An In-Depth Exploration of Student Experiences and Perspectives of Learning Technical Writing in Undergraduate Computer Science Education

by

Meghan Elizabeth Allen

M. Sc., The University of British Columbia, 2006

B. Sc., The University of British Columbia, 2001

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Curriculum Studies)

The University of British Columbia
(Vancouver)

August 2024

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

An In-Depth Exploration of Student Experiences and Perspectives of Learning Technical Writing in Undergraduate Computer Science Education

submitted by **Meghan Elizabeth Allen** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Curriculum Studies**.

Examining Committee:

Marina Milner-Bolotin, Professor, Curriculum and Pedagogy, The University of British Columbia
Supervisor

Ann Anderson, Professor Emerita, Curriculum and Pedagogy, The University of British Columbia
Supervisory Committee Member

Colleen Lewis, Associate Professor, Computer Science, University of Illinois Urbana-Champaign
Supervisory Committee Member

Cynthia Nicol, Professor, Curriculum and Pedagogy, The University of British Columbia
University Examiner

Jude Walker, Associate Professor, Educational Studies, The University of British Columbia
University Examiner

Kristin Searle, Associate Professor, Instructional Technology and Learning Sciences, Utah State University
External Examiner

Abstract

While communication skills, including technical writing skills, are important for professional computer scientists' career success, they may not be explicitly addressed within undergraduate computer science education and there is dearth of scholarship exploring students' experiences of learning these skills. Understanding how technical writing skills are learned in post-secondary computer science programs could inform better curricular and pedagogical approaches for teaching these skills, thereby equipping students with competencies necessary for their future careers. More specifically, there is a need for research on undergraduate computer science students' perspectives of what technical writing skills they are learning, how they are learning them, and how they experience this learning.

This study was designed to answer three research questions:

- RQ1: How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?
- RQ2: In what ways do students hold perspectives that may be barriers to their learning of technical writing?
- RQ3: How do course assessment practices affect students' learning of technical writing?

To answer these research questions, I conducted an interpretive qualitative study, grounded in Lave and Wenger's Situated Learning Theory and Social Theory of Learning. Ten third- and fourth-year computer science students participated. I collected data via reflective journal writing and beginning-of-term and end-of-term interviews to gather a deep view into the student experience. I followed Braun and

Clarke's method of reflexive thematic analysis to generate themes from the data corpus.

I found that computer science students' communities of practice influence their learning of technical writing in many and varied ways, that students hold nuanced perspectives about technical writing in computer science, and that course assessment practices affect the learning activities that students prioritize.

Lay Summary

Communication skills, including technical writing skills, are important for professional computer scientists' career success. This research study examined computer science students' experiences learning technical writing in their computer science courses at the University of British Columbia, with the goal of expanding knowledge about students' experiences and improving post-secondary computer science programs.

I interviewed ten third- and fourth-year computer science students and sent them weekly reflective writing prompts. The interview questions and writing prompts were designed to encourage participants to discuss how their academic or work communities influence their learning of technical writing. I transcribed the interviews and then analyzed the data to generate themes.

I found that students' experiences of learning technical writing are influenced by their communities in various and complex ways, that students hold nuanced perspectives about technical writing in computer science, and that course assessment practices affect the learning activities that students prioritize.

Preface

I, Meghan Allen, was the lead investigator in this project. I identified the research questions, designed and conducted the research study, and analyzed the data.

This research study was approved by the UBC Behavioural Ethics Board before it commenced. The approval certificate number is H22-02114 and the project title is Exploring Computer Science Students' Attitudes, Beliefs, and Experiences Related to Learning Technical Writing.

Preliminary results were presented in a five-minute lightning talk at the 54th ACM Technical Symposium on Computer Science Education. An abstract for the lightning talk has been published.

Allen, M. (2023). Computer Science Students' Experiences of Learning Technical Writing. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education Volume 2*, page 1267. Association for Computing Machinery.

The results, implications, and discussion related to Research Question 3, along with relevant parts of the Introduction, Literature Review, and Research Design chapters have been submitted for publication. I intend to publish the remainder of the findings but have not submitted for publication yet.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAT Automated Assessment Tool

ACM Association for Computing Machinery

API Application Programming Interface

CAC Communication Across the Curriculum

CLASS Colorado Learning Attitudes about Science Survey

ELL English Language Learner

FAQ Frequently Asked Question

HCI Human-Computer Interaction

IEEE Institute of Electrical and Electronics Engineers

STEM Science, Technology, Engineering, and Mathematics

UBC University of British Columbia

WAC Writing Across the Curriculum

WID Writing in the Discipline

Glossary

API Application Programming Interface: a software interface, allowing "two or more computer programs or components to communicate with each other" (Wikipedia, 2024a)

assembly language "any low-level programming language with a very strong correspondence between the instructions in the language and the architecture's machine code instructions" (Wikipedia, 2024b)

backend "the data and infrastructure that make your application work. It stores and processes application data for your users" (Amazon Web Services, 2024)

bug reports Software developers and users document problems in computer programs via bug reports that are stored in an issue tracking system

code review "a software quality assurance activity in which one or more people check a program, mainly by viewing and reading parts of its source code" (Wikipedia, 2024c)

commit message When a developer saves a copy of their code in a version control system, it's called a commit. Each time someone commits code, they can write a short message. Development teams usually follow conventions for the content of the commit messages

developer notes Software developers use a variety of written formats to collaborate with each other and document their work

empty catch blocks A block of code that is intended to handle an exceptional situation, but is instead empty

feature requests Software developers and users document software features that they would like to be implemented in an issue tracking system

frontend "what your users see and includes visual elements like buttons, checkboxes, graphics, and text messages. It allows your users to interact with your application" (Amazon Web Services, 2024)

GitHub "GitHub is a developer platform that allows developers to create, store, manage and share their code" (GitHub, Inc., 2024)

issue tracker A system that tracks bug reports and feature requests for a software development project

Java documentation Documentation describing the Java programming language and its API

manual pages Documentation pages found on Unix and Unix-like operating systems

pull request "[A] proposal to merge a set of changes from one branch into another. In a pull request, collaborators can review and discuss the proposed set of changes before they integrate the changes into the main codebase. Pull requests display the differences, or diffs, between the content in the source branch and the content in the target branch" (GitHub, Inc., 2024)

unused variable A variable in a computer program that never has a value assigned to it, nor is accessed

Acknowledgments

This dissertation could not have been completed without the support of many members of my community.

To the study participants; thank you for your generosity in sharing your time, wisdom, and experiences with me.

To my supervisory committee, Drs. Marina Milner-Bolotin, Ann Anderson, and Colleen Lewis. Thank you for your unwavering support and your thoughtful questions and guidance. I have learned so much from all of you and appreciate your mentorship.

To Dr. Ashley Welsh; your encouragement and critical friendship has made this work better. Thank you for all the conversations.

To Dr. Kari George; your suggestions and coaching were key to my ability to actually finish this dissertation. Thank you.

To the UBC Department of Computer Science, and particularly the recent department heads Drs. Chen Greif, Ian Mitchell, Joanna McGrenere, and Margo Seltzer; thank you for graciously supporting my pursuit of a PhD.

To my family; Mom, Dad, and Greg, thank you for teaching me I can do hard things and for providing strong examples for me. Dave, Kyle, and Brandon, thank you for your support and the many, many laughs.

Chapter 1

Introduction

While communication skills, including technical writing skills, are important for professional computer scientists' career success (Begel and Simon, 2008; Exter et al., 2018; Galster et al., 2022; Groeneveld et al., 2020; Li et al., 2015; Scaffidi, 2018), they may not be explicitly addressed within undergraduate computer science education. Understanding how communication skills are learned in post-secondary computer science programs could inform better curricular and pedagogical approaches for teaching communication skills, thereby equipping students with competencies necessary for their future careers. The range of communication skills that computer scientists need to use is broad and includes writing, reading, presenting, and listening. These skills are taught to computer science students in varying degrees but little is known about students' experiences of learning these skills in post-secondary computer science programs. To narrow this research study, I chose to focus on technical writing as one aspect of communication that is critical to computer scientists' professional practice (Vivian et al., 2013).

This study broadly investigates computer science students' perceptions of learning technical writing. My specific goals are to understand the nuances of students' experiences learning technical writing with others, learn about students' perspectives that may create barriers to their learning of technical writing, and investigate how students' technical writing practices and beliefs are influenced by course assessment practices.

In this chapter, I introduce my research study. I describe the context and motivation for this work, the research questions and goals, the contributions and significance of this study, and finally, the structure of this dissertation.

1.1 Motivation

Communication skills encompass a wide variety of skills including, but not limited to, writing, reading and understanding written documents, giving oral presentations, and active listening. These communication skills are acquired in a multitude of contexts, such as familial and community social interactions, formal schooling experiences, and work or volunteer situations. Communication is an inherently social activity; an individual's communication style is related to their personal history and past experiences as well as any formal education they have received.

Communication skills are important to employers in the computing industry (Galster et al., 2022; Scaffidi, 2018) and are critical to employees' success (Exter et al., 2018; Vivian et al., 2013), but the perception of those in industry is that students are lacking in communication skills (Begel and Simon, 2008; Iniesto et al., 2021; Radermacher and Walia, 2013; Vivian et al., 2016). Specifically, written communication is under-emphasized in undergraduate curricula (Exter et al., 2018) and commonly reported as a deficit in new graduates who have recently joined the workforce (Begel and Simon, 2008; Lethbridge, 1998; Radermacher and Walia, 2013). While not all computer science students aim to work in these industry environments, many do. Therefore, these findings provide motivation for post-secondary computer science programs to teach written and other communication skills.

