

How Do Students Envision Good Programmers? Investigating CS1 Students' Perceptions of Professional Programmers

Yaurie Hwang
yaurie@u.northwestern.edu
Northwestern University
Evanston, Illinois, USA

Eleanor O'Rourke
eorourke@northwestern.edu
Northwestern University
Evanston, Illinois, USA

Elise Lee
eliselee@u.northwestern.edu
Northwestern University
Evanston, Illinois, USA

Jamie Gorson
jgorson@u.northwestern.edu
Northwestern University
Evanston, Illinois, USA

ABSTRACT

We know very little about how CS1 students perceive professional programmers. Studies show that CS1 students sometimes negatively self-assess their ability to program at moments that are natural parts of professional programming. By understanding how CS1 students perceive professionals, we may be able to prevent students from negatively self-assessing at those moments. We used vignette-style questions to describe programming moments that professionals encounter. Some students believed that professionals rarely make simple errors, while other misconceptions were more nuanced, such as that professionals use resources, but are discouraged from doing so. Our findings also suggest that students' sources of information about professional programmers vary greatly, ranging from deductive reasoning to the Internet. By reducing CS1 students' misconceptions about professional programmers, we may be able to improve self-assessments for CS1 students.

ACM Reference Format:

Yaurie Hwang, Elise Lee, Eleanor O'Rourke, and Jamie Gorson. 2023. How Do Students Envision Good Programmers? Investigating CS1 Students' Perceptions of Professional Programmers. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (SIGCSE '23)*. ACM, New York, NY, USA, 3 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 PROBLEM AND MOTIVATION

Professional programmers encounter many different moments when programming. For example, professionals frequently plan [9, 11], get errors [1, 8], and ask for help [6]. Surprisingly, some CS1 students see these moments as indications of low programming ability [3]. Gorson and O'Rourke found that some students consider planning as indicative of lower programming intelligence [3], even though planning is widely documented as part of the professional process [9, 11]. Thus, we hypothesize that students may have inaccurate perceptions of the professional programming process.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE '23, March 15–18, 2023, Toronto, Canada

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

These perceptions can have a large impact on CS1 students. CS1 students sometimes negatively self-assess their programming ability when they encounter natural parts of programming [3]. Negative self-assessments may contribute to the high dropout rates in CS [7, 10].

By understanding how CS1 students perceive the professional programming process, we may be able to prevent students from negatively self-assessing from natural moments in programming. However, we are not aware of any prior studies on how CS1 students perceive the professional programming process. In this paper, we investigate these perceptions and from where they might stem.

2 BACKGROUND AND RELATED WORK

Novice programmers make frequent assessments of their programming ability [5]. At times, they make negative self-assessments, even after a successful programming session [5]. Self assessments can have lasting impacts on CS1 students, such as their belief in their ability to program [2] and decision to major in CS [7].

Gorson and O'Rourke hypothesize that students make negative self-assessments at moments that are inconsistent with how they believe professional programmers behave [4]. For example, if students negatively self-assess when they get simple errors, they might believe that professionals rarely make syntax errors. In this paper, we take a closer look at how students perceive professionals, then identify their misconceptions about professional programmers. Knowing how CS1 students perceive the programming process can help us understand how to reduce their negative self-assessments and improve their CS learning experience.

3 METHODOLOGY

We conducted semi-structured interviews with 17 CS1 students to learn more about their perceptions of professional programmers. The goal of our interviews was to better understand their expectations of professional programmers, knowledge of the professional CS field, and personal experiences in CS.

3.1 Participants

The researcher conducted two pilot interviews with two undergraduate students from a mid-sized private university in the midwestern USA. These interviews allowed the researcher to iterate on the interview questions. Next, the researcher recruited 15 additional students, 7 from a mid-sized US private university and 8 from a

large US public university for additional interviews. All participants were recruited via emails sent by the professors of CS1 courses. All participants were either currently enrolled in a CS1 course or had taken the course earlier in the same academic school year. All participants were considering a CS major or had declared one.

3.2 Interview Protocol

The interviews were conducted virtually. First, the researcher asked the participant general questions about professional programmers, such as what they do at their jobs, their salient characteristics, and the challenges of becoming one. Next, the researcher presented fictional scenarios, known as vignettes, with each describing a moment that a professional programmer might experience. Vignettes were based on the negative self-assessment moments [4]. For each vignette, the student stated the likelihood of the scenario and their reasoning why. After the vignettes, the participant cited their sources of general knowledge about professional programmers.

3.3 Analysis

To conduct the analysis, the researcher created 16 categories based on the questions asked. For each category, the researcher conducted three steps. First, the researcher collected a transcript of each interview about the category. Next, the researcher memoed about the answers to interpret them. In each memo, the researcher included a summary of the answer and the source of information. Finally, the researcher conducted a thematic analysis from the memos, summarizing the themes and important details.

4 RESULTS AND CONTRIBUTIONS

In this section, we identify two misconceptions about professional programmers. Then, we discuss sources for information about professionals. Lastly, we explore the significance of the findings.

4.1 Misconceptions About Use of Resources

A majority of participants believed that professional programmers use resources, but negative perceptions about using resources persisted. Participants answered the question "How likely would a professional would use resources frequently?". Most responses were *strongly likely* (9 participants) and *slightly likely* (5 participants). One outlier said *unlikely*.

Many participants believed that professionals should use resources under special circumstances. For instance, although Participant 4 answered that professionals are *strongly likely* to use resources, they added, "You can't directly copy and paste [code], but you can use it to understand how you would solve your own problems." Similarly, Participant 12 said, "You should be able to at least understand the problem and how to implement the solution without Google... if you use it as a crutch then your coworkers will realize that at some point." Participant 12 believed that professionals are *strongly likely* to use resources, but they should not use the resources to understand or approach the problem. They implied that this practice is discouraged in industry, saying that using resources could result in being "found out" by coworkers.

