Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #48099

Students' Experiences of Learning Technical Writing in Computer Science Courses: Perspectives on Assessment

Dr. Meghan Allen, University of British Columbia

Meghan Allen is an Associate Professor of Teaching in Computer Science at the University of British Columbia, where she has been teaching since 2007. She holds a PhD in Curriculum and Pedagogy and is interested in designing curriculum, understanding student experience, and mentoring future teaching-focused computer science faculty members.

Students' Experiences of Learning Technical Writing in Computer Science Courses: Perspectives on Assessment

1 Abstract

Post-secondary computer science students acquire a range of skills and competencies, including professional skills such as technical writing. These professional skills and other program learning outcomes are assessed via students' course work. As assessment practices continue to evolve, there is an increasing shift towards the use of automated assessment tools (AATs) in post-secondary computer science education. Scholars have studied AATs and their use, but few have considered how substantial use of AATs affects students' learning of skills that are not assessable by AATs. The shift to heavier use of AATs motivates considering the impact that assessment practices have on the student experience but limited research has examined this topic. This paper begins to fill that research gap by addressing the research question: How do course assessment practices affect students' perspectives of learning technical writing?

I conducted an interpretive qualitative study, grounded in Lave and Wenger's Situated Learning Theory and Social Theory of Learning, with 10 third and fourth-year computer science student participants. I used reflective journal writing and beginning-of-term and end-of-term interviews to gather rich data on the student experience. I generated themes from the data corpus via Braun and Clarke's reflexive thematic analysis and found that students are conflicted in their desire to learn technical writing and that their beliefs are influenced by assessment practices. They believe that technical writing is important for their careers and they want to learn technical writing in computer science courses, however, they perceive that technical writing is not assessed often or deeply enough and shared that course assessment practices affect the learning activities that they prioritize.

2 Introduction

Communication skills are integral to professional computer scientists' success [1], [2], [3]. These communicative skills and competencies are usually integrated into program learning outcomes, which are assessed via students' course work; recently, assessment in post-secondary computer science programs is shifting towards the heavier use of automated assessment tools (AATs) that tend to assess functional correctness of code. Scholars have studied AATs and their use, but few have considered how increased use of AATs affects students' perspectives on their learning of skills that are not assessable by AATs. In this study, I focused on technical writing as one aspect of communication that is critical to computer scientists' professional practice [4] that is not generally assessed by AATs.

This paper presents partial results from an interpretive, qualitative study grounded in Lave and Wenger's Situated Learning Theory [5] and Wenger's Social Theory of Learning [6] that broadly investigated computer science students' experiences and perspectives of learning technical writing. Results more closely related to these theories will be presented in another paper. The goal of this paper is to examine students' perspectives in relation to course assessment practices.

2.1 Theoretical Foundations

Post-secondary computer science education researchers hold diverse epistemological and theoretical perspectives [7], [8], [9], although, as with much higher education research, these fundamental perspectives are often unexpressed in published research articles [10], [11]. Epistemology and theoretical perspective, even when unacknowledged, affect researchers' tacit beliefs and underlie their theories of learning. Since this study uses Braun et al.'s [12] reflexive thematic analysis and they recommend that "[r]esearchers should always reflect on and specify the philosophical and theoretical assumptions" [13, p. 11], in this section I share my philosophical and theoretical assumptions.

Epistemologically, I am a constructionist; I believe that "all knowledge, and therefore all meaningful reality as such, is contingent upon human practices, being constructed in and out of interaction between human beings and their world, and developed and transmitted within an essentially social context" [14, p. 42]. Since I believe that knowledge is constructed via humans' interactions with the world and with each other, I believe that an individual's identity, history, and subjectivity influence their understanding of the world and what they learn from an experience. Therefore, my subjectivity as a researcher cannot be separated from my research. My experiences bring a particular lens to this work. Others who had similar research goals may have brought different perspectives to the work or constructed different meaning from the study. I use the first-person in this paper to acknowledge that this research project is tightly related to who I am as a researcher.

My theoretical perspective is interpretivism. I believe that interpretations are "contextually dependent on the history and culture that influences how each individual interprets and makes meaning of their world" [15, p. 1173]. Therefore, I recognize that each individual who participates in an interaction or observes an action or event may come to their own interpretation and understanding.

3 Literature Review

Technical Writing in Computer Science Courses

Integrating writing into the computer science curriculum presents a unique opportunity for learning because "writing as process-and-product possesses a cluster of attributes that correspond uniquely to certain powerful learning strategies" [16, p. 122]. Fell et al. [17] argue that effective communication skills "are best acquired while learning information and ideas across a broad range of arts and sciences courses and in-depth, within a specific discipline" (p. 204).