In addition to being crucial to workplace success, communication skills have become more prominent in recommended undergraduate computer science curricula. Two leading computer science professional organizations, The Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE), have been developing recommended undergraduate computing curricula approximately every 10 years since 1968. The computer science curricula they developed in 2001, 2008, 2013, and 2023 include professional practice as an integral component of the undergraduate curriculum, including, but not limited to, written and oral communication (CS2008 Review Joint Task Force, 2008;

Joint Task Force on Computing Curricula, 2001, 2013, 2024). The 2023 version lists effective oral and written communication skills as core skills required of a computer science graduate (Joint Task Force on Computing Curricula, 2024). Specific technical writing skills that are listed include: “[u]nderstand the importance of writing concise and accurate technical documents following well-defined standards for format and for including appropriate tables, figures, and reference” (Joint Task Force on Computing Curricula, 2024, p. 207) and “[a]nalyze written technical documentation for technical accuracy, concision, lack of ambiguity, and awareness of audience” (Joint Task Force on Computing Curricula, 2024, p. 207). Further, the 2023 recommended curricula states that “[c]ommunication skills are best taught in context throughout the undergraduate curriculum” (Joint Task Force on Computing Curricula, 2024, p. 207). These curricular recommendations are provided as guidance for computer science departments to use when they design or update their programs and courses.

I am a faculty member in computer science at the University of British Columbia (UBC) and am motivated to learn about UBC computer science students’ experiences learning technical writing. A UBC computer science curriculum mapping project was conducted in 2019 with the goal of generating a list of current program learning outcomes that describe “significant and essential learning that all students in a[n undergraduate computer science] program have had a good opportunity to achieve” (UBC Computer Science Department, 2019). None of the identified program outcomes describe writing or communication skills, which indicates a gap in our program’s curriculum in comparison to the curriculum recommended by the ACM and IEEE. A better understanding of what students learn about technical writing in our courses will allow us to more accurately specify program learning outcomes and more clearly identify gaps in our curriculum. In the future, we could design aspirational program learning outcomes that are informed by industrial needs, recommended curricula, and our students’ experiences, and refine our department’s curriculum to meet these outcomes. The findings of this study may transfer to other departments’ contexts and allow others to refine their curricula.

1.2 Context

I define technical writing as "writing to convey technical content". This definition is intentionally broad in order to capture any writing task that a participant may have undertaken. Examples of technical writing that fit this definition include commit messages, bug reports, feature requests, emails about technical issues, reports, developer notes, user documentation, and other technical documents. Prior to sharing my definition with participants, I asked each of them to define technical writing and to give examples. Since my intent was to learn about their perspectives and experiences, I honoured all definitions and examples of technical writing that participants provided. I share their definitions and examples in Section 4.2.

Scholars have been integrating technical writing into computer science curricula since at least the 1970s (Hoffman et al., 2006), but it remains a relatively understudied area of the curriculum. I provide a brief overview of prior work as context for my research goals and questions; more details are shared in Chapter 2.

Most of the literature on technical writing in computer science education describes how an institution teaches writing. Some institutions integrate writing skills across courses, often basing their curriculum design on a framework for integrating writing into other disciplinary content. For example, Writing Across the Curriculum (WAC) (Falkner and Falkner, 2012; Fell et al., 1996; Kay, 1998; Zhang et al., 2023), Communication Across the Curriculum (CAC) (Gruba and Søndergaard, 2001; Karatsolis et al., 2011), or Writing in the Discipline (WID) (Kortsarts et al., 2010; Zhang et al., 2023) are common initiatives that guide technical writing projects in computer science education. Other institutions teach writing and communication skills via a single course (Etlinger, 2006; Hazzan and Har-Shai, 2013, 2014; Kaczmarczyk, 2003; Kay, 1998; Michael, 2000; Pollock, 2001).

Students' beliefs and attitudes towards an academic discipline influence their learning of the discipline (Lewis et al., 2010; Perkins et al., 2005). To the best of my knowledge, limited research has investigated computer science students' perspectives of what technical writing skills they are learning, how they are learning these skills, and how they experience this learning. The exceptions (Cilliers, 2012; Hazzan and Har-Shai, 2014; Munir et al., 2022) that I have found will be discussed in Chapter 2. The dearth of research on computer science students' experiences of learning communication skills provides an opportunity to better understand curricular and pedagogical gaps and further understand how students' experiences and perspectives influence their learning of technical writing.

1.3 Research Goals and Questions

To begin to address the research gap, this study aims to generate a nuanced understanding of students' experiences with the curriculum regarding written technical communication.

Broadly, I investigated UBC upper-year undergraduate computer science students' experiences learning about technical writing in computer science courses and in their workplaces, their reflections on their experiences, and their perspectives towards technical writing.

This study was guided by three research questions:

- RQ1: How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?
- RQ2: In what ways do students hold perspectives that may be barriers to their learning of technical writing?
- RQ3: How do course assessment practices affect students' learning of technical writing?

My goal is to answer these research questions and use the findings to suggest implications for practice and further research.

1.4 Contributions

I conducted an interpretive qualitative study that is grounded in Lave and Wenger's (1991) Situated Learning Theory and Wenger's (1998b) Social Theory of Learning and uses Braun et al.'s (2019) method of reflexive thematic analysis. To the best of my knowledge, the use of reflexive thematic analysis is uncommon in computer science education research, so this study may provide methodological foundations for future research.

This study builds on previous projects designed to teach communication skills to computer science students (e.g., Etlinger, 2006; Falkner and Falkner, 2012; Fell et al., 1996; Gruba and Søndergaard, 2001; Hazzan and Har-Shai, 2013, 2014; Karatsolis et al., 2011; Kay, 1998; Kortsarts et al., 2010; Munir et al., 2022; Pollock, 2001; Zhang et al., 2023), but with a focus on student perspectives and experiences. To the best of my knowledge, few scholars have investigated students' perspectives on learning technical writing and my study begins to fill the knowledge gap by providing a nuanced understanding of students' experiences with the curriculum regarding written technical communication.

This study will provide a rich understanding of students' experiences of learning technical writing with others, students' perspectives about technical writing, and how course assessment practices affect students' learning of technical writing. The themes generated via reflexive thematic analysis will expand knowledge about students' perceptions of their technical writing experiences in computer science courses and their perspectives related to learning about technical writing. The results may provide a foundation for future research on students' experiences of learning technical writing, communication skills more generally, or other skills in their computer science programs. The results may inform UBC computer science program curricula updates and assessment practices. More broadly, the results will be published and might impact computer science pedagogy and curriculum related to technical writing outside of UBC.

1.5 Structure of this Dissertation

The remainder of this dissertation is structured as follows. Chapter 2 provides foundations and context for this study via a literature review. Chapter 3 describes the study's research design. Chapter 4 shares results for each of the research questions. Chapter 4 is separated into six sections. The first gives an overview of the results, the second describes participants' definitions of technical writing, and the third shares participants' reflections on their own technical writing practice. The remaining three sections discuss each of the three research questions. Each of latter three sections contains the results, discussion, and implications related to the corresponding research question. The discussions and implications are each relevant to a particular research question, so I placed them immediately following the research question's results rather than in standalone chapters. Chapter 5 summarizes the study and my contributions.

Chapter 2

Literature Review

In this chapter, I describe the literature that informed this work and situate this study in its related work. This chapter is separated into four sections. The first section discusses the theoretical perspective that I brought to this research. The second section presents the work related to teaching and learning of technical writing in computer science education. The third section shares the literature related to automated assessment tools in computer science education. The automated assessment literature is relevant because the study investigates how assessment practices affect students' learning of technical writing. Finally, in the fourth section, I situate this study in the related work.

2.1 Theoretical Perspective

Computer science students, like all students, develop written communication skills through school-based, work-based, or other experiences. To understand these experiences, it is essential to analyze them through a theoretical framework. There is not a single and correct theoretical perspective through which to understand these student experiences, rather there are a variety of lenses that could be taken, each of which would provide a "partial and contingent" (Kafai et al., 2019, p. 101) view.

This research is grounded in Lave and Wenger's (1991) Situated Learning Theory and Wenger's (1998b) Social Theory of Learning. Lave and Wenger (2002) state that "[a]ll theories of learning are based on fundamental assumptions about the person, the world, and their relations" (p. 143). In order to contextualize their Situated Learning Theory, which I will describe in detail in Section 2.1.3, I must first acknowledge and describe how someone's belief system fundamentally influences their engagement with learning theories and their approach to research. A researcher's fundamental belief about knowledge, or their epistemology, informs the theoretical perspective (including, in this case, theories of learning) that they bring to their work, which then informs their research methodology, and, in turn, the methods they use (Crotty, 1998). These aspects must be aligned and consistent in order to design a sound research project (Crotty, 1998).

2.1.1 Epistemology

An epistemology provides "philosophical grounding for deciding what kinds of knowledge are possible and how we can ensure that they are both adequate and legitimate" (Maynard, 1994, p. 10). There are various epistemologies and each leads to a different belief about what is knowable and how to discover what is knowable (Crotty, 1998). A researcher's beliefs about the nature of knowledge drive their theoretical perspective, i.e., their philosophical stance providing context for their research methodology. For any given epistemology, there may be multiple consistent theoretical perspectives that a researcher may hold. However, there would also be theoretical perspectives that are inconsistent with the researcher's epistemology, and therefore would not lead to a consistent research design. A researcher's theoretical perspective then informs the methodology, or the high-level strategy or design for the research project (Crotty, 1998). Finally, the methodology informs the research methods, or the data collection and analysis techniques. As an epistemology is a foundational belief, it is "embedded in the theoretical perspective and thereby in the methodology" (Crotty, 1998, p. 3).