Overall, we found that a majority of participants believed that programmers used resources, but still had misconceptions about using resources.

4.2 Misconceptions About Simple Errors

Some students believed that professionals are unlikely to make simple errors. We asked participants: "How often do you think professionals get simple errors, like small typos?". Out of 15 participants, 4 answered *slightly unlikely* or *strongly unlikely*.

Some believed that professionals would have enough experience to rarely make small errors. For example, Participant 14 said: "As I've taken more classes, my frequency for making typos, like missing a colon, has decreased... A professional programmer's more likely to remember that [syntax] because they're experienced." Participant 14 recalled making less simple errors as they continued to learn how to program. By the same token, they expected professionals to be *strongly unlikely* to make simple errors.

Some participants believed that professionals make simple errors, but not in typical scenarios. Participant 12 said professionals make mistakes when they are tired or multitasking: "If it's really early in the morning, then I'm a little careless in some things. I do something silly. Then, it's pretty easy to do something small, or when you're trying to work on a lot of things at one time."

Participants thought that professionals are unlikely to make simple errors for a variety of reasons, such as experience level and circumstance. Although roughly 25% of participants believed professionals do not make simple errors often, in the context of all CS1 students, a quarter of them is a notable amount.

4.3 Sources of Perceptions

Deductive reasoning was a common source for the participants' perceptions. For example, Participant 15 believed that professionals are *strongly likely* to be fast at typing because they had experienced being frustrated by their own slow typing. From this experience, they hypothesized that "As a professional, I think it'd be even more annoying if you were going really slow."

When asked about their general knowledge of professional programmers, participants most frequently cited online sources and conversations with other people. 11 out of 15 participants said online forums (e.g. Quora), articles, videos, or social media. A different set of 11 participants mentioned other people, such as friends, family, and current professional programmers. Some participants also cited personal experience working in industry and movies.

4.4 Significance

Our analysis provided insight on misconceptions that CS1 students have about professional programmers. Some students believed that professionals rarely make simple errors, while other misconceptions were more nuanced, such as that professionals use resources, but are discouraged from doing so. We also found that CS1 students use a variety of sources for their perceptions, such as personal experience and deductive reasoning.

This research has a number of implications and opens up areas for future work, including studies on students' beliefs on their ability to program. We suggest CS1 educators to integrate information about the professional programming process into their curriculum. Giving students feedback about their programming process may also be beneficial. From these additions, CS1 students may learn that the moments where they negatively self-assess are not indicators of low programming ability.

REFERENCES

- [1] Ryan Chmiel and Michael C. Loui. 2004. Debugging: from novice to expert. *ACM SIGCSE Bulletin* 36, 1 (2004), 17–21. <http://dl.acm.org/citation.cfm?id=971310>
- [2] Jamie Gorson, Nicholas LaGrassa, Cindy Hsinyu Hu, Elise Lee, Ava Marie Robinson, and Eleanor O'Rourke. 2021. An Approach for Detecting Student Perceptions of the Programming Experience from Interaction Log Data. In *Artificial Intelligence in Education*, Vol. 12748. Springer International Publishing, Cham, 150–164. https://doi.org/10.1007/978-3-030-78292-4_13 Series Title: Lecture Notes in Computer Science.
- [3] Jamie Gorson and Eleanor O'Rourke. 2019. How Do Students Talk About Intelligence? An Investigation of Motivation, Self-efficacy, and Mindsets in Computer Science. In *Proceedings of the 2019 ACM Conference on International Computing Education Research - ICER '19*. ACM Press, Toronto ON, Canada, 21–29. <https://doi.org/10.1145/3291279.3339413>
- [4] Jamie Gorson and Eleanor O'Rourke. 2020. Why do CS1 Students Think They're Bad at Programming? Investigating Self-efficacy and Self-assessments at Three Universities. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*. 170–181.
- [5] Päivi Kinnunen and Beth Simon. 2012. My program is ok – am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education* 22, 1 (March 2012), 1–28. <https://doi.org/10.1080/08993408.2012.655091>
- [6] Thomas D. LaToza, Gina Venolia, and Robert DeLine. 2006. Maintaining mental models: a study of developer work habits. In *Proceeding of the 28th international conference on Software engineering - ICSE '06*. ACM Press, Shanghai, China, 492. <https://doi.org/10.1145/1134285.1134355>
- [7] Colleen M. Lewis, Ken Yasuhara, and Ruth E. Anderson. 2011. Deciding to major in computer science: a grounded theory of students' self-assessment of ability. In *Proceedings of the seventh international workshop on Computing education research*. ACM, 3–10.
- [8] Michael Perscheid, Benjamin Siegmund, Marcel Taeumel, and Robert Hirschfeld. 2017. Studying the advancement in debugging practice of professional software developers. *Software Quality Journal* 25, 1 (March 2017), 83–110. <https://doi.org/10.1007/s11219-015-9294-2>
- [9] Sabine Sonnentag. 1998. Expertise in professional software design: A process study. *Journal of applied psychology* 83, 5 (1998), 703. Publisher: American Psychological Association.
- [10] F. Boray Tek, Kristin S. Benli, and Ezgi Deveci. 2018. Implicit Theories and Self-Efficacy in an Introductory Programming Course. *IEEE Transactions on Education* 61, 3 (Aug. 2018), 218–225. <https://doi.org/10.1109/TE.2017.2789183>
- [11] Willemien Visser. 1987. Strategies in programming programmable controllers: A field study on a professional programmer. In *Empirical Studies of Programmers: Second workshop (ESP2)*. Ablex, 217–230.