Security but not for security's sake: The impact of social considerations on app developers' choices

Irum Rauf
The Open University
irum.rauf@open.ac.uk

Dirk van der Linden
University of Bristol
dirk.vanderlinden@bristol.ac.uk

Mark Levine
Lancaster University
m.levine@lancaster.ac.uk

John Towse
Lancaster University
j.towse@lancaster.ac.uk

Bashar Nuseibeh
The Open University; Lero, University of Limerick
bashar.nuseibeh@open.ac.uk

Awais Rashid
University of Bristol
awais.rashid@bristol.ac.uk

ABSTRACT

We explore a dataset of app developer reasoning to better understand the reasons that may inadvertently promote or demote app developers' prioritization of security. We identify a number of reasons: caring vs. fear of users, the impact of norms, and notions of 'otherness' and 'self' in terms of belonging to groups. Based on our preliminary findings, we propose an interdisciplinary research agenda to explore the impact of *social identity* (a psychological theory) on developers' security rationales, and how this could be leveraged to guide developers towards making more secure choices.

CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**.

ACM Reference Format:
Irum Rauf, Dirk van der Linden, Mark Levine, John Towse, Bashar Nuseibeh, and Awais Rashid. 2020. Security but not for security's sake: The impact of social considerations on app developers' choices. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3387940.3392230

1 INTRODUCTION

Developing secure apps is a complex socio-technical activity [6]. From the urgency of getting apps out there, whether to be the first to capitalize on an innovative idea or simply to meet contractual deadlines, the complexities of monetization, and a not-entirely uncommon lack of structure in development processes, there are many things that app developers may be thinking about rather than security [7]. Previous research has shown that across different kinds of app development tasks, developers make secure decisions, but seemingly without having considered security explicitly in their reasoning [13]. This runs counter to the prediction from a social debt framework [10]—when the accrued consequences of decisions involving developer and development community eventually impact the software product. We perform an exploratory qualitative analysis of a dataset [14] covering a diverse range of app developers, focusing in particular on developers who claim to prioritize security, yet do not articulate security issues in explaining these choices. We do so to better understand *why* they make the decisions they do, and what aspects come into play.

ICSEW'20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7963-2/20/05…$15.00
https://doi.org/10.1145/3387940.3392230

2 DATA & METHOD

We analyzed a recent qualitative data set on 44 mobile software developers' rationales across different software development activities [14]. It provides app developers' prioritization choices (via card-sort or selection tasks) of different options impacting software security for six tasks and the rationales for the choices they make. Included are (1) setting up an IDE by choosing functionality; (2) fixing source-code by deciding what flaws to fix first; (3) deciding where to seek help using an API; (4) deciding whom to involve as testers; (5) what to consider when selecting an advertisement SDK; and (6) what clauses to favor for a software license agreement. The participants were not primed for security and hence were not aware that different options may differently impact software security.

Approach. The rationale analysis process was iterative. We study the subset (N=40) of app developers whose choices indicated a prioritization of security, with rationales that included clear security consideration and those that did not indicate any clear reflection about security. Moreover, we consider the different non-functional requirement (NFR) they prioritized in the data set. Only ten of the 40 prioritized security as a non-functional requirement (NFR). One researcher read the developer's rationales descriptively and coded the rationales. The codes were classified into a set of themes which captured the different identified aspects potentially affecting developers' reasoning. We identified eight new themes in the data.

Limitations. The coding scheme was discussed with another author in order to find disagreements and settle a final set of themes. We later agreed to exclude one theme–'security as a value'–because of the lack of common or convergent interpretation of the related rationales. This analysis has certain limitations, most importantly that it is a qualitative analysis of human reasoning. Our dataset is available at https://doi.org/10.17605/OSF.IO/3WHD5. It does not purport to present generalized claims, but serves as an identification of key concepts that are worthy of further in-depth study.