To the best of my knowledge, limited research has examined students' experiences of learning technical writing in computer science courses. Hazzan and Har-Shai [18] conducted surveys to

gather student perspectives' about their course that taught a broad range of soft skills, including technical writing. They found that students believed the course improved their soft skills and that most would take another course on soft skills. Their survey did not distinguish between writing-related skills and other soft skills. Zhang et al. [19] surveyed students who took courses with embedded computer science-specific writing instruction. They conducted a thematic analysis on the qualitative data and found some evidence that students' perception of writing changed after completing writing activities designed by the project. von Briesen [20] integrated four writing assignments into an Artificial Intelligence course and conducted pre- and post-term surveys to gather students' perceptions on the importance of writing in the discipline of computer science. She found that "on average, students agreed that writing in the discipline is important" (p. 119). These three studies found that computer science students believe learning writing-related and other soft skills is important but there remains limited research that explores their perspectives of this learning. My study begins to address the research gap by examining students' experiences of learning technical writing in computer science courses.

3.1 Automated Assessment Tools in Computer Science Courses

Automated Assessment Tools (AATs) have been used in computer science courses for decades, but have become more common recently [21], [22]. Scholars have studied how AATs should be configured, for example, whether and how milestone due dates should be implemented, what kind of feedback should be given, when feedback should be given, whether there should be limits on the number of submissions, and whether there should be grade penalties for any usage scenarios [21], [22], [23], [24], [25], [26], [27], [28]. Luxton-Reilly et al. [22] found that "providing automated functional testing was the most common form of assessment" (p. 8) but that automated testing was not limited to testing functionality. Some AATs assess code style [21, 29], but not all code quality issues can be automatically assessed. AATs can also measure test correctness and thoroughness and check for plagiarism [21]. In work conducted after data collection for this study was complete, scholars have integrated Large Language Models into AATs [30], [31], [32]. The functionality of an AAT puts constraints on an educator's grading practices.

In consideration of the AAT literature, Luxton-Reilly et al. [22] state "there is no broad analysis of the teachers' motivations for adopting AATs, how it is used to support teaching practice, teacher perceptions of AATs, or the impact of AATs on student behaviour and attitude." (p. 3) This study gathered student perspectives on how assessment practices influences their behaviours and priorities.

4 Context

This study was conducted at the University of British Columbia (UBC), a large, research-intensive, Canadian university. Over 2500 undergraduate students major in computer science; the major has doubled in size in the last decade. Many faculty members at UBC have transitioned their courses to use Automated Assessment Tools so it is becoming less common for students to have their course work manually assessed by faculty or teaching assistants.

5 Research Design

This study answers the research question: How do course assessment practices affect students' perspectives of learning technical writing?

5.1 Research Ethics

The research study was approved by the UBC Behavioural Research Ethics Board before the research commenced.

I am a computer science faculty member at UBC. In this study, I was conscious of the power that I have in relation to the participants who are UBC computer science students. In any study, there is a risk that participants may consciously or subconsciously be affected by what the researcher wants to hear [33]. In my study, my dual role may have heightened this risk. Upon reflection, I believe that participants openly shared their experiences.

5.2 Participants

Ten third- and fourth-year computer science students participated in this study. I chose to limit participation to 10 students because I believed their experiences would provide sufficiently rich data. The participants are diverse across their countries of birth, ethnicities, genders[1], work and academic experiences, and languages spoken. See Table 1 for details. During the data-collection term all participants were taking at least two computer science courses, and for each student, at least one of those courses was an upper-level course. The participants' computer science course registrations spanned two required second-year courses, all three required third-year courses, seven elective third-year courses, and five fourth-year courses. Each participant received $100 in gift cards to a preferred vendor for their participation in the study. All names used in this paper are pseudonyms.

5.3 Data Collection

I collected data over one term through beginning-of-term and end-of-term semi-structured interviews and weekly reflective journals.

Each 40-60 minute initial interview followed the same plan but differed in its execution as I asked follow-up questions that were dependent on participant's responses.

Participants were asked to write for 15-20 minutes per week in response to reflective prompts. Nine participants wrote in an online survey form and one chose a physical journal. The first prompt was sent to each participant after their initial interview. For the group, these prompts were sent over a three day period. The remaining 11 prompts were sent to all participants via email once per week. For each participant, if they had unanswered prompts, I included a reminder to these prompts in their weekly email. Participants responded to the prompts when they were able to; some responded weekly while others responded to multiple prompts at once.

[1] I asked participants if they were a man, woman, non-binary person, or if there was another way they'd prefer to describe their gender. They did not have to answer.

Table 1: Participant Demographics

Name	Gender	Secondary School Location	Languages Spoken	Native English Speaker	Fluent in English
Adrian	non-binary person	in Canada	2	No	Yes
David	man	in Canada	2	Yes	Yes
Etson	man	not in Canada	2	No	No
Jason	man	in Canada	1	Yes	Yes
Krish	man	not in Canada	2	No	Yes
Nadir	man	not in Canada	2	No	Yes
Nico	man	in Canada	2	No	Yes
Phyllis	queer; somewhere between woman and non-binary	in Canada	2	Yes	Yes
Victor	man	in Canada	4	Yes	Yes
Wui	woman	not in Canada	3	No	Yes

The concluding interviews were also semi-structured and took 40 to 65 minutes.

