



Student Expectations of Tutors in Computing Courses

Rachel S. Lim

University of California San Diego
ral077@ucsd.edu

Ismael Villegas Molina

University of California San Diego
isvilleg@ucsd.edu

Sophia Krause-Levy

University of California San Diego
skrausel@eng.ucsd.edu

Leo Porter

University of California San Diego
leporter@eng.ucsd.edu

ABSTRACT

Many institutions use undergraduate teaching assistants (tutors) in their computing courses to help provide more resources to students. Because of the role tutors play in students' learning experiences, recent work in computing education has begun to explore student-tutor interactions through the tutor's perspective and through direct observation of the interactions. The results suggest that these interactions are cognitively challenging for tutors and may not be as beneficial for students' learning as one might hope. Given that many of these interactions may be unproductive, this work seeks to understand how student expectations of these sessions might be impacting the interactions' effectiveness. We interviewed 15 students in a CS2 course to learn about the expectations and desires that students have when they attend tutoring sessions. Our findings indicate that there is variation in what students consider a desired result from the interaction, that assignment deadlines affect students' expectations and desires for interactions, and that students do not always want what they believe is beneficial for their learning. We discuss implications for instructors and potential guidance for students and tutors to make tutoring sessions more effective.

CCS CONCEPTS

- Social and professional topics → Computing Education.

KEYWORDS

tutoring, student perspectives, student outcomes

ACM Reference Format:

Rachel S. Lim, Sophia Krause-Levy, Ismael Villegas Molina, and Leo Porter. 2023. Student Expectations of Tutors in Computing Courses. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*, March 15–18, 2023, Toronto, ON, Canada. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3545945.3569766>

1 INTRODUCTION

Research in the field of education has shown that when instruction is given by a highly-capable tutor, there is a positive impact on student learning [4]. Specifically in computing, undergraduate tutors have been found to play an important role in students' learning and

overall course experience [3, 7, 11, 20]. As class sizes grow, tutors can help with the scalability of courses by providing more support to students when an instructor's time is limited [18]. Additionally, tutors can play an important role in providing feedback to the instructors that the instructors can then use to improve the course and the overall experience for students [8].

Findings from recent studies indicate that an ideal student-tutor interaction may be difficult to achieve. Krause-Levy et al. observed many undesirable behaviors and teaching practices in video recordings of student-tutor help sessions. They found that tutors engaged in behaviors such as not asking the student to explain their problem and explicitly giving the student the solution to their problem [15]. Work by Markel and Guo and Riese et al. sheds light on the challenges that tutors face during these tutoring sessions, including time pressure, emotional regulation, and reading the student's mind [17, 22]. However, not much is known about the student's perspective and how it may play a role in the effectiveness of the interaction.

To gain a deeper understanding of what might be causing these interactions to be ineffective, our work examines students' expectations and desires when they attend tutoring sessions. We conducted semi-structured interviews with 15 students enrolled in a Basic Data Structures (CS2) course and performed a qualitative analysis on the data using phenomenography. We identified a broad range of student expectations and desires regarding tutoring sessions.

Our results show that despite students often acknowledging what is good for their learning and saying that they want to learn, their expectations and desires for tutoring sessions are often inconsistent with their learning goals. For example, students unknowingly overlook the importance of finding their own bugs, and as a result, they do not see any issues with the tutor identifying their problems for them and at times even expect it. Additionally, students report that when a deadline is approaching, they prefer for the tutor to just give them the solution to help them make the deadline, clearly prioritizing their grade on a particular programming assignment over their long-term learning. There seems to be a disconnect between students' expectations and desires for tutoring sessions and their long-term desire to learn, and we believe that it exists due to students both being unaware of this disconnect and prioritizing short-term goals over their learning.

2 PREVIOUS WORK

A recent literature review of prior work on undergraduate tutors in computing summarizes some of the benefits of undergraduate tutors [18]. The reported benefits to students in computing include individualized assistance [10, 21], increased participation [5, 9, 27],



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9431-4/23/03.