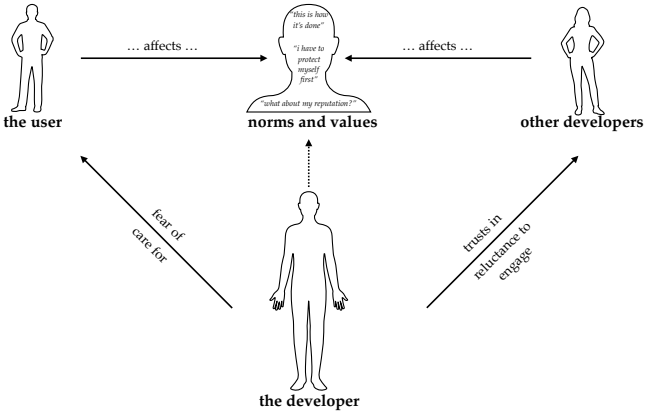
3 FINDINGS – FACTORS THAT MAY AFFECT SECURITY REASONING

Two key aspects underpin distinctions identified in the eight newly identified themes of developers' rationales:

(1) whether they had *a priori* prioritized security as an important non-functional requirement in an earlier task of the dataset;
(2) whether their reasoning addressed social aspects, comparing and contrasting users and other developers to themselves.

Across these aspects, *both of which present a different lens through which to analyze the rationales*, we identified eighth recurring themes in developers' rationales which explain how their reasoning may affect security rationale, summarized in Fig. 1.

Figure 1: The considerations we identified that a developer may have towards themselves, users, and other developers.



In what follows in Sec. 3.1– 3.2 we explore how the eight identified themes of developers' reasoning across these two aspects may give insight into their (lack) of reasoning about security.

3.1 Prioritizing security from the start (or not)

We noted some differences between those developers who had a priori prioritized security (10) and those who did not (30). These thematic differences are described below.

Th.1: Caring for users vs. Fearing them: Several developers reasoned about their users in order to decide what decisions to make. A difference that arose between the developers who prioritized security was that they seemed more driven by caring for their users. For example, one participant argued they "*wanted to prioritize things that could help the user*" (P5), or "*I would not want my users to experience any negative side effects from the ad library*" (P4), to even placing themselves in the 'shoes' of the user: "*I prioritized based on how 'extortionist' or 'evil' the clauses would feel to me as a consumer*" (P4). Yet, developers who did not prioritize security seemed more driven by negative feelings of losing their users or annoying them: "*Too many ads distract may annoy users*" (P11), or simply "*I don't want my ads to turn away my users*" (P16). Some reasoned about the impact decisions could have on their reputation: "*/…/ relying on your users will find you all the bugs, but might lose you your reputation*" (P20). These developers appeared to make choices driven by a desire to avoid getting into trouble: "*At first [I will] take care for myself and make sure my team is safe /…/*" (P21), and "*I can never be sure there aren't bugs in my code so I put anything that protects me in that case at the top*" (P17). Feelings of care or fear of negative user response may thus be an important aspect in what drives them to consider and even act securely, e.g., by considering secure behavior to be more socially responsible.

Th.2: Security as a norm: Literature has shown that developers often adopt security practice through peer influence [15], and subsequently consider them as norms or unwritten standards. Developers who did not a priori prioritize security considered their secure choices were simply the norm. For example, participants noted that security solutions "*look[ed] ok like this*" (P29), were a common practice ("*It's common practice, you start from the manual…*"(P3)) or were simply "*common sense*"(P28). This may indicate that developers are not motivated to make secure choices because they are security conscious per sé, but rather because certain patterns or ways of doing things have become ingrained—which aligns with efforts by the security community to encourage security practices in developers by making them common place.

Th.3: Being on your own or in a team: Being part of a team may provide more structure which in turn leads to more secure practices and reasoning—simply because there are systems in place such as code review. This, of course, does not necessarily push developers to consider security more explicitly, as they might simply consider it someone else's responsibility [5], but demonstrates that our participants at least orientate to the potential scrutiny of others. Looking at rationale of some of the insecure choices by developers who prioritize security a priori we see developers may shift from secure choices to insecure choices if they do not consider themselves part of the team any more: "*If it was a side project, chances are I would be publishing it as-is, to be as useful as possible immediately, offering it without any guarantees*" (P4) and "*Seems like a right approach for new personal project*" (P9). So, having a formal team structure which means that someone else will see their work clearly does impact on the way participants make decisions. However, developers do not need to be part of a team to think about themselves as a member of a group. Participants sometimes thought of themselves as member of broader social groups (e.g., being a developer, or an academic developer), even when they would be working on their own. Thinking about oneself as a member of such a group enhances one's trust in others, as some developers noted: "*[I would] try out the API and discuss with another peer with experience /…/ or ask the developer community*" (P9), or simply "*I would ask other developers I know*" (P27). This indicates that simply imagining themselves as a group member might raise similar concerns about how others in the group might evaluate their work.