After all the initial and concluding interviews were complete, I uploaded the audio recordings to NVivo Transcription [34] and they were automatically transcribed. I reviewed each transcript and made edits, where necessary, for accuracy. I transcribed the entries from the physical journal at the end of the data collection period.

5.4 Data Analysis

I used reflexive thematic analysis to generate themes from the data corpus.

Thematic analysis is a broad term for a number of approaches to find meaning in a qualitative data corpus [12]. Braun et al. [12] describe three approaches to thematic analysis: a coding reliability approach, a codebook approach, and a reflexive approach. Reflexive thematic analysis sits within a qualitative framework and emphasizes "meaning as contextual or situated, reality or realities as multiple, and researcher subjectivity as not just valid but a resource" [12, p. 848]. My theoretical perspective and research questions align with Braun et al.'s [12] approach to qualitative research, where meaning is not universal, but instead influenced by the context of the research and the researcher or researchers themselves. Therefore, reflexive thematic analysis is the most appropriate style of thematic analysis for me as a researcher and for this research project. In particular, I align with Braun et al.'s [12] views that researcher subjectivity is a resource to be valued, rather than a limitation to be mitigated.

5.4.1 Data Analysis Procedure

I followed Braun and Clarke's six-step analysis procedure [13] to conduct a reflexive thematic analysis. The analytic steps included:

1. data familiarization and writing familiarization notes;

2. systematic data coding;
3. generating initial themes from coded and collated data;
4. developing and reviewing themes;
5. refining, defining and naming themes; and
6. writing the report.

Braun and Clarke [13] state that these steps are not a linear process, instead, they should be followed iteratively, moving between steps as required. The goal of this process is to generate themes, where themes are "patterns of shared meaning, united by a central concept or idea" [13]. The term *generate* is chosen deliberately. A key tenet of reflexive thematic analysis is that the themes are actively constructed and created by the researcher(s); they are not already existing in the data waiting to emerge. A more thorough description of my analytic process is available in my dissertation [35].

5.5 Quality Considerations

Since there is more than one way to conduct thematic analysis, as described in Section 5.4, there is not a single approach to considering quality in thematic analysis. For this project, I rely on the principle of trustworthiness [36] and the reflexive thematic analysis quality considerations put forth by Braun and Clarke [13].

To enhance the trustworthiness and credibility of my study, I used Lincoln and Guba's strategies of prolonged engagement, member checks, collecting data from multiple sources, and peer debriefing [36]. A detailed discussion of how I engaged with these strategies is available in my dissertation [35].

Two key reflexive thematic analysis quality considerations raised by Braun and Clarke [13] are:

- The analysis procedures must be consistent with the "paradigmatic and epistemological assumptions" (p. 2) and those assumptions must be articulated.
- Researchers must explain their use of and orientation to reflexive thematic analysis.

In this paper, I have shared my epistemological and theoretical perspectives and explained my interpretive orientation to reflexive thematic analysis.

5.6 Limitations

All research has limitations; researchers' interpersonal skills, subjectivity, and research skills affect and limit outcomes. Qualitative research is "saturated with more concealed forms of power than quantitative and experimental research" [37, p. 589]; therefore, as a qualitative researcher, I must be aware of power relations and their influence on research. Readers must also consider this limitation.

The students who participated in this study are diverse in many ways, such as their countries of birth, ethnicities, genders, work and academic experiences, and languages spoken, but they share

commonalities as well. All participants are high-achieving and engaged students who generally enjoy writing. With a different population of participants, I would have gathered different data and constructed different themes. Another researcher may have elicited different responses from participants, but I believe that my perspective is a resource that I bring to this project.

6 Results

In the overarching research study [35], I generated eight themes with one that is relevant to the research question discussed in this paper. The relevant theme is 'Students have a conflicting desire to learn technical writing'. I found that students believe technical writing is important but prioritize learning activities that will be more heavily assessed.

In this section, I discuss students' conflicting views on the importance of technical writing and the ways in which course assessment practices influence their views. In individual participants and across the group of participants, there was evidence of nuanced and sometimes contradictory beliefs. Participants believe that learning technical writing is important, but also hold views that technicality (e.g., programming, designing technical solutions) is more important than technical writing and other forms of communication. Participants also recognized that there are curricular tradeoffs and that instructors are constrained when they decide what to include and exclude in their courses. Participants expressed that they wished technical writing was assessed more often and more thoroughly; participants prioritized the aspects of an activity that would be graded.

6.1 Technical Writing is Important

There was strong support of the idea that technical writing was important for participants' future careers, and this was true for all participants across a variety of desired career paths. For example, Krish said, "since ... a big chunk of the job does involve technical writing and getting it done [on a] day to day basis, I think, would be very important as a computer scientist".

A wide range of tasks that require technical writing skills were mentioned when participants discussed the importance of technical writing. They discussed tasks related to developer-facing software documentation, writing emails about technical content, writing design documents, the ability to communicate with diverse team members such as people who may or may not have a technical background, writing user-facing software documentation, and creating clear commit histories in GitHub.

