information sheet and please fill in sign and scan/photograph of the participant consent form and send it to us. Then read the instructions form and continue the study as instructed.

As mentioned before it is expected to take about 2 1/2 -3 hours for the complete study. It has two parts (background questionnaire leading to the design task and the post task questionnaire) and you can take a break in between to manage your time accordingly. Please note that we do not expect detailed designs for the first part. Rough sketches are more than enough.

If you have any questions or suggestions for improvement please let me know. We welcome any feedback.

After completion, submit the questionnaire given at the end of the instruction form and please send me your design sketches together with the email address assigned to your amazon.com account, as we have a small token of appreciation (15USD). We know this is not much, and in no way equals your time and contribution. It is only a token of appreciation.

Hope you will enjoy the study!

Regards X