Th.4: Trusting other developers: As alluded to above, trust in other developers is an important aspect of how developers reason. Some literature has shown that developers are more likely to trust other developers' opinion if they are of a similar socio-economic or educational status [15]. We saw that developers, whether they a priori prioritized security or not, often relied on opinions of other developers, with conscious awareness of whom to trust and why. However, they mostly reached out only after trying first themselves. This motivation to work independently seems influenced by the need to work efficiently as developers consider asking others to be more time consuming than trying to search for information themselves, e.g.,: "*I will ask people I know because it's easier to communicate efficiently with someone you already know*" (P22). Developers may thus trust other developers working on similar things: "*/…/ I have fellow devs who have used it*"' (P16)), or who are seen as an authority on the subject: "*I will first consult experts/…/*"(P31).

3.2 Thinking about 'others' and 'the self'

We noted further differences in developers' reasoning when they were explicitly thinking and classifying themselves or others.

Th.5: Thinking about 'others': Developers showed they considered different groups, most commonly reasoning about 'the user'. As noted before, this could be framed both with both positive and negative valence. For example, for some participants decision making was motivated by an intention to maximize positive user experience: "*I wanted to prioritize things that could help the user*" (P5). Some developers considered their users as customers rather than users—a subtle but important difference which reflects the tension between seeing others as a beneficiary or a source of revenue. For example, one participant noted that "*Firstly, I should take care of my business. Secondly, make customers happy*" (P24).

When it came to the development and testing process itself, some participants invoked a number of relevant others (including 'users')[1]. These included, friends, colleagues and family and sometimes even fellow developers. For example, one participant noted of their tester seeking strategies: "*I would ask a few colleagues and friends to test it first /. . . / If I was able to get a user group to test it (can be difficult to arrange) then I would, but this is rare*" (P6), while others similarly reasoned about the impact of using those close to themselves: "*I'd want it tested well, so a random user group is a good start. Another developer might be a better tester, but that's only one person. Friends and family are terrible testers, because they don't want to hurt your feelings*" (P20).

Developers thus seem to reference a variety of relevant 'others' when discussing security related decisions. However, participants were potentially primed to think of 'others' in the choices they were given for the testing task [14]. The way in which these groups are imagined (for example, whether a user is seen as a user, or more specifically as a customer; or whether another developer is seen as a friend or a colleague) will impact the way in which information is evaluated and priorities are set. Thinking about the relevance of any one of these groups at a particular time is likely to affect the decision making process.

Th.6: Thinking about 'the self': Another important aspect of how developers reasoned about their decisions is how they framed themselves. Sometimes it is clear they are thinking about themselves as individuals rather than as a member of a social group. This focus on themselves as individuals can be defensive. In other words, they can be motivated to act to protect their own self-interest: "*I can never be sure there aren't bugs in my code so I put [any license clause] that protects me in that case at the top*" (P16). Sometimes this focus on themselves as individuals can have a positive motivation. It can reflect a desire for autonomy, and respecting professional relationships with others: "*First I'll use resources I can use without the help from other people (their time is important as well)*" (P22).

In yet another case, developers may switch from talking about themselves in the first person to identifying themselves as a member of a social category. They move from a first person pronoun to a collective noun–or rather they qualify their self-description by reference to the social group: "***As a developer**, I sleep better at night if I have no knowledge of my user's passwords*" (P5). By prefacing an account with a claim to group membership ('as a developer') the participant is making a claim to motivation beyond individual self-interest. Their decision making is shaped by the norms and values of a social identity. Engaging in practices which violate these norms would lead to psychological or perhaps moral discomfort.

The same can be said for another developer, who made a slightly more refined claim to group membership as a specific group member: "*As a developer at a university that develops apps for its students to use, this commercial approach is new to me. I guess I would try to be as responsible as I could be with advertising, though*" (P6). Here again, the participant is describing the origin of the relevant norms and values ('to be as responsible as I could be') and how they are a consequence of their belonging to a group, "developer at a university" with obligation to a specific other (university students).