Post-secondary computer science education researchers hold diverse epistemological and theoretical perspectives (Ben-Ari, 1998; Kafai et al., 2019; Szabo et al., 2019), although, as with much higher education research, these fundamental perspectives are often unexpressed in published research articles (Joy et al., 2009; Tight, 2015). Epistemology and theoretical perspective, even when unacknowledged, affect researchers' tacit beliefs, assumptions, and underlie their theories of learning. Further, as Kafai et al. (2019) argue, computer science education researchers' beliefs and perspectives influence their understandings and interpretations of the teaching and learning phenomena of interest. Fincher and Petre (2004) state that these tacit assumptions underpin and constrain "what *kinds* of questions can be asked, what are *legitimate* questions, what are *appropriate* questions, and even what questions are *allowable*" (p. 16) but even more broadly, the researcher's belief system affects how they choose to investigate those questions and how they interpret the findings (Crotty, 1998).

As discussed above, theoretical perspectives and their concomitant research methodologies and methods provide "partial and contingent" (Kafai et al., 2019, p. 101) knowledge about phenomena under study. There is not a correct or unifying theoretical perspective and studying the same phenomena through various lenses may prove valuable (Moon and Blackman, 2014). Kafai et al. (2019) suggest that researchers foreground their perspectives as part of the context surrounding their work. This context is necessary for readers to be able to understand the "conceptual framing" (Fincher and Petre, 2004, p. 16) and provides necessary context to consider the validity of the findings.

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" (Crotty, 1998, p. 42). Therefore, I do not believe that there is an objective truth in the world waiting to be discovered (an objectivist view), nor do I believe that reality entirely depends on an individual's perceptions (a subjectivist view) (Moon and Blackman, 2014). Since I believe that knowledge is constructed via interactions of humans 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. It follows, then, that 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.

2.1.2 Theoretical Perspective

The theoretical perspective that I bring to my work is that of 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" (Moon and Blackman, 2014, 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. Through this research, I aim to uncover these various understandings rather than attempting to find a single understanding.

I bring my interpretivist lens to this qualitative research study that uses reflexive thematic analysis (Braun and Clarke, 2006). I describe the methodology, data collection methods, and analysis technique in more detail in Chapter 3.

2.1.3 Situated Learning Theory and the Social Theory of Learning

Lave and Wenger (1991) theorize that learning is "an integral and inseparable aspect of social practice" (p. 31) and must be considered in the sociocultural and historical context in which it occurs. Situated learning views learning as an activity that occurs in a *community of practice* via *legitimate peripheral participation* in contrast to a traditional view that distinguishes the cognitive aspects of an individuals' learning processes as distinct from everyday activity (Lave, 2009). I will define the terms community of practice and legitimate peripheral participation later in this section.

Lave (2009) defines learning as "changes in knowledge and action" (p. 201) and states that situated activity, or activity as considered in a particular context, always involves such changes. Situated learning, then, "emphasizes the relational interdependency of agent and world, activity, meaning, cognition, learning, and knowing" (Lave, 1991, p. 67); an individual's learning must be considered in its context and in relation to others. The interdependency between a learner and their context leads to learning for all individuals involved in the context. Traditionally, learning was studied as if it was a process solely contained in the learner's mind and the lived-in world was ignored (Lave, 2009). The Situated Learning Theory brings the lived-in world to the forefront.

Wenger (2009) expands on the Situated Learning Theory with his Social Theory of Learning. He bases his theory on four premises:

> [w]e are social beings;
> [k]nowledge is a matter of competence with respect to valued enterprise;
> [k]nowing is a matter of participating in the pursuit of such enterprises, that is, of active engagement in the world;
> [m]eaning – our ability to experience the world and our engagement with it as meaningful – is ultimately what learning is to produce (p. 210)

The Social Theory of Learning maintains the tenets of Situated Learning, namely the belief that learning occurs through legitimate peripheral participation in communities of practice. However, the Social Theory of Learning stresses the identities of both learner and 'expert' that are formed in relation to and with communities. While Lave and Wenger (1991) do discuss identity in their book and state that "learning involves the whole person" (p. 53), both Wenger and Lave place considerably more emphasis on identity in their later work (Lave, 2009; Wenger, 1998b). I believe that considering the learner's identity in relation to their community or communities of practice is essential because someone's subjectivity and history influence their beliefs and their understanding of the world and the activities they partake in.

Although Wenger's (2009) Social Theory of Learning focuses on learning that occurs via legitimate peripheral participation in communities of practice, when he writes, "[i]n spite of curriculum, discipline, and exhortation, the learning that is the most personally transformative turns out to be the learning that involves membership in these communities of practice" (p. 212), he acknowledges that not all learning occurs in these settings. While he contends that situated learning is particularly transformative for the learner, he acknowledges that learning can also take place in alternative contexts.

Communities of Practice

Lave and Wenger (1991) define a community of practice as "a set of relations among persons, activity, and world, over time and in relation with other tangential and overlapping communities of practice" (p. 98). This definition emphasizes relations amongst people but also with their activities and the world. A community is influenced by (and influences) the social, cultural, and historical forces of the world in which it's embedded. Over time, the community itself will change through its members' personal growth and as membership in the community renews; this change and parallel changes in other communities will cause the community's relationships with tangential, overlapping communities to be fluid and dynamic.

Wenger (1998a) states that a community of practice defines itself via its "joint enterprise as understood and continually renegotiated by its members" (p. 2), mutual engagement, and "shared repertoire of communal resources (routines, sensibilities, artifacts, vocabulary, styles, etc.)" (p. 2). A key notion of a community of practice is that it defines itself. Communities of practice do not need to be tightly connected to organizational structures within groups or institutions.

Lave and Wenger (1991) use the term newcomers to represent learners who are new to the community and the term oldtimers to represent longstanding community members (who are, of course, continuously learning). Engagement with a functioning community provides newcomers the opportunity to learn; as Lave and Wenger (1991) state, "the practice of the community creates the potential "curriculum" in the broadest sense – that which may be learned by newcomers with legitimate peripheral access" (p. 93). The community's curriculum is not explicit; it exists in the everyday practice of community members. Community oldtimers do not (necessarily) engage in explicit teaching, but newcomers learn from them nonetheless. In fact, Lave and Wenger (1991) state that oldtimers conferring legitimacy to newcomers is more important than teaching.

Lave and Wenger (1991) also discuss the learning that occurs amongst peers and near-peers in a community of practice. When peers and near-peers can circulate knowledge amongst themselves, Lave and Wenger (1991) claim that the spread of knowledge is particularly fast and effective. This supports the notion that learning occurs amongst all community members and is not transmitted solely from oldtimers to newcomers.

Community members "are engaged in the generative process of producing their own future" (Lave and Wenger, 1991, p. 57). The practice of the community is not static; instead, it evolves as the membership and individual member's practice and knowledge evolve. Communities are simultaneously reproducing and transforming their practice (Lave, 1991). These do not have to be competing goals. The community has valuable knowledge and practices that must be learned and reproduced. Over time, naturally, some community members leave the community and others take more central roles in the community. The natural process of evolving membership and the dynamic practices of community members as they learn together lead to transformation of the community's practice. Guile and Young (1999) argue that a purely transmissive and reproductive model assumes that the community's knowledge is "always unproblematic and clearly understood" (p. 153). New knowledge, or new practices, must be generated as a community evolves (Guile and Young, 1999).

Communities of practice are both fluid and overlapping; an individual may belong to many communities, including those for their work, hobbies, and personal lives (Masika and Jones, 2016). Each community has its own "practices, rituals, conventions and histories" (Masika and Jones, 2016, p.141) and the fluidity of individuals' membership allows practices to move between communities. Wenger (1998b) discusses the benefits of this cross-community learning when he considers reconciling membership in multiple communities of practice:

> the work of reconciliation can be integrated in the community's enterprise and thus, to some extent, become part of a shared learning practice. Such communities will not only gain the allegiance of their members, they will also enrich their own practices (p. 216)

Cross-community learning could lead to transformative change in one or multiple communities.

Legitimate Peripheral Participation

Legitimate peripheral participation describes the process of becoming a "full participant in a sociocultural practice" (Lave and Wenger, 1991, p. 29) as an individual becomes more knowledgeable and skilled and moves from peripheral participation in the community's activities to central participation. A newcomer's participation will initially be peripheral to the main contributions of the community, but, over time, as they learn from oldtimers, their work becomes more central and they become more expert. Lave and Wenger (1991) see ongoing participation as involving "negotiation and renegotiation of meaning in the world" (p. 51), which "implies that understanding and experience are in constant interaction – indeed, are mutually constitutive" (pp. 51-52). The concept of legitimate peripheral participation "makes a fundamental distinction between learning and intentional instruction" (Lave and Wenger, 1991, p. 40). Newcomers learn the ways of the community via their participation in the community.

Identity

Wenger (1998b) states that "[b]ecause learning transforms who we are and what we can do, it is an experience of identity" (p. 215). We cannot consider situated learning without considering learner's identities and how identity both affects and is affected by interactions amongst community members. Wenger (1998b) describes how we, as individuals, learn about our effects on the world and how the world treats us through participation in action. Our identity influences how we affect the world, as our identity influences our power relations with other people and institutions. Identity also influences the way in which we are treated and the way we interpret that treatment. He describes how our identities and individual boundaries within a community are set and reset through interactions in community:

> boundaries are experienced very participatively but concretely through our inability to engage fully in an activity, participate in a conversation, perform a demanding task, notice a subtle cue, or respond to an unspoken expectation. Boundaries are met in that brief look of disapprobation or incomprehension when we have just put our foot in our mouth, the sensation of which can then linger as an indelible part of our identity. (Wenger, 1998b, p. 193)

These interactions then lead back into our identities and how we perceive ourselves or our actions as accepted or unaccepted in a community. This identity formation and transformation may not be obvious; as individuals focus on participating in community they may not be aware of the simultaneous changes to their identity (Wenger, 1998b).

Wenger (1998b) further describes learning as a “process of becoming” (p. 215) and states that we learn “in the service of an identity” (p. 215) rather than learning being the end goal itself. He explains that people learn in order to be a particular kind of person, for example, a member of a profession, or to be able to do particular things. Wenger (1998b) discusses how an identity has both a past and a future; a community should enable identity development by allowing individuals’ identities and histories to contribute to the community’s practice and by allowing an individual opportunities to participate that provide a path to a valued future.

