Strategies and Challenges in Recruiting Interview Participants for a Qualitative Evaluation

Carolin Brandt
Delft University of Technology
The Netherlands
c.e.brandt@tudelft.nl

Andy Zaidman
Delft University of Technology
The Netherlands
a.e.zaidman@tudelft.nl

ABSTRACT

This paper reports on our experience when recruiting participants for a round of semi-structured interviews to evaluate a new tool and process for developer-centric test amplification. We highlight strategies that helped us convince software developers to join and discuss challenges that we faced.

KEYWORDS

Participant Recruiting, Semi-Structured Interviews, Qualitative Study, Experience Report

ACM Reference Format:
Carolin Brandt and Andy Zaidman. 2022. Strategies and Challenges in Recruiting Interview Participants for a Qualitative Evaluation. In *Proceedings of 1st International Workshop on Recruiting Participants for Empirical Software Engineering (RoPES'22).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recruiting participants for a study can be a daunting task for novice researchers. Even more so for qualitative, interview-based studies, as the time investment of each participant is large and there will be a direct interaction between the researcher and the participant. In this experience report, we highlight strategies we employed and challenges we faced in the participant recruitment of our recent study of developer-centric test amplification [3]. Our goal is to inspire other researchers who are preparing a recruitment and contribute points for discussion at the RoPES workshop.

The aim of our study was to evaluate a new tool and interaction process that lets software developers better collaborate with an automatic test generation tool. We chose to conduct semi-structured interviews to gather qualitative insights on what developers think of the new process and collect open challenges to address in future research projects. Neither the earlier method of test amplification, nor our new developer-centric approach was known in the wider developer community. Therefore, we let the participants try out our approach in the first half of the interview and discussed their impressions and opinions in the second half. An interview was planned for 45 to 60 minutes.

To recruit participants, we published a text on our website[1] that explains the goal and process of the interview, along with how someone can sign up. We posted short advertisements for the study on Twitter[2] and LinkedIn, in which we linked to our website. In addition, we sent out personal emails to participants of an earlier survey related to the same topic and to our industry contacts in software engineering. Over two weeks, we recruited 16 participants.

In the following, we report on some choices that we think positively impacted our participant recruiting and discuss challenges that we faced.

2 OUR APPROACH TO MOTIVATE PARTICIPANTS

When recruiting for our study, we adopted several strategies to convince developers to join our study. We considered and communicated the value participants would get from participating, addressed the reader directly in our advertisement texts and streamlined the appointment selection.

2.1 Why Your Participants Should Participate

An aspect that we find crucial is to consider the value an individual will get from participating.

> Why is it worth for them to spend their time on our study? What do we give back to them?

We find it important to give something back to the participants and to communicate this in the advertisement.

In our case, the participants learned about a new tool and technique that could help them in their day-to-day test development. We explained that their feedback would be used to improve the tool further to fit their needs, making it a tool that could save them time in the future. We pointed out the chance for them to share their opinion and knowledge about software testing and their practices. Still, we consider the aspect of "helping someone in their research" to have an impact in convincing individuals to join. Especially when it comes to personal contacts, but also several of the other participants stated that they chose to participate to support ongoing research.

This focus on providing value to the participants was greatly influenced by Moritz Beller's previous work in our group [1]. His IDE plugin WatchDogs collected detailed statistics on how often developers use functionality related to testing and debugging. To convince developers to install and use his plugin, he included functionality that shares the collected data with the users, letting them gain insights into their own behavior as well.

RoPES'22, May 17, 2022, Virtual

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

[1] https://testshiftproject.github.io/test-amplification-interviews
[2] https://twitter.com/laci_noire/status/1328334375537299461?s=20

2.2 Why YOU Should Participate in My Study

When writing an advertisement text, we find it effective to address the reader directly.

> "Would *you* share your knowledge with us?"

A study invitation should be captivating and make the reader enthusiastic about participating. They should feel the fact that we are looking for *their* input. We see this aspect missing in other study invitations and drafts from our students. Possibly because of a fear of not being formal enough? During our interviews, we experienced how motivated the developers were to share their own approaches and opinions on software testing. We conjecture that putting emphasis on them sharing their opinions helped us recruit participants.