Finally, some developers considered themselves as a potential member of a user group. In other words, they can imagine themselves as developers and users at one and the same time. This is an important observation as it shows how participants can actively try to switch footing from one identity to another in order to try and evaluate the impact of decisions they might make: "*I prioritized based on how 'extortionist' or 'evil' the clauses would feel to me as a consumer*" (P4). This participant is emphatically imagining themselves as a consumer rather than a user. Thus, participants seem to think about themselves at different levels of inclusivity– sometimes as sovereign individuals and sometimes as members of social groups. The social groups themselves can be drawn in different ways (as a developer, or as a developer in a university– and even as a member of an imagined community of consumers). These different ways of thinking about the self can impact on security relevant decision-making in different ways at different times.

Th.7: Relying on 'others': When seeking help on a confusing API, a variety of social considerations became apparent, rather than straightforward technical or functional considerations. Some participants revealed their desire to work independently: "*I like to try and work things out fully first /. . . / Then ask a local expert*" (P6), while others noted efficiency: "*Searching for info in the web is in 99% of cases faster than asking another person*" (P11). Trust, efficiency and ease of understanding are key aspects here in understanding why developers reason like this over anything else. Since developers come from all type of backgrounds and development environments, their trust in different resources may varingly lead to (in)secure behavior. When developers are asked on how they seek testers for their app, they seemed influenced by the trust they have in social connections rather. For example, one participant noted that "*user groups will be the most honest, friends will be the most dedicated*" (P16), and others noted specifically the concept of friendliness: "*A friendly tester(s) loosely familiar with the concepts of the product will be likely to explore most of the options within the product*"(P33). This may indicate that developers prioritize social interaction by how comfortable they are engaging with, and placing trust in, others.

Th.8: Licensing and 'self'-defensiveness: When developers were asked to consider what clauses to include in a software license agreement, they showed little security consideration in their reasoning. Rather, it seemed developers focused on avoiding getting into trouble, such as one participant noting that "*liability is the only one i care about*" (P20). Indeed, as another participant noted, "*getting sued is worse than having someone copy your work as far as I'm concerned*" (P27). This hints at developers thinking of how others might affect them, and push them to be defensive, protecting their own interests and ability to 'be' an active developer, rather than consider what effect the license might have on the user.

[1]It should be noted that the task eliciting these rationales presented participants with a variety of sources, thus potentially priming them to think of social others.

4 TOWARDS A RESEARCH AGENDA

The aspects of developers' reasoning that we have discussed above all share a key thing: social considerations. We propose that research should investigate, in depth, to what extent social considerations may affect both the reasoning about security and eventual behavior towards security. As Fig. 1's summary of the relationships and considerations identified in our analysis shows, developers have conflicting relationships with both users and other developers, which manifest in both positive (e.g., caring for, trusting in) and negative (e.g., fearing, being reluctant to engage) ways. Moreover, an particularly important relationship is that of *the developer with themselves*: the self-reflection, mediated by norms and values their environment has instilled in them, driving them to reason in a particular way. This indicates that, depending on the environment a developer is situated in, there may be many different 'kinds' of typical developers, shaped by their interactions with users, other developers, and norms and values imposed on them by their wider environment. In order to understand how these different kinds of developers can eventually be stimulated to explicitly reason and act upon security as a key priority, acknowledging and dealing with that diversity is key. The comparing and contrasting of the self with their users and developers, and those people affecting the way developers reason, implies that developers' *social identity* may be a vital construct in order to better understand their reasoning.

The Social Identity Approach (SIA) [3, 4] is the product of four decades of work on the social psychology of the self and its relationship to groups and group processes. Pioneering work by Tajfel [8, 9] and Turner [11, 12] demonstrated that our sense of self is not fixed, but changes as a function of changes in our social context. People can define (and redefine) themselves along a continuum from a more idiosyncratic 'personal identity' to a more collective 'social identity'. At the same time, because multiple group memberships are available to us, our social identities can also dynamically update and adapt in their focus. As social contexts change, or as we begin to think about ourselves in relation to different individuals or groups, our sense of who we are also changes. This is important because our identities (personal and social) shape how we make sense of, and act, in the world. As different aspects of identity become more or less important to us, so do the values that we prioritize, and the degree that we can influence (and be influenced) by others changes so [1]. The Social Identity Approach (SIA) thus offers a psychological model that has the potential to explain variation in the way our participants orientate to decisions that have security implications. We have seen, in our data, examples of participants talking about themselves in both personal and social identity terms. We can also see them talking about different groups to which they can belong, and different groups with which they can interact. While we can see variation in the way they talk about and prioritize decisions that have security implications, we cannot yet make causal inferences between identities and security related decision-making—an important next step for future empirical work.