Individuals perceive and understand events differently, depending on their identities and histories. Lave (2009) writes that “people in the same situation, people who are helping to constitute “a situation” together, know different things and speak with different interests and experience from different social locations” (p. 206). Therefore, we must attend to individual participant’s interpretations of their experiences rather than attempting to amalgamate a single understanding of learning in a community. Masika and Jones (2016) write “[i]ndividuals bring to a community, personal histories of involvement with other social groups whose norms may complement or conflict with one another” (p. 145); these norms and personal histories could enrich the community, but they could instead marginalize some community members.

Identity is an “integral aspect of a social theory of learning” (Wenger, 1998b, p. 145) and is central to my understanding of the theory.

Situated Learning Theory in Computer Science Education

Situated Learning Theory has been used as an underlying theory of computer science education research in various contexts dating back to at least 2006. Multiple literature reviews examining papers published in computer science education venues identified Situated Learning Theory (Malmi et al., 2020a, 2014, 2010; Szabo et al., 2019; Szabo and Sheard, 2022) as a theoretical foundation for research projects, although these reviews do not provide a precise list of the papers they considered.

Communities of practice and legitimate peripheral participation have been used as a framework for investigating students’ identities (Blunt and Pearson, 2021;

Große-Bölting et al., 2021; Kapoor and Gardner-McCune, 2018, 2019; Weidler-Lewis et al., 2017), as context for a study of a small online community that transformed from a gaming league to a team that developed games (Dym et al., 2023), to inform a partnership project that brought industry professionals and faculty members together to create and teach a course related to the industry professional's expertise (Tenenberg, 2010), to explore teaching in a non-majors context where the "alignment of a community of practice does not yet exist" (Guzdial and Tew, 2006, p. 51), and in a phenomenographic study to explore CS and IT Students' experiences participating in the discipline (Peters et al., 2015). As in my study, many of these research projects focus on students' experiences or on professional practice. Further, scholars have created communities of practice for computer science education researchers (Fincher and Tenenberg, 2006) and teachers (Cooper et al., 2015; Decker et al., 2024; Sentance and Humphreys, 2018) to enable learning from and with similar-minded colleagues.

How Situated Learning Theory and the Social Theory of Learning Apply to this Project

Situated Learning (Lave and Wenger, 1991) and the Social Theory of Learning (Wenger, 2009) stem from Lave and Wenger's consideration of an apprenticeship[1] model of learning. Although most undergraduate education does not mimic traditional apprenticeship, these theories are relevant to this study because communication skills are learned and practiced in social contexts.

Given the social situatedness of communication and technical writing more specifically, a constructivist and interpretivist approach allows me to attend to individuals' interpretations and perceptions. As Lave (1991) states, "[i]n the interpretive view, meaning is negotiated, the use of language is a social activity rather than a matter of individual transmission of information, and situated cognition is

[1] Lave and Wenger (1991) give tailoring as an example of apprenticeship, where apprentices "engage in a common, structured pattern of learning experiences without being taught, examined, or reduced to mechanical copiers of everyday tailoring tasks" (p. 30) and become "skilled and respected master tailors" (p. 30). They acknowledge multiple forms of apprenticeship, including "cognitive apprenticeship, apprenticeship learning, and even life as apprenticeship" (p. 29) and state that it is "evident that no one was certain what the term meant" (p. 29). The various definitions of the word apprenticeship led them to more fully consider how people learn via legitimate peripheral participation in communities of practice and define Situated Learning Theory.

always interest-relative" (p. 66). An interpretive perspective in the context of Situated Learning Theory is a suitable and consistent foundation for investigating the social activity of language use. As my study focuses on learning a specific language skill, technical writing, the social context of this learning cannot be separated from the learning; "language use and, thus, meaning are situated in interested, intersubjectively negotiated social interaction" (Lave, 1991, p. 67).

In this study, I consider the communities of practice that form in my participants' lives, whether at school, at work, or in other contexts. As described earlier, Wenger (1998a) states that a community of practice defines itself via its "joint enterprise as understood and continually renegotiated by its members" (p. 2), mutual engagement, and "shared repertoire of communal resources (routines, sensibilities, artifacts, vocabulary, styles, etc.)" (p. 2). These communities are not pre-defined, nor are they defined by organizational structure. Post-secondary computer science students may belong to multiple communities of practice (Vihavainen et al., 2013), for example, the community of computer science students at their institution, communities formed in each course, and, for some, via internships or other interactions, the community of a particular workplace or the broader community of practicing computer scientists. Students learn in these multiple communities and their experiences must be considered in light of the social context in which they occur.

Lave and Wenger (1991) explicitly set aside schooling when writing about Situated Learning Theory, but saw the consideration of schooling as future work. They stated that future research could consider "social organization of schools themselves into communities of practice, both official and interstitial, with varied forms of membership" (Lave and Wenger, 1991, p. 41). While my study does not focus on identifying communities of practice in an educational setting, Lave and Wenger's (1991) willingness to consider such a context implies that they believe communities of practice could exist in formal educational settings. They agree that educational settings are socially-situated contexts worthy of study: "schools themselves as social institutions and as places of learning constitute very specific contexts." (Lave and Wenger, 1991, p. 40)

Communication is a social practice, therefore it is appropriate to consider students' experiences of learning technical writing with focus on the communities in which the writing occurs. Lave and Wenger (1991) distinguish social practice as the “primary, generative phenomenon” (p. 34) with “learning is one of its characteristics” (p. 34). They contrast this with the view that cognitive processes are the primary phenomenon in learning. They position legitimate peripheral participation as “an analytical viewpoint on learning, a way of understanding learning” (p. 40) that exists “no matter which educational form provides a context for learning” (p. 40). So, although formal settings were not at the forefront when they developed their theories, these theories still apply to my study, which considers students' broad experiences learning technical writing across contexts. However, this is not to claim that students' experiences in school are always consistent with Lave and Wenger's (1991) definition of legitimate peripheral participation. As one example, large transmissive lectures generally do not involve participation in a legitimate professional practice (Vihavainen et al., 2013).

Since this study will investigate students' experiences and perspectives and I believe that an individual's experiences are socially constructed, it is important to consider an individual's identity when understanding their experiences. The explicit focus of Wenger's (2009) Social Theory of Learning on identity provides a lens to help me understand how an individual's identity affects their experiences.

Lave and Wenger (1991) hoped that the perspective of legitimate peripheral participation in communities of practice would “[draw] attention to key aspects of learning experience that may be overlooked” (p. 41). In this study, I hope to use their theories to investigate the socially-situated nature of learning technical writing that may have been overlooked in the literature.

2.2 Technical Writing in Computer Science Education

In this section, I discuss the ways in which technical writing has been taught and studied in computer science education and the ways in which students' experiences and perspectives related to learning technical writing have been studied in computer science education.

2.2.1 How Technical Writing Skills are Taught in Computer Science Programs

In this subsection, I focus on how technical writing skills have been taught in computer science programs, although these skills are often discussed in the context of communication skills more broadly. As described in Chapter 1, Writing Across the Curriculum (WAC) (Falkner and Falkner, 2012; Fell et al., 1996; Kay, 1998), Communication Across the Curriculum (CAC) (Gruba and Søndergaard, 2001; Karatsolis et al., 2011), or Writing in the Discipline (WID) (Kortsarts et al., 2010) are common initiatives that guide technical writing curricular projects in computer science education. These projects are situated across North America, Australia, and Asia, and generally involve a cross-section of an institution's computer science curriculum.

Further computer science education literature on technical writing provides information about how technical writing skills are taught via a single course at a particular institution rather than conceptualizing or theorizing about how students learn these skills in the discipline (Etlinger, 2006; Hazzan and Har-Shai, 2013, 2014; Kaczmarczyk, 2003; Kay, 1998; Michael, 2000; Pollock, 2001).

Institution-Wide Writing and Communication Initiatives

Writing across the curriculum projects, dating back to at least the 1970s (Rutz and Grawe, 2017), aim to integrate writing into disciplinary courses. Writing across the curriculum (Fell et al., 1996), or as Gruba and Søndergaard (2001) and Karatsolis et al. (2011) re-framed it, "communication across the curriculum" (CAC), was a common thread amongst those who had embedded communication skills into a cross-section of their program's computer science curriculum. 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" (Emig, 1977, p. 122). Falkner and Falkner (2012), citing Emig (1977), state that "higher cognitive functions develop most fully when supported by communications activities" (p. 379). Fell et al. (1996) 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). The success of WAC projects is based on the symbiotic relationship between learning a discipline through writing and learning writing through a discipline.

Kay (1998) describes a course that "complements and extends" (p. 117) his institution's WAC approach; in addition to WAC activities, his institution requires that students take an upper-division course with a significant writing component. The course Kay describes is taught by a computer scientist, with the belief that the instructor's disciplinary knowledge of writing practices is necessary to create integrated, authentic tasks and assessments. The goals of the course are to make writing the primary focus, teach appropriate tone and level (which varies depending on the audience), and have students write both long and short documents that take a variety of forms (e.g., email, technical reports, webpages).

Falkner and Falkner (2012) propose a methodology "designed to assist academics in the development of communication skills activities integrated with disciplinary content across the curriculum" (p. 379); specifically, they provide a computer-science-specific mapping of the WAC ideas with computer science concepts. They present three types of activities: writing to learn, such as reflections or forum postings; writing in the discipline, such as glossaries or user documentation; and communicating disciplinary content, such as presentations or performance reports. They believe that integrating writing to learn activities provide the necessary learning for success in writing in the discipline activities, and that practice with both writing to learn and writing in the discipline activities allows students to gain experience that is necessary to succeed in the culminating communicating disciplinary content activities.