<https://doi.org/10.1145/3545945.3569766>

improved attitude and motivation [9], and a significant positive impact on students' learning [3, 11, 20].

A recent study by Krause-Levy et al. examined the contents of student-tutor interactions through video recordings of the sessions. Interestingly, these findings suggest that student-tutor interactions may not be as beneficial as described above. Some of the key findings were that tutors did not help the student debug in most of the interactions, asked guiding questions in only about one-third of the interactions, and explicitly gave the student the solution in over half of the interactions [15].

Other recent work has identified that tutors face many challenges during these interactions that may be leading to the issues seen in the work by Krause-Levy et al. [15]. Markel and Guo began to explore these challenges through the experience of a single tutor and identified some of the challenges that tutors face during tutoring sessions. These challenges include teaching vs. bug-fixing, triage and prioritization, and maintaining student engagement [17]. Another study by Riese et al. focused on understanding the challenges faced by instructional assistants (including undergraduates) through the use of reflection essays collected from the tutors in their training courses [22]. The work highlights how tutors have various challenges including, but not limited to, student-focused challenges (e.g., individual content knowledge, unprepared students), defining and using best practice (e.g., identifying good practices, mapping student knowledge to properly teach), and threats to best practice (e.g., time constraints, tutor knowledge insecurity) [22].

Our study builds upon the work of Krause-Levy et al., Markel and Guo, and Riese et al. by examining the perspectives of the students themselves and what they desire from these interactions. By gaining a better understanding of student expectations and desires, this work provides insight into why some interactions with tutors may be ineffective and may inform potential remediation efforts.

3 STUDY DESIGN

3.1 Research Question

To understand the range of student perspectives on tutoring sessions, we asked the following research question: What are students' expectations and desires when they attend tutoring sessions?

3.2 Course Context

We interviewed students who were enrolled in a Basic Data Structures (CS2) course at the same large research-intensive public university in North America as Krause-Levy et al. [15]. Some of the key topics covered in this 10-week course are stacks, queues, heaps, binary search trees, and time complexity. Computer science students typically take this course as the second or third course during their first year as an undergraduate.

During the course, students were given the opportunity to attend tutoring sessions to receive individual help from undergraduate tutors. Help was available on all days of the week. At these designated times, a student could submit a ticket to an online queuing system to request individual help from a tutor. Students usually attend these sessions to ask for help with debugging their programming

assignments. These sessions were available both in person in the computer labs and remotely via Zoom.

The tutors were all undergraduate students who had previously taken and earned a passing grade in the course. At our institution, tutors are selected by the course instructors. As part of a lengthy application, instructors see information such as an applicant's overall GPA, grades received in individual courses, and courses for which they have previously tutored.

3.3 Data Collection

The study was announced near the end of the term to the students enrolled in CS2 during Spring 2022 by the instructors of the course. The recruitment text instructed students to submit a form to indicate their interest in participating in a 45-minute interview with a \$15 gift card as compensation. Only students who indicated on the initial interest form that they had previously attended tutoring hours were scheduled for an interview. Table 1 shows how the participants self-identified the frequency of their tutoring hours attendance when taking a CS course. Students were told beforehand that the interviews would be about their personal opinions regarding their experiences during tutoring interactions.

Table 1: Frequency at which participants attended tutoring hours.

Attends tutoring hours	N	%
A few times throughout the quarter	5	33.3%
About once or twice a week	6	40.0%
3 or more times per week	4	26.7%

In total, 15 students were interviewed. All of the interviews were conducted by the first author of this paper and followed a semi-structured interview format to allow the interviewer to ask students to elaborate on their responses. The structured interview questions can be found in Figure 1. Participants were all enrolled in CS2 at the time of the interview but were asked to provide answers relevant to their experience throughout all their programming courses so far.

The interviews were conducted remotely over Zoom in the last three weeks of the Spring 2022 term. The audio was recorded and then transcribed and anonymized for analysis per our approved Human Subjects protocol.