A first exploration into social identity in software development [2] noted that aspects of social identity affect software developers' behavior, and that this holds several implications for effectiveness of software development *that takes place in teams*. However, we also need to understand the wider demographic of app developers working solo, in not well defined teams, and without established organizational support—where the 'others' are likely even further removed. Thus, based on our analysis here, we propose several key **research questions** which need to be answered to better understand the effect of a wider range of developers' interaction with, and consideration of, others, on the security of apps they develop.

- Which, if any, social considerations mediate decisions made in software development?
- How do software developers identify with their users?
- How do software developers identify with 'other' developers?
- What are the effects of software developers identifying as part of a specific group on their attitude and behavior?

Conclusion— we identified a number of, primarily social, considerations that software developers exhibit in their reasoning about common software development activities, all of which may come in place of explicit security considerations. To understand *why* this happens, and *how* we may move these developers towards security, interdisciplinary research is needed. Social identity in particular provides an interesting lens to into the why and how of improving security choices of developers–and hence security of the resultant apps on which users increasingly rely in their daily lives.

Acknowledgments

This work is partially supported by EPSRC grant EP/P011799/1, Why Johnny doesn't write secure software? Secure software development by the masses and SFI grant 13/RC/2094.

REFERENCES

[1] Dominic Abrams, Margaret Wetherell, Sandra Cochrane, Michael A Hogg, and John C Turner. 1990. Knowing what to think by knowing who you are: Self-categorization and the nature of norm formation, conformity and group polarization. *British journal of social psychology* 29, 2 (1990), 97–119.
[2] Andreas Bäckevik, Erik Tholén, and Lucas Gren. 2019. Social identity in software development. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 107–114.
[3] Rupert Brown. 2020. The social identity approach: Appraising the Tajfellian legacy. *British Journal of Social Psychology* (2020).
[4] S Alexander Haslam. 2001. *Psychology in organizations*. London, Sage.
[5] Kai-Uwe Loser and Martin Degeling. 2014. Security and privacy as hygiene factors of developer behavior in small and agile teams. In *IFIP International Conference on Human Choice and Computers*. Springer, 255–265.
[6] Todd Sedano, Paul Ralph, and Cécile Péraire. 2017. Software development waste. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 130–140.
[7] Sofia Sherman and Irit Hadar. 2015. Toward defining the role of the software architect. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE, 71–76.
[8] Henri Tajfel. 1981. *Human groups and social categories: Studies in social psychology*. Cambridge: Cambridge University Press.
[9] Henri Ed Tajfel. 1978. *Differentiation between social groups: Studies in the social psychology of intergroup relations*. London:Academic Press.
[10] Damian A Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2013. What is social debt in software engineering?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 93–96.
[11] John C Turner, Michael A Hogg, Penelope J Oakes, Stephen D Reicher, and Margaret S Wetherell. 1987. *Rediscovering the social group: A self-categorization theory*. Oxford & New York: Basil Blackwell.
[12] John C Turner, Penelope J Oakes, S Alexander Haslam, and Craig McGarty. 1994. Self and collective: Cognition and social context. *Personality and social psychology bulletin* 20, 5 (1994), 454–463.
[13] Dirk van der Linden, Pauline Anthonysamy, Bashar Nuseibeh, Thein T. Tun, Marian Petre, Mark Levine, John Towse, and Awais Rashid. 2020. Schrödinger's Security: Opening the Box on App Developers' Security Rationale. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE)*.
[14] van der Linden, Dirk and others. 2020. Schrödinger's Security (ICSE 2020) Appendices. http://hdl.handle.net/1983/f43803de-4ade-488f-be1a-a2e8ba30c201. Online; accessed 8 January 2020.
[15] Shundan Xiao, Jim Witschey, and Emerson Murphy-Hill. 2014. Social influences on secure development tool adoption: why security tools spread. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 1095–1106.