Gruba and Søndergaard (2001) and Karatsolis et al. (2011) discuss work in the context of their universities' Communication Across the Curriculum (CAC) initiatives. Gruba and Søndergaard (2001) describe a course developed at the University of Melbourne in which the students were asked to plan and deliver a conference as an authentic activity that would allow them to develop their communication skills. Students were responsible for all aspects of the conference, with support from the instructors. Students found this new-to-them approach to learning challenging, but learned transferable skills through the experience. Karatsolis et al. (2011) describe their technical communication course that co-exists with communication components that are embedded into many (55%) of their existing courses. They suggest that computer science faculty members are well situated to teach writing as they have significant experience from their own disciplinary practice, despite having little formal training in writing. They present a tripartite model for teaching communication skills, where each part of the model is implemented in the technical communication course as well as in other courses. This broad implementation gives students multiple opportunities to learn and use their communication skills. The model starts with practising technical communication skills; for example, students might practice the writing skill of using parallel structure. The second part of the model is simulating practice; students could write a resume and cover letter in response to an ad for a mock job. In the third part of the model, students use their communicative skills in real contexts, for example by writing user documentation that will be distributed to real users.

Kortsarts et al. (2010) describe a three-step program where students learn oral and written communication skills across three courses, as consistent with their institution's Writing in the Discipline (WID) initiative. They focus on the communication skills taught via a project in the students' second core course in computer science. Students were expected to submit weekly written and oral progress reports in lab and give four oral presentations in lecture over the semester. This interdisciplinary project was conducted by computer science faculty members and an English/writing faculty member; the communication content was taught by the English/writing faculty member. Overall, they believe that integrating communicative skills explicitly into the course project was successful.

In a project supported by their university's Writing Development Initiative, Zhang et al. (2023) embedded writing instruction across five courses in their curriculum and provided centrally-managed, writing-focused teaching assistants and project management support. They found that their interventions improved students' writing, as evaluated by writing-focused research assistants, and that students also perceived an improvement in their own writing.

These institution- or department-wide communicative initiatives provide models for integrated, discipline-specific writing instruction (Falkner and Falkner, 2012; Gruba and Søndergaard, 2001; Karatsolis et al., 2011; Kay, 1998; Kortsarts et al., 2010; Zhang et al., 2023). The discipline-specific nature of these initiatives allows students to learn targeted writing skills (Falkner and Falkner, 2012; Karatsolis et al., 2011; Kay, 1998; Kortsarts et al., 2010; Zhang et al., 2023) that are selected as relevant learning activities by computer science faculty members in the context of non-communication-specific computer science courses.

Single Course to Teach Communication Skills

Pollock (2001), Hazzan and Har-Shai (2013, 2014), and Etlinger (2006) each discuss a single course intended to teach communication skills. In these publications, authors explained why learning technical communication skills is important for computer science students and then described a course at their institution that intends to teach these skills. These papers tend to give information that could be useful to others who want to implement similar courses, but lack details about how the author(s) conceptualized communication skills.

Etlinger (2006) describes a second-year course intended to give students practice writing and giving oral presentations, giving and receiving peer feedback, and writing professional documents (such as résumés). He describes the course assignments and the rationale behind each of them.

Hazzan and Har-Shai describe a course intended to teach soft skills (2013) and their analysis of students' experiences of learning these skills (2014). Their Situated Learning framework led them to embed students' learning opportunities into real-world contexts as much as possible. They analyzed students' assignment submissions over the term to create a model of students' mental construction of soft skills. The process begins with students distinguishing between soft and technical skills, then students characterize and rank soft skills, identify the soft skills they want to learn, and define the concept of a soft skill. Hazzan and Har-Shai include communication and technical writing skills in their definition of soft skills, but only two of the 14 scheduled class sessions were dedicated to writing.

Pollock (2001) describes how she evolved a graduate-level computer science course from its purely technical curriculum to become a communication-intensive technical course. She states that learning communication skills within the discipline is important, so that students are adequately prepared for conducting computer science research, but also for other careers that computer science graduate students pursue after graduation.

These papers provide examples for faculty members who wish to create a single course to teach communication skills. These resources generally provide enough information that an educator could adopt and adapt the described curriculum to suit their context. The single-course model contrasts with the institution- or department-wide communicative initiatives described above although it's possible for an institution to combine both approaches.

2.2.2 Studies of Students' Experiences and Perspectives about Learning Technical Writing in Computer Science

Limited research has explored students' experiences and perspectives of learning technical writing in computer science at the undergraduate level.

Hazzan and Har-Shai (2014) conducted surveys to gather student perspectives' about their course on soft skills. They asked rating-scale multiple choice questions (Andres, 2012) about student experience where the response options ranged from 'Not at all' to 'To a very great degree' (e.g., "To what degree did the course improve [your] ability to offer feedback?" and "To what degree did the course improve [your] teamwork skills?"). They asked few open-ended questions, and therefore were not able to dig deeper into the reasons beneath students' opinions about their experiences developing soft skills in computer science. 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.

Munir et al. (2022) explored common problems in upper-year computer science students' writing but did not collect data on students' experiences or perspectives about learning technical writing. In a following paper from the same group, Zhang et al. (2023) described a project that aimed to "collaborate with a writing specialist and a community of course instructors, centralize the management of writing teaching assistants, and introduce a variety of relevant genres and contexts to help students develop and apply writing skills" (p. 610). The researchers gathered students' perceptions on their experiences via a survey with 86 first-year and 49 second-year student respondents and conducted a thematic analysis on the qualitative data. They found some evidence that students' perception of writing changed after completing writing activities designed by the project. Students perceived report writing to be challenging. Further, writing documentation was new to 12 of the participants and they had positive attitudes towards learning this skill. Zhang et al. (2023) noted that 10 first-year students perceived that the "writing process helped them better understand their code and other [computer science] concepts" (p. 614). Finally, they also found that students prefer more, smaller, integrated writing assessments, appreciate being able to resubmit their work, and want to be provided with more examples of high-quality technical writing.

Cilliers (2012) conducted a study of students' perceptions of integrating writing skills into a programming course. Their survey asked students about motivation, mastery of writing skills, and self-efficacy with an aim to measure "perceived benefit of each of a number of academic writing interventions" (Cilliers, 2012, p. 1028). They found that students perceive most writing activities as beneficial, but the activities most often deployed by instructors are not necessarily the ones that students perceive to be the most useful.

These three projects (Cilliers, 2012; Hazzan and Har-Shai, 2014; Zhang et al., 2023) begin to explore students' experiences of learning communicative skills in computer science courses. Their findings that students want to learn communicative skills (Hazzan and Har-Shai, 2014) and positively experience this learning (Cilliers, 2012; Zhang et al., 2023) are promising, but leave room for additional exploration.

2.3 Automated Assessment Tools in Computer Science Education

Automated Assessment Tools (AATs) have been used in computer science courses for decades, but have become more common recently with at least 778 publications on the topic between 2010 and 2021 (Luxton-Reilly et al., 2023; Paiva et al., 2022). Scholars have focused on how automated assessment tools 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 (Ala-Mutka, 2005; Baniassad et al., 2021;

Leinonen et al., 2022; Luxton-Reilly et al., 2023; Mitra, 2023; Paiva et al., 2022; Rao, 2019; Zamprogno et al., 2020). Luxton-Reilly et al. (2023) found that "providing automated functional testing was the most common form of assessment" (p. 8) in software engineering and computer science courses but that automated testing was not limited to testing functionality. Some AATs assess code style via measurable software metrics and find problems such as unused variables, empty catch blocks, and inconsistent code formatting (Paiva et al., 2022), but not all quality issues can be automatically assessed. AATs can also measure test case correctness and thoroughness and check for plagiarism (Paiva et al., 2022). The limitations in what AATs can assess puts constraints on grading practices and, perhaps intentionally, causes students to prioritize adherence to measurable aspects of their work.

Leinonen et al. (2022) studied how different approaches to providing feedback affected student behaviours in an introductory computer science course. They found different patterns of student behaviour when they provided feedback immediately or at a scheduled time. Similarly, Baniassad et al. (2021) found that students' behaviours changed when they implemented regression penalties[2]. In consideration of the AAT literature, Luxton-Reilly et al. (2023) state "[a]lthough some reviews have considered aspects of AATs other than implementation details, such as measures of effectiveness, 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) My study gathered student perspectives on how the use of automated assessment tools changes their behaviours and priorities and may give further evidence to support Ala-Mutka (2005) who argues for educationally-sound assessments that are not simply driven by available technology.

2.4 Situating this Study in Related Literature

This research study relies on Lave and Wenger's (1991) Situated Learning Theory and Wenger's (1998b) Social Theory of Learning as its theoretical foundation.

[2] Baniassad et al. (2021) defined regressions as scores that went down; they gave penalties for such scores.

These underlying theories influenced the research questions that I chose to explore, the data collection plans and the way that I conducted the research, and the way that I conducted data analysis and generated themes. Throughout the research design and implementation, I consciously considered these theories and how they affected decisions and actions that I took.

Although some researchers have examined students' experiences of learning technical writing skills and the affect of assessment practices that rely on automated assessment tools, these topics are underexplored. As stated in Chapter 1, the dearth of research on computer science students' experiences of learning communication skills provides an opportunity to learn from students and better understand curricular and pedagogical gaps. This study will further investigate students' experiences and student learning of technical writing, beginning to fill the gap in the literature.

Chapter 3

Research Design

My research goal was to investigate students' experiences learning about technical writing and students' perspectives towards technical writing. As described in Section 2.1.2, I brought an interpretive theoretical perspective to the research. I wanted to understand and interpret the multiple, socially-situated realities of the participants in the study.

I conducted an interpretive qualitative research study. I chose to collect data via interviews and reflective journals and analyze the data via reflective thematic analysis. In this chapter, I describe the ethics of this study, my positionality, the participants, data collection methods, data analysis methods, quality considerations of this research study, and the limitations of my research design.