6.1.1 Technical Writing Creates Career-Related Opportunities

Participants believed that technical writing and communication skills were an expectation for those entering industry positions and that career progress via promotion was more likely to be given to someone with strong communication skills. David believed communication skills were necessary for getting a job: "for [an] industry job ... it's like table stakes". Victor focused on career progress: "when you come out into the real world, ... the people who get promotions and move their careers upward; it's people who communicate well."

One participant thought that they get more opportunities than their peers because of their communication skills. They didn't think this was fair, but recognized that they may be preferred by employers or collaborators over others who have weaker communication skills.

Jason believed, "it's a transferable skill, so it's helpful if I decide I don't want to do computer science anymore."

6.1.2 Strong Technical Writing Allows Your Work to be Recognized

A number of participants expressed that technical writing is a method for communicating their ideas and having their work recognized. Phyllis said, "if I just write a bunch of code but no one knows how it works, they won't be able to understand what I've done, build off of it, critique it, you know, actually recognize me for my work." Nico and Victor shared similar ideas: "if you can't communicate what you're doing to other people, they don't really understand how important it is the work you do" (Nico) and "if you can't communicate what value you're bringing to somebody else, you're just going to get swept under the rug" (Victor).

Participants also discussed career-related situations where a mix of written and verbal communication skills would be valuable. In particular, they mentioned scenarios like explaining their ideas to colleagues and trying to convince their colleagues to make a particular decision, or technical interviews where the candidate may be asked to solve a problem and share their thinking process out loud while working on the solution on a whiteboard. Interestingly, participants were primarily focused on work-related contexts in which written or verbal communication skills would be valuable. Participants discussed a few educational situations in which these skills would be valuable – research papers, presentations, group work – but predominately discussed benefits for future career success.

Many participants discussed the ability to communicate in writing with non-technical people or people who are not computer scientists. For example, Jason said, "being able to explain what you're doing to non-technical people is extremely valuable". Victor believed that explaining the work is crucial: "you can write all the code you want but if you can't explain any of it, like what good is that to anyone?"

6.1.3 Technical Writing and Other Technical Skills are Interrelated

Most participants shared the belief that technical writing skills and other technical skills are interrelated. The participants stated that strong technical skills are necessary for strong technical writing. For example, Jason said, "I do think you need strong technical skills to be a very good technical writer". He uses the quality of his technical writing to judge his competence in the technical domain: "I perceived the quality of my writing as a very strong indicator of my understanding of the domain". Victor explicitly discussed the interplay between learning technical material and the ability to communicate the technical material: "[b]ut really, they go hand in hand, right? Like how much hard technical material can you learn if you're not communicating it properly."

Multiple participants stated that technical writing helped them organize and clarify their thoughts. For example, Nadir and Jason are more able to find their mistakes when they put their thoughts in

writing: “when you actually put it in writing, you figure out your mistakes” (Nadir) and “you look at it and you see the problems that arise from it” (Jason). Further, Jason finds that he has to organize his thoughts before writing with a partner: “in order for whoever figured it out to explain it to the partner, we need to have already organized the thoughts in a way that’s conducive to writing”.

Wui believes her technical skills hold her back from being an excellent technical writer: “I just have a lot of gaps in my technical knowledge to be really good and proficient at technical writing.”

6.2 Participants Desire More Curricular Attention but Recognize Curricular Tradeoff

Most participants wished there was a stronger focus on learning technical writing in computer science courses. Phyllis explicitly stated that she wished professors “would just spend more class time on writing and make it a graded part of their courses”. Nico would appreciate more time to practice. He said, “I wish we had more opportunities to do technical writing. Little comments such as git issues or commit messages are not enough to practice.” After a code review at work, Wui wished she had learned more technical writing skills in her courses: “it’s embarrassing, like, I’m sharing my screen; this is what I’m writing. I feel like there should be a way for me to learn that in class.”

Despite wishing that professors spent more class time teaching technical writing, Phyllis was conscious of the fact that it would take away from other curriculum and that professors may prioritize other material. She said, “if they had to divide their course time and prioritize what to teach they probably want to go for something more technical ... first and then have the technical writing later.” Others mentioned the same tradeoff, for example, David said, “a few professors know that it’s important. But there’s just so much material to go through”. Jason also stated that he thought there should be more courses on technical writing, while acknowledging that “for all of the [computer science] courses I have taken, the course schedule is tight and technical writing doesn’t reasonably fit within the scope of the material.”

Wui noticed when one of her professors spent class time on technical writing. For her, that demonstrated his belief in the importance of that kind of technical writing and that it was possible to teach these technical writing concepts in a limited amount of time. She said, “he would even tell us, this is the importance of comments, like good comments does this. like, ... what does this line mean in a comment? we spent 30 minutes talking about comments.”