3.4 Qualitative Analysis

Phenomenography is a method for qualitative analysis used to capture the variation in perspectives and experiences [14]. We used phenomenography to identify the range of student expectations and desires.

Analysis was performed by the first three authors of the paper, who began by independently reading through the interview transcripts and extracting all the quotes that were relevant to our research question. Next, each of us identified common threads among the quotes that were pulled, and we compared our findings to come up with a set of categories to classify the different expectations and desires that we observed among students. We then went back through all the transcripts to extract quotes corresponding to our described categories. We subsequently discussed the process and

- From your perspective, describe in a few sentences what you think the job of a tutor is during lab hours.
- When you attend a tutoring session with a bug in your code, what are some things that you expect the tutor to do to help you?
- Do you believe there are any limits to the kind of help the tutor should give you? Is there anything the tutor shouldn't do to help you?
- Thinking of a positive tutoring experience you had, what made the experience positive for you?
- Thinking of a negative tutoring experience you had, what made the experience negative for you?

Figure 1: Structured questions asked by the interviewer.

further refined and improved our categories. Finally, once a consensus on the variation and descriptions of categories was reached, we repeated the process of extracting quotes and classified them according to the final categorizations.

3.5 Definitions and Framing

3.5.1 Expectations and Desires. As we read through the transcripts, a distinction emerged between students' expectations and desires. In our analysis, we chose to have a clear distinction between the two. Our definitions are as follows: Expectation: something that a student thinks that they should receive during a tutoring session, regardless of whether they would want it or not. Students typically wanted what they expected, and we will note when there is an exception. Desire: something that a student wishes for but does not necessarily expect.

3.5.2 The Course of Time. There are several different intervals of time that are relevant to students throughout the progression of a course. The first is the time span of the course itself, from beginning to end. Next, there is the timeline for each assignment in terms of when it is released, when a student starts working on it, and when it is due. Finally, there is the time span of an individual tutoring session and the different points within the session. These different intervals of time are important to note, as students' expectations and desires may vary depending on what point they are at within any of these timelines.

4 RESULTS

We group our results into two categories that represent students' expectations and desires during different points in a tutoring session: during the session and leaving the session, as well as a category for expectations and desires for the overall interaction.¹ We also contrast the results in the context of other timelines as necessary. Quotes from participants are labeled using anonymous identifiers (S1-S15).

¹Students' expectations and desires entering the session were about the entire interaction and therefore fall under the overall interaction.

4.1 During the Tutoring Session

4.1.1 Tutor helps the student to understand. Students expressed that it was important for the tutor to help them to understand how to solve the problem, but they had different ideas of how tutors should perform this task. For example, S14 expects the tutor to ask questions such as "how does this line impact the code?" because "that helps the tutor to [know] if I really understand the logic behind my purpose of writing it. If the purpose is not correct, they could improve my logic. If it is correct, then they will move to the next step." Other students expect that the tutor could sufficiently help them understand how to solve the problem without asking any questions. They expect that the tutor could accomplish this by providing a detailed explanation of the student's own code and why it has issues. "They'd explain the way or how it's actually supposed to be [...] to help us understand why our code is wrong" -S6.

4.1.2 Tutor guidance. There was widespread agreement among students that it is beneficial for them to think for themselves. They expect the tutors to make space for them to go through parts of the process on their own. This includes actions on the tutors' part such as giving hints ("they would give you some really good hints that will lead you to think" -S5), asking guiding questions ("ask me questions to lead me to realizing what mistake I made" -S13), and "giving them the tools to debug on their own" -S11.

Many students' desires change as the deadline for the assignment approaches. They expressed a desire for tutors to "give more direct help instead of saying, 'Oh, try to debug this way,' because sometimes, people are on that time crunch" -S11. S3 felt overwhelmed near deadlines, and "because I have to turn it in today, I just really can't think [...] clearly, or [think about] the hints that they're giving me, really that well."

4.2 Leaving the Tutoring Session

In this subsection, we first need to clarify a term, "the answer", which had a different meaning depending on the student. Once we have described the variance in student beliefs about their expected "answer", we then present the expectations and desires that students have at the conclusion of a tutoring session.

