# Challenges of Recruiting Developers in Multidisciplinary Studies

Irum Rauf, Tamara Lopez, Helen Sharp, Marian Petre
School of Computing & Communications, The Open University, Milton Keynes, UK
firstname.lastname@open.ac.uk

## Abstract

Security is a multivalent phenomenon so while it is technical, it is also social. Within research it is thus part of computing, but also other disciplines, including psychology. This paper reports on an empirical study conducted as part of multi-disciplinary research involving researchers from software engineering, social psychology and software security to understand how social identity influences the way developers engage with security in code. The goal of this report is to document the challenges faced in recruiting from different recruiting channels and identify how these challenges hindered our understanding of priming condition. We hope that this report will help other researchers in understanding these challenges of designing multi-disciplinary examinations and overcoming any pitfalls in future studies.

**ACM Reference Format:**
Irum Rauf, Tamara Lopez, Helen Sharp, Marian Petre. 2022. Challenges of Recruiting Developers in Multidisciplinary Studies. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Research in developer-centered-security requires an understanding of social and technical factors [7]. The understanding of non-technical aspects of developers' security lends itself to multi-disciplinary research. Secure development researchers often find the ecological validity of the empirical studies a challenge in terms of study design and recruiting the right participants [4]. This challenge becomes paramount when designing and implementing a study that meets the requirements of different research domains working to collectively investigate a shared area of research. This report documents our experience of recruiting developers for a multidisciplinary research study and highlights lessons we learned in the process. Our experience provides richer understanding into constraints to empirical research with the need to achieve ecological validity.

## 2 The Study

The study investigated how developers' salient identity influences their secure engagement with code using social identity theory [2]. We built on experimental work by Levine et al. [3] that assessed the helping behaviour of participants under conditions that primed them to think of themselves as individuals, or as part of a community mediated by feelings of social concern and responsibility. Following this work, in the study reported here, we hypothesized that a social identity would lead (on average) to decision making that is more security orientated, i.e., when identifying with a community, a developer would be more concerned to avoid 'damaging' other developers through poor security practice.

The study design was pre-registered [1] and approved by the authors' university's ethics committee. It was designed as a between-group online study in which participants are grouped into three conditions: personal and social identity conditions and a control condition. Participants of each group are prompted in the beginning to think of themselves either as individuals (personal priming), as part of developer community (social priming) or with no specific identity priming (control condition). The remainder of the study consisted of two code review [2] tasks and manipulation questions designed [5] to check for identity priming and its effect on participants' feelings of social concern. The study ended with security awareness and demographic questions. The final study was presented as a 20-to 30-minute online experiment.

We collected 124 valid responses through two online crowdsourcing communities for freelance (FL) developers - 82 from Upwork.com and 42 from freelancer.com. We paid £10/ hour (net amount) to each FL developer. We initially also considered hiring from Qualtrics research services (QRS). However, due to bottlenecks on expected changes and outcomes of the study, the agreement to obtain a concrete sample size of 75 from QRS was dissolved.

## 3 Challenges

The study satisfied the requirements of a multidisciplinary study outlined below and also achieved ecological validity. However, due to unanticipated challenges in the recruiting pathway the results of identity priming on developers' secure engagement with code were inconclusive. Our statistical analysis did not show any significant difference in developers' secure engagement with code under personal and social priming conditions. Below we highlight how the multidisciplinary nature of the study constrained the study design and led to unexpected challenges in the recruiting pathway, eventually affecting the study results.

### 3.1 Guarding the psychological priming:

*3.1.1 Req. 1- Psychological Priming of developers:* The study required that participants were not primed for their identity as part of a developer community in their recruiting pathway. To address this, we framed psychological priming questions by mentioning that the study is in collaboration with psychologists to study some human factors in coding and to help the psychologists know more about them as individuals. This was followed by the priming questions,


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://osf.io/4dk7t?view_only=7385245c30eb4a02ba579d373ce80fe7
[2] adapted from exercises at www.securecodewarrior.com

i.e., for personal identity or social identity priming. The psychologist in our group considered this sufficient to nullify any effect of priming during recruitment.

*3.1.2 Challenge 1 - Implicit Incentives:* We ensured that participants from all recruiting channels were paid £10/ hour (net amount) to nullify any uneven effect of incentives. However, we realized that participants were keenly aware of their reputation as developers in both online platforms and thus had an implicit incentive to achieve high feedback scores and completed job metrics within profile data.

*3.1.3 Challenge 2 - Screening Questions:* Screening questions are often used to filter out unintended audience in paid data gathering services. QRS suggested they could supply a pool of developer participants, but were not aware of the importance of preserving the priming conditions to the study design and required the addition of explicit questions to screen out non-developers. We denied the addition of direct screening questions about being a developer but agreed to add questions about their programming experience, i.e., if they had coded in last 12 months and if they code in Python. The addition of these questions, which was not initially envisaged during design time, threatened to affect the priming. In the event, QRS was only able to provide 8 valid responses and requested to change screening questions again. We found it difficult to change our screening questions again as we had planned to recruit participants simultaneously from two recruiting channels and already collected valid 62 responses from upwork.com. Hence, the project with QRS was mutually cancelled.

Existing studies also reported challenges in getting a required sample size from outsourcing research services. Danilova et al. [1] added small-programming pre-test to screen out non-developers but could only get less than half of the expected responses. We avoided pre-testing the programming knowledge of participants to avoid their psycholgoical priming as developers.