6.3 Technical Writing Was not Assessed Frequently or Deeply Enough

Participants consistently stated that their technical writing was not assessed often or deeply enough. For example, Victor said, “there isn’t really enough done in a lot of our classes to assess the clarity of our writing.” When technical writing was graded, they often found it to be a small percentage of their grade. For example, Nadir shared that “it was two percent of our grade, so very small and very little emphasis on how we wrote it, but rather what we were reflecting upon.” Adrian and Phyllis believed that the assessment of their work was too easy. Adrian earned full marks regardless of the quality of their writing, which meant they didn’t learn much from the assessment: “we weren’t being graded for anything aside basic understanding, I don’t think I

learned all that much". Phyllis didn't have to put effort into improving her group's work because the TA's feedback didn't suggest that improvement was expected: "we weren't really sure that the demands of the course required us to actively improve what we had already done."

Phyllis discussed two computer science courses in which she had learned about technical writing. She was assigned a reading on how to write in a Human Computer Interaction course, but it wasn't assessed. The lack of assessment made her question its importance. She didn't think it was important to course staff because "it doesn't seem like ... we were going to be penalized ... for not reading it, ... so I can't even be sure that they were really thinking it was important". She was also assigned a module in an Applied Machine Learning course on how to communicate results. It was also not assessed: "[t]here was a very nice primer about how to explain things effectively and then we have examples and then discuss it a bit in class but it was never tested."

Multiple participants stated that their written assignments are usually assessed on technical correctness and not on technical writing. Jason relayed an experience with assessment of a written assignment: "all of those written assignments are basically ... just being graded on correctness." Etson also finds that feedback and assessment focuses on correctness: "TA only grades on the correctness of the answer. Therefore, I have not received feedback on grammar, meaning, or organization."

Krish discussed the assessments he experienced; he believed that the lack of assessment of technical writing meant that what was important was whether the code works. He said, "[a course] did have a lot of coding but the code could be based on what the person, like there was no check about technical writing or stuff like that. It was more about as long as the code works." Phyllis explicitly stated that her perception of her learning was influenced by the lack of assessment: "[s]ince the mechanics of writing were not assessed or a focus, I do not feel I learned much."

6.3.1 Participants Appreciate Assessment and Feedback on Technical Writing

Wui wished that technical writing was assessed: "I wish I did [get assessed] because we spent so much time making those documents". Nico and Wui would appreciate feedback: "ordinary written assignments don't mark technical writing, only content, and I'd like to get feedback for improvement" (Nico) and "we don't really get a lot of feedback on the technical writing part." (Wui)

When Wui had her technical writing assessed, she found it to be a meaningful lesson. She said, "I thought the grading was harsh, but it was a good lesson" and elaborated to say: "it was very rewarding to have someone to take the time and give us feedback ... and the fact that we would be docked off marks if our documentation was unclear and the TA couldn't understand our writing."

6.4 Participants Prioritize What is Assessed

Participants expressed that assessment practices influence their learning and how they prioritize course activities. Despite consensus that technical writing was important, multiple participants mentioned that if technical writing is not assessed, they do not prioritize it. For example, Nadir

said, “because it’s not marked, I’m not too concerned about it”. Krish shared a similar perspective: “the comments part, marks weren’t associated with so it wasn’t really important for us to write them.”

Krish also mentioned that although he had learned that certain technical writing practices were important, it was assessment that made him implement those practices in assignments. He said, “[t]he practice made me realize it was important but the grades forced me to do it” and elaborated with: “I feel like comments are a part of the grade that you get in 210 ... before that I didn’t really care much about adding comments to my code.”

Some participants explicitly mentioned PrairieLearn [38], an automated assessment tool that is used in many UBC computer science courses. The tool allows manual grading, but is most commonly used as an autograding system where instructors create assessments (that are often randomized) and can programmatically assess submissions. Etson found that most computer science exams focus on writing code and multiple choice questions: “most of the [computer science] course is about the coding and answer the question. My multiple choice on PrairieLearn”. David mentioned the autograder’s focus on code correctness when he said, “you get a hundred percent on PrairieLearn, ... most of the autograder stuff is if the code works, it works and it doesn’t give you any more extra points for documentation” and he found that when working up to a deadline, there was only time to do work that led to grades: “there’s no time to document everything”. He also found that he is doing less writing with the introduction of the automated assessment tool: “now with [PrairieLearn]. So, I’m doing way less writing. Mostly it’s just multiple choice now. I just click, click, click and then I’m done my final exam.”

Participants believe that more emphasis on writing in their courses, with accompanying graded activities, would require students to put more care into their technical writing. For example, Krish said “[w]riting comments or proper documentation is asked for but hardly enforced. If the importance of the task was expressed and represented as part of the grade then maybe it would have more of an impact.” Phyllis wished that professors would spend more class time on writing and include it in their assessments. She said, “I think the added motivation to write well might create better writing skills and standards in the long term.”

Wui discussed a situation in which she lost marks for code quality and commenting; she found that this taught her a good lesson. She said, “they just have five marks allocated for code quality and commenting. And we got a two out of five because we didn’t comment our functions ... I thought the grading was harsh, but it was a good lesson”. With her partner, she tried to improve their code and add comments after losing these marks: “after getting docked off three marks ... me and my partner sat down and just looked at our code for like an hour and tried to make things easy to read and commented things properly.” She realized that the assessment of comments and code quality enforced the importance of technical writing: “I think those practices enforced technical writing.”