3.1 Research Ethics

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

I was conscious of conducting myself ethically during every stage of the research; "[b]ecause the objects of inquiry in interviewing are human beings, extreme care must be taken to avoid any harm to them" (Fontana and Frey, 1994, p. 372). My ethical duty could have been challenging to navigate; it goes beyond receiving approval from the ethics board and informed consent from participants and must include consideration of the "potential interpersonal impacts of the inquiry"

(Marshall and Rossman, 2016, p. 57). Although ethical dilemmas often arise in the midst of research and cannot be anticipated or planned for (Ellis, 2007; Marshall and Rossman, 2016), I did not encounter situations that presented an ethical dilemma.

As a faculty member in the Department of Computer Science at UBC, I was particularly conscious of the power that I have in relation to the participants who are UBC computer science students. In the consent form, I noted my dual role as researcher and Associate Professor of Teaching in the department. The consent form also stated that I would not be sharing individual comments with the department, but would be sharing the overall findings publicly, and the findings would therefore be available to all department members. I used pseudonyms for all participants but I let them know that they may still be identifiable.

I asked students questions about their experiences and perspectives in relation to courses offered by my department. Some participants seemed aware that I was a faculty member and others did not. In the interviews, I focused on the students' experiences and did not explicitly draw attention to my role as a faculty member in the department.

In any research study, there is a risk that participants may consciously or subconsciously be affected by what the researcher wants to hear (Bergen and Labonté, 2020). In my study, my dual role may have heightened this risk. Upon reflection on my experiences interviewing participants, I believe that participants were open in sharing their experiences. I did not sense reluctance in participants' responses and some participants seemed unaware that I was a faculty member, despite my disclosure on the consent form. For example, participants often described courses to me as if I had no knowledge of the curriculum. Therefore, I do not believe that my dual role was overly influential in the data I collected.

3.2 Positionality

Holmes (2020) defines positionality as "an individual's world view and the position they adopt about a research task and its social and political context" (p. 1). Holmes (2020) suggests that researchers locate themselves in terms of the subject, the participants, and the research context and process. In this subsection, I describe my positionality and its relation to this research study.

As the researcher, and as is common in qualitative research, I was the primary instrument of data gathering and analysis (Merriam, 1998). Therefore, my personal subjectivity, history, and lived experience have been integrated into all phases of this research. In relation to the subject, I have more than 15 years of experience teaching and working with undergraduate computer science students at UBC. I have varying degrees of knowledge of the particular courses that the study participants discussed.

In relation to the participants, I had not previously met any participants but they may have been aware that I was a faculty member in the Department of Computer Science. My faculty position gives me authority and power that another researcher may not hold.

In relation to the research process, my positionality influences the way that I conduct research. Braun et al. (2019) describe their view on researcher positionality: "[t]he researcher is a storyteller, actively engaged in interpreting data through the lens of their own cultural membership and social positionings, their theoretical assumptions and ideological commitments, as well as their scholarly knowledge" (pp. 848-849). A core principle of reflexive thematic analysis is that researcher positionality is a resource that leads to knowledge development (Braun et al., 2019). Braun and Clarke (2019) describe coding as "creative labour" (p. 594), highlighting the labour-intensive and subjective nature of the analytic process. Further, they describe the goal of reflexive thematic analysis as "not to "accurately" summarize the data, nor to minimize the influence of researcher subjectivity on the analytic process, because neither is seen as possible nor indeed desirable. The aim is to provide a coherent and compelling interpretation of the data, grounded in the data." (Braun et al., 2019, p. 848) The analysis and themes discussed in this dissertation stem from my creative labour. Another researcher would likely have generated different themes.

3.3 Participants

I recruited third-year or fourth-year students who were taking a minimum of two computer science courses in the term that I conducted the research. I focused recruitment on these participants so that I could learn about their experience in the program rather than focusing on a specific course. My goal is to untangle students' experiences with learning how to write technical documents from the course-specific technical content that the students are learning. By recruiting students who are taking more than one computer science course, I was able to ask them about their experiences across courses. I wanted to learn how they experience learning these skills broadly across the curriculum rather than in a specific course taught by a specific faculty member.

I planned to work with 10 participants in this study because I thought they would provide sufficiently rich data on students' experiences and perspectives. I recruited participants via email; the UBC computer science undergraduate program emailed my study invitation to all third- and fourth-year computer science students and 16 students expressed interest. Two students were ineligible for the study because they were not taking two computer science courses. The remaining 14 potential participants were sent the consent form and a notice of COVID-related risks of research for their review. Two declined to participate due to the time required for the study. Of the 12 remaining potential participants, 10 were selected at random. I used random selection in order to avoid any unintentional bias in choosing participants.

I recruited a diverse group of students; see Table 3.1 for demographic details, including participant (pseudo)names[1], genders[2], secondary school location, number of languages spoken, whether they are a native English speaker, and whether they are fluent in English.

I hoped to recruit students with a range of English fluency, but nine of the 10 participants were fluent in English and only one considers themself to be an English Language Learner. I asked students to identify their own English fluency rather than asking them to take a test because I am interested in their perception and understanding of their fluency and educational experiences. Nine of the 10 participants are multi-lingual. Half of the participants attended secondary school in Canada.

[1] A pseudonym is a fictitious name chosen to protect the individual's privacy.

[2] 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 were also told they did not have to answer.

Table 3.1: Participant Demographics

Name	Gender	Secondary School Location	Languages Spoken	Native English Speaker	Fluent in English
Adrian	non-binary person	outside Canada	2	No	Yes
David	man	in Canada	2	Yes	Yes
Etson	man	outside Canada	2	No	No
Jason	man	in Canada	1	Yes	Yes
Krish	man	outside Canada	2	No	Yes
Nadir	man	outside 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	outside Canada	3	No	Yes

Nine of the 10 participants were enrolled in computer science programs within the Faculty of Science while one was in a combined business and computer science program. Of the nine Science students, three were enrolled in a second degree program and had completed a previous undergraduate degree in a different field.

The participants are high-achieving and highly involved students in the Department of Computer Science. Five work as teaching assistants for computer science courses. Seven have computer science-related work experience through internships, the co-operative education program, or the work learn program. Three have conducted research with graduate students, post-doctoral fellows, or professors in the department.

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.

One participant is a mono-lingual English speaker. Languages represented by the nine multi-lingual participants are Cantonese, English, French, Hindi, Indonesian, Korean, Mandarin, Russian, and Vietnamese.

Each participant received $100 in gift cards to a preferred vendor for their participation in the study.

3.4 Data Collection

I chose to collect data over one semester through beginning-of-term and end-of-term interviews and weekly reflective journals.

3.4.1 Rationale for Data Collection Methods

In this section, I explain the rationale for using these data collection methods.

Interviews

Interviews are a common data collection method in qualitative research. They attempt to understand "the lived experience of other people and the meaning they make of that experience" (Seidman, 2019, p. 9)[3]. Interview formats fall on a spectrum from structured interviews, which mimic a written questionnaire with no variation in questions across participants, to completely unstructured conversations between interviewer and interviewee. However, Brinkmann (2018) states that the ends of the spectrum are only possible in theory because even the most structured interview will contain some unexpected conversation and even the most unstructured interview must have a purpose and opening question. I used the most common interview in the social sciences: the face-to-face semi-structured interview (Brinkmann, 2018). A semi-structured interview is purposefully planned and at least partially follows an interview guide, but allows for follow up questions, and permits the plan to evolve during the interview depending on what the interviewee says. This kind of interview allowed me to engage participants in a consistent way while leaving flexibility to probe individual experiences and beliefs.

Brinkmann (2018) describes interviews as "humane, intersubjective, and responsive encounters" (p. 578) that occur between two (or more) individuals at a

[3] Seidman (2019) focuses on phenomenological interviews. Although I used interviews in a different context, much of his writing related to interviews is still relevant.

particular place and time. The intersubjectivity of interviews makes them an appropriate data collection method for an interpretive qualitative research study. The intersubjectivity affects the way that each participant engages in the interview and therefore affects the information that may be generated, which is inherent to an interpretive study; I, as the researcher, and the participants engaged in a conversation that was unique to our individual circumstances. The data generated from the interviews depended on both my and the participant's willingness to engage in deep conversation (Marshall and Rossman, 2016). The interviewee and I both perceived our own reality and made own own meaning of the interview. Therefore, the interview data collected reflect "a reality constructed by the interviewee and interviewer" (Brinkmann, 2018, p. 587) with my reality, as the researcher, likely to be the story that is told (Stake, 2005). This is consistent with reflexive thematic analysis; my perspective and subjectivity is valued as a "analytic *resource*" (Braun and Clarke, 2020, p. 3) in generating themes from this study.

Reflective Journaling

I chose to collect reflective journals so that participants could further share their lived experience, their thoughts, opinions, and reflections, and the significance of their experiences (Paterson, 1995).

Reflective journaling can take many forms, such as paper-based, video, online or via email (Janesick, 1999; Parikh et al., 2012) and can serve multiple purposes. I chose written journaling (on paper or online) for this study. I rely on Moon's (2013) definition of reflection: "mental processing with a purpose" (p. 4) that has "close association with, or involvement in, learning and the representation of learning" (p. 4). Through journaling, participants describe their lived experience, their thoughts, opinions, and reflections, and the significance of their experiences (Paterson, 1995). Reflective journals for the context of this interpretive qualitative study on educational experiences, therefore, involved mental processing of learning, learning experiences, or representations of learning.