## 3.2 How did we address ecological validity?

*3.2.1 Req 1- Diversity of developers:* In order to achieve ecological validity, we were interested in recruiting developers from real-life settings, i.e., a sample that represents a diverse group of developers and working in their real working conditions. Additionally, the pre-registration of the study required a sample size of approx. 40 participants per identity manipulation to get acceptable power.

*3.2.2 Req 2 - Building a representative sample:* To minimise bias in our study, we required recruiting participants from more than one channel [8]. The hiring of professional developers was not scalable and required recruiting from more than one channel. We avoided more than two recruiting channels as it added noise to our recruiting pathway by adding different incentives, e.g., professional contacts may be incentivised differently than freelancers hired through freelancing websites. The psychologists in the group advised that this noise in recruiting pathway can effect the study results by influencing the behavior of participants differently.

*3.2.3 Challenge- Keeping Recruitment Channels to minimum:* In order to recruit developers from diverse backgrounds and gather a large sample size, we recruited developers via online pool of FL developers and paid research services (QRS) to expedite the hiring process. In the wake of poor-quality data from QRS, we avoided hiring full sample size from Upwork to triangulate our data sources. We also avoided advertising to our professional contacts and other social fora to avoid noise through incentivsation in recruiting pathway. We thus opted for another freelancing website to triangulate our data and to keep our recruiting channels to two at most. Freelancer.com had a high service fee and a hard to use interface (compared to Upwork) when hiring more than one developer for a job, so we decided to collect more data from Upwork and rest from freelancer.com. This kept distribution within two sources and provided a reasonable sample size to compare between the two groups in case of successful priming.

## 3.3 Conflicting study requirements and expectations of Research Services

*3.3.1 Req 1 - Code Review Tasks with open-ended questions:* We wanted to capture developer's behavior *in the moment* with evidence of security when they look at code. We ruled out the option of performing coding tasks, as code reviews more closely depicted a situation in which developers help each other - analogous to helping behavior reported in the Levine et al. [3] work in an emergency situation. We also required that participants share their own perspectives on security when examining code by including open-ended questions in the study.

*3.3.2 Req 2- Avoid too long and too short a study:* We aimed for 15-30 minute online study. A longer study would result in developers losing their focus as the study was online, non-interactive and unsupervised (no feedback was given on their answers) and a study too-short would not meet the psychological and security requirements set out for our study.

*3.3.3 Challenge- Rejecting recommendation for quantity over quality of responses:* To increase response rate, shorter studies are encouraged by paid recruiting services. QRS recommended a 9-minute survey to capture users' attention by adding option based questions instead of open-questions. QRS also requested to allow mobile users to do the study. These requests were rejected as they would have an effect on the quality of results.

# 4 Conclusion

Our experience of recruiting developers for multi-disciplinary study demonstrates how ecological validity can be preserved under various constraints. The report also shows that third-party recruiting services and aspects of the ecology itself can present particular challenges that can curtail inquiry and limit results. Though the results of our psychological experiment were inconclusive due to different challenges faced in recruiting, the open nature of our responses nonetheless provided qualitative insight into how developers engage with security in code [6]. We conclude that achieving ecological validity is not only about recruiting the right developers - it also requires careful study design efforts to support multi-disciplinary investigations and to ensure collection of "rich enough" data to permit multiple pathways for analysis.

# 5 Acknowledgement

We would like to thank all the study participants and acknowledge John Towse, Mark Levine, Joseph Hallett and Awais Rashid for their valuable input in the study design. The work was partially supported by UKRI/EPSRC (EP/P011799/1, EP/P011799/2, EP/R013144/1 and EP/T017465/1), NCSC, and SFI (13/RC/2094 and 16/RC/3918).

# References


[1] Anastasia Danilova, Alena Naiakshina, and Matthew Smith. 2020. One size does not fit all: a grounded theory and online survey study of developer preferences for security warning types. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 136–148.

[2] S Alexander Haslam. 2004. *Psychology in organizations*. Sage.

[3] Mark Levine, Amy Prosser, David Evans, and Stephen Reicher. 2005. Identity and emergency intervention: How social group membership and inclusiveness of group boundaries shape helping behavior. *Personality and social psychology bulletin* 31, 4 (2005), 443–453.

[4] Michelle L Mazurek and Daniel Votipka. 2020. Ecological validity and study design for empirical secure development studies. *Empirical Evaluation of Secure Development Processes* (2020), 12.

[5] Tom Postmes, S Alexander Haslam, and Lise Jans. 2013. A single-item measure of social identification: Reliability, validity, and utility. *British journal of social psychology* 52, 4 (2013), 597–617.

[6] Irum Rauf, Tamara Lopez, Helen Sharp, Marian Petre, , Mark Levine, , John Towse, Thein Tun, , Dirk Van der Linden, , , Awais Rashid, and Bashar Nuseibeh. [n. d.]. Influences of developers' perspectives on their engagement with security in code. In *Accepted at International Conference on Cooperative and Human Aspects of Software Engineering (CHASE '22)*.

[7] Irum Rauf, Marian Petre, Thein Tun, Tamara Lopez, Paul Lunn, Dirk Van der Linden, John Towse, Helen Sharp, Mark Levine, Awais Rashid, and Bashar Nuseibeh. 2021. The Case for Adaptive Security Interventions. *ACM Transactions on Software Engineering and Methodology (TOSEM)* (2021).

[8] Margaret-Anne Storey, Neil A Ernst, Courtney Williams, and Eirini Kalliamvakou. 2020. The who, what, how of software engineering research: a socio-technical framework. *Empirical Software Engineering* 25, 5 (2020), 4097–4129.