4.2.1 Context: Closing in on "the answer". Across the interviews, students had different ideas about what "the answer" is in the context of a tutoring session. Based on what students said, we identify several different stages in the process of working through a debugging problem, which are shown in Figure 2. Progressively, each stage gets more specific and closer to the final solution, the final stage being the exact code needed to correct the issue.

Broadly, all of these stages fall into the categories of "Identifying the Problem" and "Fixing the Problem", as shown in Figure 2. We will use the terms "conceptual answer" and "full programming answer" to refer to these categories, respectively.

Students differed in which stage in the process they viewed as "the answer". Some felt that they were being given the answer if the tutor told them the location of the problematic code, while others viewed only the last stage, the exact code fix, as the answer. The majority did not consider any of the problem-identification steps to be a part of the answer.

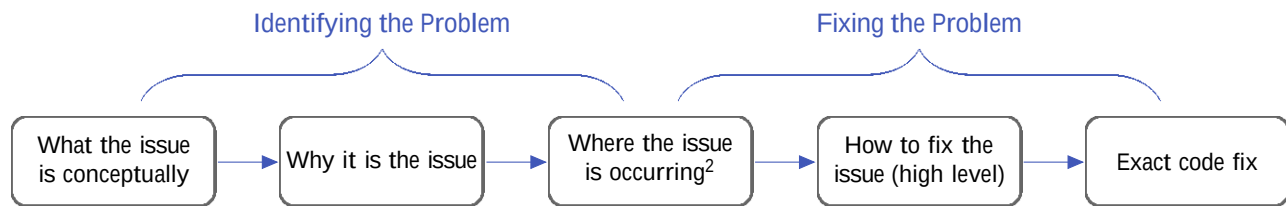


Figure 2: Different stages in the process of working through a debugging problem.

Throughout this section, we use the phrase “the answer” to refer to what a student believes to be the answer, regardless of which stage that is at.

4.2.2 Tutor gives the answer. In some cases, a tutor may provide what the student believes to be the answer. None of the students we interviewed expected that the tutor would always give only the answer, but several expect to be given the answer along with an explanation: “If they gave me the answer and described to me thoroughly how they got to that answer so I can replicate it later, I think that is ideal” -S8.

Other students desire to be given what they believe to be the answer, along with an explanation, only when they are extremely stuck. They consider themselves to reach this point under conditions such as “after [I ask] a certain amount of questions” -S10 and after a certain amount of time, up to “when I’ve basically just been bashing my head on the same portion of code for the past few hours” -S2 and as low as “not too long because usually when I understand something, I can understand it in a minute or two” -S5.

There was a widespread expectation that tutors should give the exact code solution for small non-conceptual errors. Students said that they want these to be resolved quickly because they view such an error as one that a student “commits by small mistake or just a bit out of concentration” -S14 and is therefore not worth spending too much time on if it does not involve the core concepts of the course.

Students’ expectations and desires are most commonly impacted by the approaching deadline. Many students admitted that they desire a direct answer near the deadline in spite of the long-term effect they know it would have on their learning. “If I come to a tutor session [near the deadline], I’d rather have the tutor tell me the answer so that I can finish my homework and turn it in on time. But if I go to a tutor session [...] before the due date, then I will just prefer to think about it for a while on my own so that I can learn better” -S7. Similarly, “If the deadline is very close, I would disregard any conceptual errors that I have, which is really bad. That is not something I want, but I would rather disregard my foundation knowledge than get a bad grade on my assignment. Which is really pathetic, but that’s how it is” -S15.

4.2.3 Student has a better understanding of the problem. Tutors do not always give the students the answer, and students do not always expect them to. However, students do expect to have made progress by the time they leave the session, whether they are able to solve the problem on their own during the session or

after. Some students expect that they will be able to come to the full programming answer (“even if you didn’t get the exact direct fix, you figured out how to do it and you know how to do it” -S10), while others only expect that they will be able to come to the conceptual answer (“So using the material, I can probably tie it back to what I was doing wrong and then somehow correct myself with it. [...] I can slowly build up, slowly and then just come to be able to correct my own mistake. It’s like cool” -S9.)