Reflective journals can gather data and provide understandings of learning experiences that can be difficult to gather via other research methods (Phelps, 2005), including "reflective insights ... which can have a highly significant impact on learners" (Phelps, 2005, p. 42). I am unable to observe all events that influence an individual's experience, and am never able to observe an individual's thoughts or feelings about an event, but a journal can capture such events and reactions. Further, journaling can give research participants an "active voice" (Janesick, 1999, p. 522) and allow them "to reflect, to dig deeper if you will into the heart of the words, beliefs, and behaviors" (Janesick, 1999, p. 513) of their experiences. Journal writing can be "a tangible way to evaluate our experience, improve and clarify one's thinking" (Janesick, 1999, p. 521). Reflective journaling can be highly personal, serving to provide "a holistic and unexpected portal into the everyday life" (Rodriguez, 2017, p. 139) and "a vehicle for inner dialogue that connects thoughts, feelings, and action" (Hubbs and Brand, 2005, 62). Participants may feel more comfortable sharing these personal thoughts in a journal than they would in a face-to-face interview (Rodriguez, 2017). Journaling also gives participants time to reflect and carefully compose their thoughts, in comparison to an interview where they must answer relatively quickly. The combination of these two data collection methods provided more data than if I only used face-to-face interviews.

Rodriguez (2017) and Lutz and Paretti (2019) used reflective journaling as a research method in educational case studies; both groups found it to be an effective method. Rodriguez (2017) found that reflection on academic experiences was "a method of discovering and framing truths" (p. 127). Lutz and Paretti (2019) found that reflective journaling in response to prompts is effective at gathering experiences and perceptions from participants. A participant understands their own experience better than any one else can, even if they are not able to understand or articulate all factors that affect their experience, so reflection can help participants understand their experiences even more deeply (Phelps, 2005). Lutz and Paretti (2019) describe the extensibility of reflective journaling as a research method; it can gather data efficiently from many participants. However, the time needed for analysis increases proportionally to the number of entries collected. I limited the reflective journal entries to once per week per participant primarily to respect participants' time, but also to keep the time needed for analysis manageable.

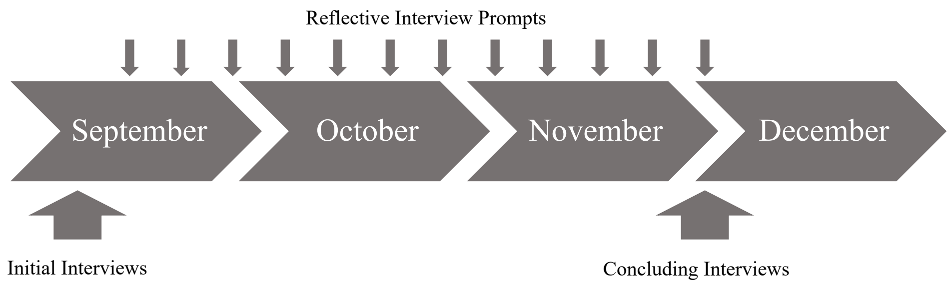

Figure 3.1: Data Collection Timeline Over the Fall 2022 Semester.

3.4.2 Data Collection Procedures

In this section, I describe the data collection procedures. The data collection timeline is shown in Figure 3.1.

Initial Interviews

Interviewers need to be skilled and well prepared for each interview (Marshall and Rossman, 2016). They need to be able to ask questions that elicit the interviewee's descriptions and reflections and, crucially, must be careful listeners who can ask follow-up questions for more information when necessary. In general, for interpretive research, interviewers will use open-ended questions that invite the interviewee to share their experiences and thoughts. They should avoid leading questions that suggest ideas to interviewees. I prepared for the initial interviews by drafting open-ended questions and gathering feedback from my PhD supervisory committee on the planned questions.

The initial interviews were conducted in a project room in the UBC Computer Science building. My choice to conduct interviews in a space other than my office was intentional; I used a project room as a neutral space in an attempt to reduce the power dynamics.

The participant read the notice of COVID-related risks and read and signed the consent form before we began the interview. I reminded the participant that the interview would be recorded and started the audio recording.

Each initial interview followed the same interview plan but differed in its execution as I asked follow-up questions that were dependent on the participant's previous responses. The planned questions are available in Appendix A. Each interview took 40 to 60 minutes. During the interview, I listened attentively while making notes about follow-up questions I wanted to ask. I attempted to build rapport with each interviewee with a friendly, curious attitude and active listening but balanced the closeness of the relationship with an appropriate distance between myself and the participant. Seidman states that "the desire to build rapport with the participant can transform the interviewing relationship into a full "We" relationship in which the question of whose experience is being related and whose meaning is being made is critically confounded" (Seidman, 2019, p. 102). I listened and asked follow-ups and with two brief exceptions refrained from sharing my experiences with participants.

After the interviews were complete, I uploaded the audio recordings to NVivo Transcription (Lumivero, 2024b) and they were automatically transcribed. I reviewed each transcript and made edits, where necessary, for accuracy.

Reflective Journaling

Participants were asked to write for 15-20 minutes per week in response to reflective prompts. Each participant was given a choice about whether they'd like to write in a physical journal or an online survey form. Nine participants chose to write in an online survey form and one chose a physical journal. I transcribed the entries from the physical journal at the end of the data collection period. The full text of the prompts is available in Appendix B.

The first reflective journal 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.

Concluding Interviews

The concluding interviews were conducted in a similar manner to the initial interviews. Nine interviews were again conducted in a project room in the UBC Computer Science building. The tenth interview was conducted online via Zoom due to a participant's illness.

On arrival for the in-person interviews, I offered each participant an opportunity to re-read the consent form they had previously signed since it had been over two months since the study began. I reminded the participant that the interview would be recorded and started the audio recording. For the online interview, I emailed the consent form prior to the interview and at the beginning of the interview I checked that the participant had a chance to re-read it in order to remember what they had consented to.

I again conducted semi-structured interviews; each interview followed the same interview plan but differed in its execution as I asked follow-up questions that were dependent on the participant's previous responses. The planned questions are available in Appendix C. Each interview took 40 to 65 minutes.

After all the interviews were complete, I uploaded the audio recordings to NVivo Transcription (Lumivero, 2024b) and they were automatically transcribed. I reviewed each transcript and made edits, where necessary, for accuracy.

3.5 Data Analysis

I conducted a reflexive thematic analysis of the data corpus to construct themes (Braun and Clarke, 2006).

3.5.1 Rationale for Data Analysis Method

Thematic analysis is a broad term for a number of approaches to finding meaning in a qualitative data corpus (Braun et al., 2019). Braun et al. (2019) describe what they see as three approaches to thematic analysis: a coding reliability approach, a codebook approach, and a reflexive approach. Briefly, coding reliability is a post-positivist[4], scientific approach to thematic analysis. Its goal is consensus via a

[4]Positivism "postulates the objective existence of meaningful reality. It considers such meaningful reality to be value-neutral, ahistorical and cross-cultural" (Crotty, 1998, p. 40); post-positivism

"singular, shared, and "correct"" (Braun et al., 2019, p. 847) analysis. 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" (Braun et al., 2019, p. 848). Braun et al. (2019) describe codebook thematic analysis as a collection of methods that sit somewhere between coding reliability and reflexive thematic analysis. This approach may use a structured process but aligns more closely with a non-post-positivist qualitative framework.

My theoretical perspective and research questions align with Braun et al.'s (2019) approach to qualitative research, where meaning is not universal, but instead influenced by the context of the research and the researcher or researchers themselves. Their approach is "informed by the unique standpoint of the researcher, and ... is fluid, flexible and responsive to the researcher's evolving engagement with their data" (Clarke et al., 2015, p. 223); 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 (2019) views that researcher subjectivity is a resource to be valued, rather than a limitation to be mitigated. As the main researcher, I am interpreting the data, constructing themes, and sharing the stories. My findings should therefore be "understood as contingent upon the specific context and the particular interpretive/theoretical lens" (Finlay, 2021, p. 104) through which they were generated.

3.5.2 Data Analysis Procedure

I followed Braun and Clarke's (2020) six-step analysis procedure to conduct a reflexive thematic analysis of my data corpus. As I moved through the analytic process, I read and re-read *Thematic analysis: A practical guide*, a recent book published by Braun and Clarke (2022a). The analytic steps included:

1. data familiarization and writing familiarization notes;

2. systematic data coding;

relaxes the absoluteness of positivism and its "claims to validity are tentative and qualified" (Crotty, 1998, p. 40).

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

Braun and Clarke (2020) are clear that these six steps are not a linear process, instead, they should be followed iteratively, moving between steps as required. I describe my analytic process in detail below.

The goal of this process is to generate themes, where themes are "patterns of shared meaning, united by a central concept or idea" (Braun and Clarke, 2020). 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 somehow already existing in the data waiting to emerge.

Step 1: Familiarizing Myself with my Data

Interview audio recordings were uploaded to NVivo Transcription (Lumivero, 2024b), an online automatic transcription service, which generated an initial draft transcription for each interview. The draft transcript noted the speaker and a time stamp for each entry. I did a careful pass on each transcript, listening to the audio recording and correcting errors in the draft transcript. There were occasional phrases or words that I could not understand in the audio recording; these utterances were marked as unknown in the transcripts.

I attempted to capture exactly what each participant said in each interview; however, it's unlikely that my transcription perfectly captures participants' words and intent given the large corpus and inherent subjectivity of the task. As an example, I inserted punctuation into the transcripts as I understood participants' meaning but a misinterpretation in my transcription could alter the intended meaning.

Completing the interview transcription was the first step I took in familiarizing myself with the data. I spent 1.5 to 4 hours per transcript, often re-listening to responses many times.

Once all transcripts were complete, I read each one to further familiarize myself with the data. I took notes as I went, writing down topics that participants discussed, things that struck me, and phrases or concepts that were common across interviews. I then created a concept map to group similar ideas together so that I could see a summary of my notes on one page.

I started analysis before data collection was complete; I was working on familiarizing myself with the data and systematic data coding (see Step 2 below) of initial interviews and the available reflective journal entries before I conducted the concluding interviews.

Not all of the reflective journal entries were available when analysis began. I re-read the entries to familiarize myself with the data and then coded the entries that were available with the initial interviews. I continued to add journal entries to the corpus and code them as they became available.

