How (Not) to Recruit Students Outside of Computer Science: An Experience Report

Elisa Hartmann
Chemnitz University of Technology
Chemnitz, Germany
elisa.hartmann@informatik.tu-chemnitz.de

Janet Siegmund
Chemnitz University of Technology
Chemnitz, Germany
janet.siegmund@informatik.tu-chemnitz.de

***Abstract*—To get a detailed understanding of program comprehension and programming, which are on the verge of becoming the 4th literacy, it is crucial to study students who are real novice programmers, since they did not have any contact with programming and have not developed any mental model. Studying how they build such a mental model can shed light on important cognitive steps that they take. To this end, we set out to recruit students from non-computer science departments. In this paper, we recount the difficulties of recruiting participants from a department that is not related to computer science.**

I. Introduction

In our line of research on program comprehension and programming language acquisition, we conduct studies with human participants. Especially for programming language acquisition studies, we need students with different levels of programming experience. For example, some studies evaluate how novice programmers acquire different cognitive levels during programming, and they start with programmers who had no prior contact with programming [1]. Typically, the first-year students of our university have already been programming for one or two years. For our studies, in which we take a look at the cognitive processes that are part of program comprehension by using neuroimaging techniques, it would be helpful to observe how complete novices react to source code: Since they have not seen source code before, they have no specified part(s) of their brain to react to this, compared to undergraduate students who have been receiving some sort of programming training [2].

Of course, it would also be interesting to observe how experienced programmers with many years of experience react to source code. Maybe they have specialized programming so well that there is a unique area in the brain that reacts to source code? Unfortunately, recruiting professional programmers is also rather difficult, as they typically cannot be offered the compensations that students receive (e.g., bonus points for assignments or a voucher), but require financial compensation. For example, Dag Sjøberg describes that for one big case study, he hired four different companies, which made the case study cost $ 250,000 [3].

Since we are part of the Computer Science Department, we have relatively easy access to students of computer science who are enrolled in our courses. To this end, we often design experiments as part of mandatory assignments that fit the learning goals of a course. However, when we want to ensure that students have no programming experience, we typically have to look beyond our department, since computer science students often have some experience with programming before they enroll in our university and may take programming-related courses during their course of study.

Unfortunately, recruiting from outside our department is considerably more difficult, because we cannot simply approach them in courses, and because the content of an experiment is often less relevant for them, as it is farther away from their course of study. Of course, offering financial compensation of non-computer science students for participation would technically be an option, but we usually do not have the financial resources to do so (let alone for programming experts).

II. (Failed) Idea

Students of psychology departments are often required to participate in different experiments as part of their study regulations. For their participation in these experiments, students receive credit points to complete their course of study.

For this reason, we approached two members of the Department Of Psychology to see whether we could also award credit points to students of Psychology. These members agreed with our assessment that psychology students would be good participants for our study, and also that offering a new perspective regarding how research questions in computer science are approached is a valuable experience for psychology students. Unfortunately, participants in the psychology department are also a scarce and valuable resource. Thus, many of the experimental hours are scheduled internally, so we could not award credit points to psychology students.

III. Solution

In the end, there were three possible solutions to still recruit students with no programming experience for our studies. First, we could collaborate with research groups from the other departments who can award credit points for participation to students. However, for our already fleshed out studies, this would be difficult to implement, as we would need to make sure that every researcher involved agrees with a study design (e.g., the definition of research questions and

This work is supported by the DFG (SI 2045/2-2)

the operationalization of variables). While having diverse backgrounds is helpful to create valid studies, it still requires detailed communication and cooperation to satisfy each of these different backgrounds. Especially with our studies ready to go, this is not feasible. Additionally, publishing the results of studies would be difficult, as the fields for Psychology and Computer Science are rather disjointed. How should we decide to which doctoral student we would assign a study to? Even though the research community is beginning to move away from the importance of papers, doctoral students still need well-published papers to finish their PhD. Thus, this solution is only applicable in a few cases.

Second, if we would draw resources from the participant pool, it would be fair if we could also contribute to it. In other words, we could require computer science students to also take part in empirical studies to complete their course of study. Since the human factor plays an important role in computer science and empirical research has become standard, it would make sense for students from the computer science department to also get in touch with such studies. If this pool of participants would be accessible for both departments, then a bigger, more diverse sample would be possible for research groups of different scientific domains. Unfortunately, this is only a long-term solution, as changing the study regulations requires tremendous bureaucratic effort. As a result, we intend to follow this approach from a long-term perspective, but we need other short-term options.

Third, we could motivate students from psychology in another way to take part in our study. Specifically, during our talks with the staff from the psychology courses, we learned that students must, at some point, conduct data analysis on the level of programming languages, such as R or Python, and not solely based on dedicated programs with a user interface that hides programming details, such as SPSS.

We decided to offer a voluntary (i.e., without credit points) programming course that is tailored to data analysis as students learn in their course of study. Not being part of the general study program, it is no bureaucratic effort to offer this course and a viable short-term solution. To ensure that the course attracts students of psychology, we conducted a survey among them. Based on their responses, the course will cover topics in data preprocessing (including outlier removal), descriptive statistics (including visualizations), and inferential statistics (e.g., conducting significance tests). As programming language, we will use Python. The course is planned to last about one week during the semester break (March 2022), so students have no other obligations during their course of this study.

Within this course, we have integrated our experiment. Specifically, we explore several teaching approaches and compare the success of these approaches by giving students programming assignments and collect their responses in terms of correctness and response times. The assignments are similar to what we use in our computer science programming course, so that we can compare the results across different departments. After this course, we plan to conduct interviews about the learning experience.

A major drawback of this plan is that mainly students who are really interested in learning how to do statistical analysis on the programming language level might take part. More so, the statistics course is typically unpopular with psychology students, so students who take a non-credit programming course might be more inclined to statistical analysis. Therefore, our sample might possibly be biased.

We are eager to see how our programming course is received. With a positive response, we will offer it in a modified form for other fields of study from the humanities to diversify our sample.

Unfortunately, we have no way of knowing how well this approach works in practice. Although this requires more effort compared to design experiments only, as we have to integrate our experiment within that course and at least partly align it with programming courses at the computer science department, we still believe that this is worth the effort. Not only can we approach students from other domains and make our sample more diverse in terms of academic background, we can also make their invested time worthwhile beyond participation in our experiment. Especially since programming is on the verge of becoming the 4th literacy, we hope that we can also lower the barrier to entry for non-computer science students to acquire basic programming skills.

ACKNOWLEDGMENT

We thank the staff from the psychology department for supporting us in exploring the option to award credit points to psychology students.

REFERENCES

[1] A. Ahadi, R. Lister, and D. Teague, "Falling Behind Early and Staying Behind When Learning to Program," in *Psychology of Programming Interest Group Annual Conference (PPIG)*, 2014, pp. 77–88.

[2] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann, "Understanding Understanding Source Code with Functional Magnetic Resonance Imaging," in *Proceedings of the 36th International Conference on Software Engineering*. Association for Computing Machinery, 2014, pp. 378–389.

[3] W. Tichy, "Empirical Software Research: An Interview with Dag Sjøberg, University of Oslo, Norway," *Ubiquity*, vol. 2013, no. December, pp. 1–14, 2013.