4.3 Overall Interaction

Students also have expectations and desires about what they will get out of the interaction as a whole.

4.3.1 Open dialogue. Some students expect the chance to converse with the tutor and have a two-sided discussion. “If I don’t have a decent conversation where it’s back and forth on the concept and of me trying to understand it, then I don’t think that time with that tutor was actually really helpful for me” -S6.

4.3.2 Getting the tutor’s perspective. Students expect to be exposed to a new perspective when they attend a tutoring session. For example, S4 stated that the open dialogue and conversation with the tutor “makes me step outside of my perspective, outside of what my mindset is currently, and view it from an outsider perspective, who is viewing my code and just reading it for the first time”.

4.3.3 Session length. There were differing opinions on the ideal length of a session. Some desire a short and concise session (“I liked the fact that it was all very quick [...] quick is always better” -S10), while others prefer a longer interaction when possible but expect that “tutors can’t really fully help you due to a queue of more students, right? So, they can only help you for a short amount of time” -S1. On the flip side, some students both prefer and expect that their session is not “short and quick, along the lines of just giving me a direct answer just right away, and then me not really understanding how to fix similar errors later on” -S13.

4.3.4 Impact of a long queue. Students expect the help they receive from the tutors to be affected by the number of students waiting to get help. “I think more consistently I’ve gotten better help when the tutor hours are more free and open. That way the tutor wouldn’t feel so pressured to keep everything so compact as far as the guidance. Whereas, when I go during the times where a lot of students attend, then the help shifts more towards just giving an immediate answer or correction, rather than the more gradual advice” -S13.

4.3.5 Session is engaging and interactive. Several students expect to be heavily engaged during the session (“Engagement is

²“Where the issue is occurring” could be interpreted as either identifying the location of the issue or figuring out exactly which part of the code needs to be fixed, thus it could be in either category.

very vital in interaction because [...] then only would the student understand” -S14). One student, S7, desires but does not expect the tutors to engage them. “I would prefer them [to be engaging], but then I wouldn’t say they should do it as I wouldn’t call it an obligation to stay engaged [because] the wait time is really long, and I really think that can be resolved as a higher priority than to have the interactive engagement with the with the students.”

4.3.6 Tutor interpersonal skills. Students mentioned expectations for tutors to be patient (“when I go in, I’m very like ‘what the heck is going on’. So, I’d like it when, I think they should be patient” -S8.) and desires for tutors to be empathetic and relatable (“they are able to empathize with you while trying to solve the problem [...] I think some of the most positive tutoring experiences that I’ve had, have just been when the tutor has humanized themselves, in the sense that they don’t portray themselves as a teacher, but more as a peer you are having a conversation with” -S4) during tutoring sessions. Additionally, one student expects that the tutor will “not judge you if you don’t know something. [...] they are always down to help you with the most basic thing” -S12.

5 DISCUSSION

5.1 “The Answer” and Debugging

Many students’ definitions of “the answer” did not include the conceptual answer, so although they did want the tutor to guide them, they were content with the tutor identifying the problem for them and only providing guidance towards the full programming answer. When students do not engage in the problem-identification stages, they skip a significant part of the debugging process and do not learn how to locate their own errors. This could be a cause for students in earlier courses having more trouble finding their bugs than fixing them [13].

If students do not view figuring out the conceptual answer as a part of the problem-solving process that they need to engage in themselves, they might expect the tutors to find their problems for them. If a tutor does give a student the conceptual answer and then guides them to the full programming answer, the student learns how to fix their problem but does not learn how to find the problem on their own. This may create a cycle of dependency where students do not learn the problem-identification process and continue to rely on tutors to do it for them.