7 Discussion

Zhang et al. [19] and von Briesen [20] also found that computer science students believe that it is important to learn technical writing.

Zhang et al. [19] embedded writing instruction and assessment into large, first- and second-year computer science courses at the University of Toronto and collected students' perceptions on their experiences via a survey with 86 first-year and 49 second-year student respondents. Ten of the 86 first-year survey respondents commented that writing documentation is important after taking a course with embedded writing instruction and assessment. I do not believe that the survey explicitly asked about importance, so the lack of comment by other respondents provides no indication of their attitudes. The 10 participants in my study were taking third- and fourth-year courses and all believed that technical writing is important, although they had not taken courses with explicitly embedded writing instruction and assessment. The contexts and participant pools for the two research projects differed, but both were conducted in computer science departments in large, research-intensive Canadian universities. Neither study examined shifts in attitude as students progress through a computer science program, but this could be an interesting avenue of future research. Ten of Zhang et al.'s 135 survey respondents reported that writing helped them better understand both their code and the related computer science concepts that they were learning [19]. These findings are consistent with my findings that students believe technical writing is important and that it can help them understand technical concepts.

von Briesen [20] integrated four writing assignments into an Artificial Intelligence course and conducted pre- and post-term surveys to gather students' perceptions on the importance of writing in computer science. She found that "on average, students agreed that writing in the discipline is important" (p. 119) and that seven of 18 students wished to have further opportunities to write in computer science courses. My study provides further evidence that students want to learn technical writing in computer science courses and also that they are disincentivized from doing the work necessary to achieve this learning when they are primarily assessed and receive feedback via automated tools that do not provide feedback on technical writing.

Feedback is intended to help students learn and should be structured such that it helps the student understand "how to act to close the gap between current and good performance" [39, p. 204]. Much of the feedback that post-secondary students receive is in response to graded assessments. Participants commented that their technical writing was rarely assessed when teaching assistants graded their work and this issue will be compounded by the growing use of automated assessment tools (AATs) in post-secondary computer science courses [21]. These tools do a good job of providing feedback for criteria such as correctness, but current tools do not provide feedback that allows a student to understand how to improve other aspects of their work, such as their technical writing. For technical writing, AATs therefore cannot provide feedback that helps a student assess their current performance and improve their writing towards 'good performance'. Students can over-rely on AATs [28] and spend their effort trying to get their solution to pass all of the autograder tests [22] rather than thinking more holistically about the solution they are designing.

8 Implications

8.1 Implications for Research

Automated assessment tools (AATs) are becoming more commonly used in computer science courses, but implications of their use on student learning and students' experiences needs to be

further explored. Scholars have studied the impact of AATs on some student behaviours, such as submission patterns [23], and some student learning, such as ability to write functionally correct code [22]. However, a holistic study of students' learning has not been conducted. Future research could investigate whether students are achieving the full range of programs' expected learning outcomes, including professional skills such as communication. AATs do not prevent educators from assessing professional skills, but their affordances may influence the skills that educators choose to assess.

The participants in this study were high-achieving students who believed that technical writing is important and generally enjoy writing. Gathering experiences and perspectives from a broader or different set of students might generate different findings. A broader study would provide a richer and more thorough understanding of students' experiences.

8.2 Implications for Practice

Computer science instructors should consider further integrating technical writing and broader communication skills into their course or program learning outcomes and find ways to assess those skills. North American computer science programs are experiencing enrolment booms [40]; large class sizes and limited resources lead to higher adoption of AATs because manual assessment becomes unmanageable [41]. In order to manage the scale of computer science courses and the desire to teach and assess technical writing and communication skills, we need to develop assessment practices or processes that are efficient and meaningful. Findings from this study suggest that assessing technical writing in a meaningful way will encourage students to prioritize it as a learning activity.

Participants expressed a perspective that technical writing was important and stated that they wanted to improve their technical writing skills. However, they also reported that technical writing is not assessed, under assessed, or assessed too easily and that they prioritize aspects of assessments that will be more heavily graded.

When considering assessment in a course or program, a holistic approach to the types and forms of assessment that students receive would provide more thorough feedback on their progress for a larger set of learning goals or program learning objectives. Students would benefit from completing a variety of assessments tailored to meet the full set of learning objectives. These varied assessments could be embedded in courses across the curriculum and include autograded assessments.

Acknowledgements

I am grateful to Drs. Marina Milner-Bolotin, Colleen Lewis, and Ann Anderson for their advice and mentorship throughout this project.

References

[1] W. Groeneveld, H. Jacobs, J. Vennekens, and K. Aerts, "Non-cognitive abilities of exceptional software engineers: A Delphi study," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, February 2020, pp. 1096–1102. [Online]. Available: https://dl.acm.org/doi/10.1145/3328778.3366811

[2] P. L. Li, A. J. Ko, and J. Zhu, "What makes a great software engineer?" in *Proceedings of the IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE Computer Society, May 2015, pp. 700–710. [Online]. Available: https://ieeexplore.ieee.org/document/7194618/

[3] A. Begel and B. Simon, "Struggles of new college graduates in their first software development job," in *Proceedings of the 39th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, March 2008, pp. 226–230. [Online]. Available: https://dl.acm.org/doi/10.1145/1352135.1352218

[4] R. Vivian, K. Falkner, and N. Falkner, "Analysing computer science students' teamwork role adoption in an online self-organised teamwork activity," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*. Association for Computing Machinery, November 2013, pp. 105–114. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2526968.2526980

[5] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1991.