When it comes to sending out large numbers of emails, we feel that a little personalization goes a long way to convince someone to join. We greeted everyone by their name and added a personal entry sentence where possible. Also we translated the email to the language that we normally speak with the recipient. In total we sent out 18 emails with a custom name and 5 fully individual ones. We recruited 7 participants through the online postings or our previous survey, and 9 participants through asking our contacts directly.

2.3 Avoiding Scheduling Mails

The sign-up process and our technical setup for the interviews was focused around minimal extra effort and time investment for our participants. To let them immediately see when they could participate, we used an online calendaring service. They could pick a date and time directly after reading our advertisement. We used Picktime[3], other options are Calendly or Doodle.

2.4 No Technical Setup

To avoid complications of locally setting up our tool, we provided a browser version of IntelliJ to use during the interview[4]. In a later study, we used a second computer and Zoom's "take over control" feature [2]. We communicated up front that there was no technical setup necessary from the participant's side. We hypothesize that being able to immediately act on our advertisement and the expectation of a clear, time-boxed participation helped us to convince software developers to join.

3 CHALLENGES

The participants we recruited shared great insights, which made our study a success and majorly shaped out ongoing research [4, 5]. Still, we faced several challenges with respect to participant recruiting.

3.1 Unfamiliar Example Project

To enable anyone with a basic knowledge of Java and JUnit join, we used a small, easy to explain example project during our study. Therefore, we could not observe how developers would use our tool on a project that they are deeply familiar with. Designing a study for this would require picking a project with sufficient developers working on the project and agreeing to participate in the interviews beforehand.

3.2 Selection Bias of Convenience Sampling

Among others, we recruited previous industry and university colleagues as participants, who learned and developed software under similar circumstances as the main researcher. This opens up the threat that the same views, about, e.g., software testing, are reenforced during the study. However, such selection criteria might also be valid. Our goal was to evaluate a tool that helps developers improve their test suite. Individuals that are not interested in improving a test suite would not be in the targeted user audience. Therefore, we conjecture it to be valid to not explicitly recruit them for our study.

3.3 Clear Stopping Criteria for Recruitment

A feedback we received during the reviews was the lack of a clear stopping criterion during the recruitment. Because of the linear time frame of the study, we stopped reaching out to more participants when we felt we have "enough" registrations for the time we planned to spend on the interviews. For future studies, we would like to find a clearer stopping criterion that we can communicate in the final publication.

4 CONCLUSION

All in all, participant recruiting was a solvable challenge for our qualitative evaluation. Through advertising the value of participating, addressing texts directly to the potential participants and streamlining the sign-up and setup we could recruit a sufficient number of participants. Nevertheless, we faced challenges such as needing to introduce an unfamiliar example project, the threat of selection bias in convenience sampling and the lack of a clear stopping criterion.

We look forward to the discussions at the RoPES workshop, especially about recruitment platforms such as Prolific and other strategies to reach more and diverse participants while avoiding overly-researched groups.

ACKNOWLEDGMENTS

This work was sponsored by the Dutch science foundation NWO through the Vici "TestShift" project (No. VI.C.182.032).

REFERENCES

[1] Moritz Beller, Georgios Gousios, Annibale Panichella, Sebastian Proksch, Sven Amann, and Andy Zaidman. 2019. Developer Testing in the IDE: Patterns, Beliefs, and Behavior. *IEEE Trans. Software Eng.* 45, 3 (2019), 261–284.

[2] Casper Boone. 2021. *TestAxis: Save Time Fixing Broken CI Builds Without Leaving Your IDE.* Master's thesis. Delft University of Technology. http://resolver.tudelft.nl/uuid:f8375d5f-3bbd-4559-863b-6951e9d6bab0.

[3] Carolin Brandt and Andy Zaidman. 2022. Developer-Centric Test Amplification The Interplay Between Automatic Generation and Human Exploration. *Empir. Softw. Eng.* (2022). To Appear.

[4] Nienke Nijkamp, Carolin Brandt, and Andy Zaidman. 2021. Naming Amplified Tests Based on Improved Coverage. In *2021 IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM).*

[5] Wessel Oosterbroek, Carolin Brandt, and Andy Zaidman. 2021. Removing Redundant Statements in Amplified Test Cases. In *2021 IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM).*

[3] https://www.picktime.com/

[4] https://lp.jetbrains.com/projector/