5.2 Impact of Demand for Tutors

At our institution, tutors are often in heavy demand, particularly on the day or two prior to a programming assignment deadline. This demand places pressure on tutors to complete the tutoring interaction quickly [17], and some instructors even limit the amount of time a tutor is allowed to spend with each student. This limit has been explained as solving two problems: 1) a tutor help session is designed to get a student “unstuck”, not find the bug for them (i.e., tell them the next step in the debugging process and move onto the next student) and 2) ensures many students get help. This policy, although pragmatic, introduces its own set of challenges in light of the expectations of students identified in this study.

If some students believe what is helpful for their learning would be an “engaging” session where the tutor explains the relevant

concepts and helps the student find the mistake in their thinking, there may not be enough time with the tutor for that interaction to take place. However, students also express the desire to get the full programming answer in certain cases, and a tutor can provide the full programming answer in such a short window of time. Given student desires and tutor time constraints, this helps explain the recent finding that many such help sessions result in students receiving the answer [15].

Moreover, as mentioned previously in 5.1, if students become accustomed to receiving the conceptual answer without learning the process, students can become dependent on the tutors for help, which increases the length of the queue for help, which reduces the quality of help the tutor can provide.

5.3 Conflicting Desires

The majority of students stated that if the tutor could guide them to solve their problem on their own, it would be more helpful for their learning than if the tutor simply told them the solution. However, many of those students had desires that did not align with that sentiment. They still wanted the tutor to give them the answer. We elaborate on some reasons for this disconnect below.

5.3.1 Students view their learning as a lower priority. The view that learning is secondary to getting “the answer” may be especially prominent for the students who desire more direct help or desire the answer near the assignment deadline. Students state they are under a lot of pressure to finish their assignments before the deadline, so they prioritize that over their learning. S15, who is also quoted at the end of 4.2.2, described it as “The knife is hanging on your neck. You don’t really think about all the good things you can do. So you just want to get through it, get done with it. ‘Just survive’ kind of instincts.”

Of course, for instructors, the prioritization of completing an assignment over learning the relevant concepts and programming skills is concerning, particularly as there is evidence that students seeking to master the course material perform better in computing courses [28]. Instructors could try to alleviate this issue by designing the course policies to de-emphasize the need to get as many possible points before the deadline, lower the weighting of programming assignments, and/or take other steps to encourage mastery of the material, however mastery learning without clear deadlines faces its own challenges [19].

5.3.2 Students may not know what is best for their learning. In some interviews, students stated that getting the answer would be beneficial for their learning, which conflicts with previous studies [26]. Several students specifically stated that it would benefit their learning if they were given the answer, citing that they would review the answer to understand it on their own after the session. Although the students may fully intend to revisit the problem at a later time, not all students will end up doing so. Prior work by Liao et al. found that lower-performing students will often stop working on their assignments once they get the provided test cases to pass, without taking the time to understand why their solution passes the test cases [16]. Lower-performing students are also unlikely to work on assignments after the deadline [16]. This suggests that if

tutors do simply give the answer to students without any form of explanation, it may impact lower-performing students more [16].

Students being unaware of what is better for their learning is well established in the learning literature. For example, students sometimes report lower satisfaction with active learning techniques despite stronger learning outcomes [1, 25] (or even the inverse scenario [6]) and students often believe they will have better retention of what they learn using blocked practice rather than interleaved practice, when interleaved practice is better for long-term retention [2, 24]. To address this issue, instructors could set standards for tutoring sessions and share them with students, making clear that the expectation aligns with what we anticipate is better for their learning.

5.4 Implications

Our findings indicate that although students may have the desire to learn, there are other outside factors that influence their expectations and desires for tutoring sessions. Students who do not recognize the importance of identifying their own problems may not realize that debugging is a learning goal of the assignment. Students commonly acknowledged that the purpose of completing an assignment is to help them learn the concepts of the course, but they did not mention anything about learning by finding and correcting their own errors. Additionally, some of these students will eventually become tutors and will carry this mindset with them, which may explain why Krause-Levy et al. observed many interactions in which the tutors did not teach or mention the process of debugging [15]. In response to this issue, instructors could communicate these learning goals to both students and tutors, emphasizing the importance of going through the process of debugging and explaining to students how it benefits their learning.