[6] E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1998.

[7] Y. Kafai, C. Proctor, and D. Lui, "From theory bias to theory dialogue: Embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education," in *Proceedings of the 2019 ACM Conference on International Computing Education Research*. Association for Computing Machinery, July 2019, pp. 101–109. [Online]. Available: https://dl.acm.org/doi/10.1145/3291279.3339400

[8] M. Ben-Ari, "Constructivism in computer science education," in *Proceedings of the 29th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, March 1998, pp. 257–261. [Online]. Available: https://dl.acm.org/doi/10.1145/274790.274308

[9] C. Szabo, N. Falkner, A. Petersen, H. Bort, C. Connolly, K. Cunningham, P. Donaldson, A. Hellas, J. Robinson, and J. Sheard, "Review and use of learning theories within computer science education research: Primer for researchers and practitioners," in *Proceedings of the 2019 Working Group Reports on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, July 2019, pp. 89–109. [Online]. Available: https://dl.acm.org/doi/10.1145/3304221.3325534

[10] M. Tight, "Theory application in higher education research: The case of communities of practice," *European Journal of Higher Education*, vol. 5, no. 2, pp. 111–126, April 2015. [Online]. Available: https://doi.org/10.1080/21568235.2014.997266

[11] M. Joy, J. Sinclair, S. Sun, J. Sitthiworachart, and J. López-González, "Categorising computer science education research," *Education and Information Technologies*, vol. 14, no. 2, pp. 105–126, June 2009. [Online]. Available: https://doi.org/10.1007/s10639-008-9078-4

[12] V. Braun, V. Clarke, N. Hayfield, and G. Terry, "Thematic analysis," in *Handbook of Research Methods in Health Social Sciences*, P. Liamputtong, Ed. Springer Singapore, 2019, pp. 843–860.

[13] V. Braun and V. Clarke, "One size fits all? What counts as quality practice in (reflexive) thematic analysis?" *Qualitative Research in Psychology*, vol. 18, no. 3, pp. 328–352, July 2020. [Online]. Available: https://doi.org/10.1080/14780887.2020.1769238

[14] M. Crotty, *The foundations of social research: Meaning and perspective in the research process*. SAGE Publications, Inc., 1998.

[15] K. Moon and D. Blackman, "A guide to understanding social science research for natural scientists," *Conservation Biology*, vol. 28, no. 5, pp. 1167–1177, October 2014. [Online]. Available: https://doi.org/10.1111/cobi.12326

[16] J. Emig, "Writing as a mode of learning," *College Composition and Communication*, vol. 28, no. 2, pp. 122–128, May 1977. [Online]. Available: https://doi.org/10.2307/356095

[17] H. J. Fell, V. K. Proulx, and J. Casey, "Writing across the computer science curriculum," in *Proceeding of the 27th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, March 1996, pp. 204–209. [Online]. Available: https://dl.acm.org/doi/10.1145/236452.236540

[18] O. Hazzan and G. Har-Shai, "Teaching and learning computer science soft skills using soft skills: The students' perspective," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, March 2014, pp. 567–572. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2538862.2538885

[19] L. Zhang, B. Simion, M. Kaler, A. Liaqat, D. Dick, A. Bergen, M. Miljanovic, and A. Petersen, "Embedding and scaling writing instruction across first- and second-year computer science courses," in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education Volume 1*. Association for Computing Machinery, March 2023, pp. 610–616. [Online]. Available: https://doi.org/10.1145/3545945.3569729

[20] E. von Briesen, "Do we need to write? Researching perceptions of disciplinary writing importance and skills in an advanced computer science course," *Journal of Computing Sciences in Colleges*, vol. 39, no. 5, pp. 119–128, November 2023. [Online]. Available: https://dl.acm.org/doi/10.5555/3637068.3637079

[21] J. C. Paiva, J. P. Leal, and A. Figueira, "Automated assessment in computer science education: A state-of-the-art review," *ACM Transactions on Computing Education*, vol. 22, no. 3, June 2022. [Online]. Available: https://doi.org/10.1145/3513140

[22] A. Luxton-Reilly, E. Tempero, N. Arachchilage, A. Chang, P. Denny, A. Fowler, N. Giacaman, I. Kontorovich, D. Lottridge, S. Manoharan, S. Sindhwani, P. Singh, U. Speidel, S. Stephen, V. Terragni, J. Whalley, B. Wuensche, and X. Ye, "Automated assessment: Experiences from the trenches," in *Proceedings of the 25th Australasian Computing Education Conference*. Association for Computing Machinery, January 2023, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3576123.3576124