Students are under a lot of pressure to perform well academically, and they work against many time constraints such as assignment deadlines. Thus, it might be difficult for them to take the necessary time to prioritize their learning. The assignment deadline pressures are likely two-fold: 1) the deadline pressures students to have their assignment done by that day, and 2) a long queue on the day of the deadline leads to pressure on the tutors to spend less time with each student. Students expressed feeling this stress, and previous work has shown that tutors feel this stress as well [17, 22]. With both the tutor and the student being burdened by these challenges, there may be relief on both ends if the tutor provides a quick fix for the student. However, this is at a detriment to the student's long-term learning and understanding.

Our institution has a tutor training course that all tutors must take during their first quarter as a tutor. Research has shown that training courses for instructional assistants both help them to make better critical observations about various aspects of teaching [12] and increase student satisfaction towards instructional assistants [23]. Although tutor training courses may be useful in attempting to address some of these issues, it may be insufficient to only communicate these expectations to tutors during their first quarter of tutoring. It may be necessary to provide tutors with regular reinforcement and reminders to prioritize students' learning during tutoring sessions and specific instructions for how to engage in good teaching practices. Thus, instructors should set clear

guidelines for their tutors to follow that will help them provide students with the help that would be best for their learning.

5.5 Limitations

Our data came from a subset of students in one course at one institution, thus our findings may not encompass the views of all students. Future work should replicate this study at a variety of institutions and courses to further understand the student perspective across institutional settings and positions in their academic careers.

6 CONCLUSION

Prior work has shown that student-tutor interactions are not as effective as instructors might have hoped and that there is a high cognitive load and social pressures on tutors during these interactions. Our work extends our understanding of these interactions through 15 interviews of students in a CS2 course, for which we used phenomenographic analysis to identify the range of students' expectations and desires during tutoring sessions. We found that students' short-term expectations and desires for the tutoring sessions often conflict with their long-term desire to learn. Potential reasons for this disconnect include (1) factors that students are unaware of and (2) students viewing other short-term constraints as a higher priority than their learning. Future work should aim to bring forth ways to mitigate these issues and implement interventions to help students understand what would be the most beneficial for their learning.

7 ACKNOWLEDGMENTS

We thank the reviewers for their helpful suggestions. This work is supported in part by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650112, NSF Award #2121592, as well as a UCSD Sloan Scholar Fellowship and a Gates Millennium Scholarship.

REFERENCES

- [1] Peter J. Allen and Frank D. Baughman. 2016. Active Learning in Research Methods Classes Is Associated with Higher Knowledge and Confidence, Though not Evaluations or Satisfaction. *Frontiers in Psychology* 7 (2016). <https://doi.org/10.3389/fpsyg.2016.00279>
- [2] Alan D Baddeley and DJA Longman. 1978. The influence of length and frequency of training session on the rate of learning to type. *Ergonomics* 21, 8 (1978), 627–635.
- [3] Maureen Biggers, Tuba Yilmaz, and Monica Sweat. 2009. Using collaborative, modified peer led team learning to improve student success and retention in intro cs. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)*.
- [4] Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13, 6 (1984), 4–16.
- [5] Mark J. Canup and Russell L. Shackelford. 1998. Using software to solve problems in large computing courses. *ACM SIGCSE Bulletin* 30, 1 (1998), 135–139.
- [6] David E Copeland, Jenna R Scott, and Jeremy Ashton Houska. 2010. Computer-based demonstrations in cognitive psychology: Benefits and costs. *Teaching of Psychology* 37, 2 (2010), 141–145.
- [7] Sue Ellen DeChenne, Natalie Koziol, Mark Needham, and Larry Enochs. 2015. Modeling sources of teaching self-efficacy for science, technology, engineering, and mathematics graduate teaching assistants. *CBE—Life Sciences Education* 14, 3 (2015), ar32.
- [8] Adrienne Decker, Phil Ventura, and Christopher Egert. 2006. Through the looking glass: reflections on using undergraduate teaching assistants in CS1. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. 46–50.
- [9] Paul E. Dickson. 2011. Using undergraduate teaching assistants in a small college environment. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*.

- [10] Paul E. Dickson, Toby Dragon, and Adam Lee. 2017. Using undergraduate teaching assistants in small classes. In *Proceedings of the 48th ACM Technical Symposium on Computer Science Education (SIGCSE '17)*.
- [11] Ronald Erdei, John A. Springer, and David M. Whittinghill. 2017. An impact comparison of two instructional scaffolding strategies employed in our programming laboratories: Employment of a supplemental teaching assistant versus employment of the pair programming methodology. In *Proceedings of the 47th IEEE Frontiers in Education Conference (FIE '17)*.
- [12] Francisco J. Estrada and Anya Tafilovich. 2017. Bridging the Gap Between Desired and Actual Qualifications of Teaching Assistants: An Experience Report (ITICSE '17).
- [13] Irvin R. Katz and John R. Anderson. 1987. Debugging: An Analysis of Bug-Location Strategies. *Human-Computer Interaction* 3, 4 (1987), 351–399.
- [14] Päivi Kinnunen and Beth Simon. 2012. Phenomenography and grounded theory as research methods in computing education research field. *Computer Science Education* 22, 2 (2012), 199–218.
- [15] Sophia Krause-Levy, Rachel S. Lim, Ismael Villegas Molina, Yingjun Cao, and Leo Porter. 2022. An Exploration of Student-Tutor Interactions in Computing. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1 (ITICSE '22)*.
- [16] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold, and Leo Porter. 2019. Behaviors of higher and lower performing students in CS1. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. 196–202.
- [17] Julia M. Markel and Philip J. Guo. 2021. Inside the Mind of a CS Undergraduate TA: A Firsthand Account of Undergraduate Peer Tutoring in Computer Labs. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*.
- [18] Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *Proceedings of the 15th ACM Conference on International Computing Education Research (ICER '19)*.
- [19] Claudia Ott, Brendan McCane, and Nick Meek. 2021. Mastery Learning in CS1 - An Invitation to Procrastinate?: Reflecting on Six Years of Mastery Learning (ITICSE '21). 18–24.
- [20] Inna Pivkina. 2016. Peer learning assistants in undergraduate computer science courses. In *Proceedings of the 46th IEEE Frontiers in Education Conference (FIE '16)*.
- [21] Stuart Reges, John McGrory, and Jeff Smith. 1988. The effective use of undergraduates to staff large introductory CS courses. *ACM SIGCSE Bulletin* 20, 1 (1988), 22–25.
- [22] Emma Riese, Madeleine Lorås, Martin Ukrop, and Tomáš Effenberger. 2021. Challenges Faced by Teaching Assistants in Computer Science Education Across Europe. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 547–553.
- [23] Guido Rößling and Jacqueline Götz. 2018. Preparing First-Time CS Student Teaching Assistants (ITICSE 2018).
- [24] Dominic A. Simon and Robert A. Bjork. 2001. Metacognition in motor learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 27, 4 (2001), 907.
- [25] OJ Sizemore and Gary W. Lewandowski Jr. 2009. Learning might not equal liking: Research methods course changes knowledge but not attitudes. *Teaching of Psychology* 36, 2 (2009), 90–95.
- [26] Noreen M. Webb, Jonathan D. Troper, and Randy Fall. 1995. Constructive activity and learning in collaborative small groups. *Journal of educational psychology* 87, 3 (1995), 406.
- [27] A.B. WeikleDee. 2016. More insights on a peer tutoring model for small schools with limited funding and resources. *Journal of Computing Sciences in Colleges* 31, 3 (2016), 101–109.
- [28] Daniel Zingaro, Michelle Craig, Leo Porter, Brett A. Becker, Yingjun Cao, Phill Conrad, Diana Cukierman, Arto Hellas, Dastyni Loksa, and Neena Thota. 2018. Achievement Goals in CS1: Replication and Extension (SIGCSE '18). 687–692.