[23] J. Leinonen, P. Denny, and J. Whalley, "A comparison of immediate and scheduled feedback in introductory programming projects," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education Volume 1*. Association for Computing Machinery, February 2022, pp. 885–891. [Online]. Available: https://doi.org/10.1145/3478431.3499372

[24] K. M. Ala-Mutka, "A survey of automated assessment approaches for programming assignments," *Computer Science Education*, vol. 15, no. 2, pp. 83–102, February 2005. [Online]. Available: https://doi.org/10.1080/08993400500150747

[25] J. Mitra, "Studying the impact of auto-graders giving immediate feedback in programming assignments," in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education Volume 1*. Association for Computing Machinery, March 2023, pp. 388–394. [Online]. Available: https://doi.org/10.1145/3545945.3569726

[26] L. Zamprogno, R. Holmes, and E. Baniassad, "Nudging student learning strategies using formative feedback in automatically graded assessments," in *Proceedings of the 2020 ACM SIGPLAN Symposium on SPLASH-E*. Association for Computing Machinery, November 2020, pp. 1–11. [Online]. Available: https://doi.org/10.1145/3426431.3428654

[27] D. M. Rao, "Experiences with auto-grading in a systems course," in *Proceedings of the 2019 IEEE Frontiers in Education Conference*. IEEE Computer Society, October 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/FIE43999.2019.9028450

[28] E. Baniassad, L. Zamprogno, B. Hall, and R. Holmes, "STOP THE (AUTOGRADER) INSANITY: Regression penalties to deter autograder overreliance," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, March 2021, pp. 1062–1068. [Online]. Available: https://doi.org/10.1145/3408877.3432430

[29] L. Saliba, E. Shioji, E. Oliveira, S. Cohney, and J. Qi, "Learning with style: Improving student code-style through better automated feedback," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education Volume 1*. Association for Computing Machinery, March 2024, pp. 1175–1181. [Online]. Available: https://doi.org/10.1145/3626252.3630889

[30] G. Nagakalyani, S. Chaudhary, V. Apte, and G. Ramakrishnan, "TA buddy: AI-assisted grading tool for introductory programming assignments," in *Proceedings of the 56th ACM Technical Symposium on Computer Science Education Volume 2*. Association for Computing Machinery, February 2025, p. 1736. [Online]. Available: https://doi.org/10.1145/3641555.3705051

[31] B. Kimmel, A. L. Geisert, L. Yaro, B. Gipson, R. T. Hotchkiss, S. K. Osae-Asante, H. Vaught, G. Wininger, and C. Yamaguchi, "Enhancing programming error messages in real time with generative AI," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, May 2024. [Online]. Available: https://doi.org/10.1145/3613905.3647967

[32] B. Pereira Cipriano, M. Silva, R. Correia, and P. Alves, "Towards the integration of large language models and automatic assessment tools: Enhancing student support in programming assignments," in *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*. Association for Computing Machinery, November 2024. [Online]. Available: https://doi.org/10.1145/3699538.3699588

[33] N. Bergen and R. Labonté, ""Everything Is Perfect, and We Have No Problems": Detecting and limiting social desirability bias in qualitative research," *Qualitative Health Research*, vol. 30, no. 5, pp. 783–792, 2020. [Online]. Available: https://doi.org/10.1177/1049732319889354

[34] Lumivero, "NVivo Transcription." Lumivero, 2024, retrieved from: https://lumivero.com/products/nvivo-transcription/.

[35] M. Allen, "An in-depth exploration of student experiences and perspectives of learning technical writing in undergraduate computer science education," Ph.D. dissertation, University of British Columbia, 2024. [Online]. Available: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0445050

[36] Y. S. Lincoln and E. G. Guba, *Naturalistic Inquiry*. SAGE Publications, Inc., 1985.

[37] S. Brinkmann, "The interview," in *The SAGE Handbook of Qualitative Research*, 5th ed., N. K. Denzin and Y. S. Lincoln, Eds. SAGE Publications, Inc., 2018, pp. 576–599.

[38] PrairieLearn, Inc., "PrairieLearn," 2024, retrieved from: https://www.prairielearn.com on January 9, 2024.

[39] D. J. Nicol and D. Macfarlane-Dick, "Formative assessment and self-regulated learning: A model and seven principles of good feedback practice," *Studies in Higher Education*, vol. 31, no. 2, pp. 199–218, 2006. [Online]. Available: https://doi.org/10.1080/03075070600572090

[40] J. L. Tims, C. Tucker, M. A. Weiss, and S. Zweben, "Computing enrollment and retention: Results from the 2021–22 undergraduate enrollment cohort," *ACM Inroads*, vol. 14, no. 4, pp. 24–43, December 2023. [Online]. Available: https://doi.org/10.1145/3629980

[41] D. M. Souza, K. R. Felizardo, and E. F. Barbosa, "A systematic literature review of assessment tools for programming assignments," in *Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training*. IEEE Computer Society, April 2016, pp. 147–156. [Online]. Available: http://ieeexplore.ieee.org/document/7474479/