Once the concluding interviews were conducted, I followed the same process of transcription and note taking of those audio recordings.

Step 2: Systematic Data Coding

I engaged in the "creative labour" (Braun and Clarke, 2019, p. 594) of coding over a period of four months. I used NVivo (Lumivero, 2024a) to code the transcripts and reflective journal responses. My goal was to be slow and careful, iterating on the codes that I was creating as I read and re-read the transcripts and reflective journal responses.

Reflexive thematic analysis can use an inductive approach – developing codes that are grounded in data – or a deductive approach – coding based on a pre-determined conceptual or theoretical framework (Braun and Clarke, 2020). The approach taken lies on a spectrum ranging from inductive to deductive (Braun and Clarke, 2020); my approach was primarily inductive. I created codes based on data. However, my analysis was influenced by Lave and Wenger's (1991) Situated Learning Theory and Wenger's (2009) Social Theory of Learning.

I chose to use both semantic and latent codes. Braun et al. (2019) describe semantic codes as ones that closely match participant's language and stay at the ""surface" of the data" (p. 853) and latent codes as ones that capture a "deeper, more *implicit* or conceptual level of meaning" (p. 853). As an example of capturing "surface" meaning, I coded the text "I would say that the attitude of just being prepared to be wrong or open to being corrected by anybody was really helpful for just fixing things fast without getting too stuck or afraid of getting feedback" with the code 'openness to feedback'. As an example of implicit meaning that I coded with the latent code 'no mention of technical writing in initial description of work-related tasks', I coded participants' descriptions of their work-related tasks that did not include any writing tasks.

I found that I was initially focusing on semantic codes but as I worked my way through the coding and theme generation I was more able to abstract away from the specific comments and find deeper meaning. Braun et al. (2019) suggest that initial coding tends to be semantic and that "latent-level meaning can be easier to "see"" (p. 853) at later stages of analysis.

I revisited my codes multiple times through the process and merged, deleted, or clarified as necessary. In early coding, I went back to the previously-coded items and re-coded after major changes to the codebook.

I did two full coding passes on the corpus. I worked my way through the corpus in a different order in the second round of coding, as suggested by Braun and Clarke (2022a). During the second round of coding, I paid particular attention to my research questions and areas of focus.

I kept notes about my ideas, reflections, and questions throughout the analysis process and have regularly revisited and reflected on these notes.

Step 3: Generating Initial Themes from Coded and Collated Data

Throughout the coding process I was beginning to consider and generate possible themes. I was becoming aware of commonalities in the participants' experiences, reflections, and perspectives. I revisited these initial commonalities in the concluding interviews in order to collect further data on these ideas.

During coding, I grouped common codes together into groups in NVivo (Lumivero, 2024a). Although I didn't construct a theme from any single code group directly, the code groups started to guide and organize my thinking around common meaning.

Once I had completed two thorough coding passes on the data corpus, I began constructing potential themes. I used NVivo (Lumivero, 2024a) to construct themes by collecting similar codes and code groups together. I constructed initial themes and then took a break from my data and analysis before coming back to re-examine the initial themes. I continued to iterate on the initial themes, with breaks, until I felt that the themes I had constructed captured the meaning in my data.

Step 4: Developing and Reviewing Themes

After the initial theme construction, I examined the potential themes, their associated codes, and the data coded to each theme. In this stage, I identified some codes that were serving as topic summaries rather than encoding deeper meaning in the data; I re-coded those data snippets into a set of more meaningful codes. I further tweaked themes by moving codes between themes, re-naming themes, and splitting some themes into two where each represented a more concrete idea.

I reviewed themes to check that I constructed themes that represent "something important about the data in relation to the research question ... [that] represents some level of *patterned* response or meaning within the data set" (Braun and Clarke, 2006, p. 82). As Braun and Clarke (2006) suggest, I was flexible about what is considered a theme and did not attempt to define a rigid quantifiable measurement that determines whether something is considered a theme.

I created themes at a latent level, my intent was to not only consider "what a participant has said or written" (Braun and Clarke, 2006, p. 84), but also to "identify or examine the underlying ideas, assumptions, and conceptualizations – and ideologies – that are theorized as shaping or informing the semantic content of the data" (Braun and Clarke, 2006, p. 84).

I then re-engaged with the data corpus with my candidate themes in mind. I did not revise my themes through this process but did code a small number of additional text snippets.

Step 5: Refining, Defining, and Naming Themes

After developing the themes, I drafted names for them. I then reviewed the themes with my research peer, Ashley Welsh[5]. We discussed the themes and individual codes, but my research ethics approval does not allow me to share the raw data

[5] I discuss my research peer in Section 3.6.1.

with anyone except my supervisor. Ashley asked me questions about the themes and after our meeting I made minor changes to the theme names or codes assigned to each them. Ashley and I noticed that I had one commonly-used code that did not capture participants' meaning; instead, it captured the topic of "learning in community". I re-visited all data that was assigned this code and re-coded with more specific codes that captured participants' meaning.

I drafted "theme definition[s]" (Braun and Clarke, 2022b, p. 108), or short descriptions of each theme, and shared them with my committee prior to a meeting in which we discussed the themes. Again, my ethics approval prevented me from sharing the raw data corpus with my full committee. My committee asked me probing questions about the themes and after our meetings, I made further minor changes to the theme names or codes assigned to each theme. For example, they asked questions that helped me identify further commonality around the area of community in my initial themes. In my meetings with Ashley and my committee, we also discussed the potential implications of the findings and thought about the "so what" for my study, which further helped clarify the themes that I have generated and believe will be useful to share with the community.

Step 6: Writing the Report

My dissertation will serve as the initial report on this study. I intend to publish the findings in other academic venues, but have not done so yet.

In reflexive thematic analysis, you are "producing your analysis as you write it, not simply describing the analysis you finished before writing started" (Braun and Clarke, 2022b, p. 118). I began writing this dissertation in parallel with the previous five analysis steps and the writing process has prompted me to revisit and re-organize the themes and the way that I present them. Originally, I planned to present each of the eight themes individually but I found that presentation to be unorganized. I re-organized the themes into three groups, each of which answers a research question. The act of writing, re-writing, and editing helped me clarify the main findings of my study.

In reporting the themes, I made choices that are consistent with my analytic method of reflexive thematic analysis. Each theme had many associated codes and participant quotes. When presenting the findings, I chose the quotes that best illustrated the theme that I was discussing. I aimed to follow Braun and Clarke's (2006) advice: "[c]hoose particularly vivid examples, or extracts which capture the essence of the point you are demonstrating, without unnecessary complexity" (p. 17). Further, I have intentionally avoided quantifying the number of participants who expressed an idea that contributes to a theme but have, at times, provided qualifiers such as 'many', 'most', or 'some'. As a researcher employing reflexive thematic analysis, I was generating themes that matched patterns in the data; I was not using quantitative counts to determine whether a pattern was relevant enough to include as a theme.

As I was writing, I found that I wanted to describe participants' foundational beliefs about technical writing before discussing their experiences and perspectives in more detail. I decided to share participants' definitions of technical writing and their reflections on their own technical writing in an initial results section before discussing the remainder of the themes.

I conducted member checks with participants as I was writing the draft of this dissertation. I describe the member checks more fully in Section 3.6.1; they did not provide sufficient new data to require reconsidering my themes.

In reporting participant quotes, I elided filler words such as 'like', 'um', and 'right' when I could improve readability and maintain authenticity to the participant's natural speech (Lingard, 2019).

3.6 Quality Considerations

Reliability, objectivity, and generalizability – criteria for judging the quality of quantitative research – are often applied inappropriately as quality measurements for qualitative research (Tracy, 2010). Qualitative research is then criticized for its inability to meet these inappropriate standards (Marshall and Rossman, 2016). Much qualitative research does not share the same epistemological foundation or theoretical perspective as the quantitative research for which the standards were created, thereby rendering the criticism inapplicable (Tracy, 2010). Scholars have developed, discussed, and debated appropriate criteria for measuring the quality of qualitative research (e.g., Braun and Clarke, 2020; Guba and Lincoln, 1981; Lincoln and Guba, 1985; Marshall and Rossman, 2016; Morse, 2018; Tracy, 2010). Lincoln and Guba (1985) spurred a debate that still continues: how should qualitative work be judged, and should there be explicit criteria or checklists against which to judge it or is qualitative work too broad an umbrella to attempt to fit into one set of criteria (Marshall and Rossman, 2016)? Braun and Clarke (2020) argue that qualitative research should be evaluated on its own terms, in consideration of the particular approach that was used.

Since there is more than one way to conduct thematic analysis, as described in Section 3.5.1, there is not a single approach to considering quality in thematic analysis. For this project, I rely on the principle of trustworthiness that Lincoln and Guba (1985) propose as a quality measure for qualitative work and the quality considerations for reflexive thematic analysis put forth by Braun and Clarke (2020).

3.6.1 Trustworthiness of this Qualitative Research Study

To enhance the trustworthiness and credibility of my study, I used Lincoln and Guba's (1985) strategies of prolonged engagement, member checks, collecting data from multiple sources, and peer debriefing.

Prolonged Engagement

I engaged with each participant over the course of one full semester. The prolonged engagement allowed me to build rapport with participants and presented multiple opportunities to follow up on findings from initial data collection.

Member Checks

Lincoln and Guba (1985) suggest conducting member checks; they define member checks as a process "whereby data, analytic categories, interpretations, and conclusions are tested with members of those stakeholding groups from whom the data were originally collected" (p. 314). There is some tension between the idea of asking members to check my findings and the inherent tenet of reflexive thematic analysis that I, as the researcher, bring a unique and valued perspective to the work and that my perspective is a resource in generating the themes. To balance this tension, I conducted member checks in two ways. First, in the concluding interviews, I asked further questions about commonalities and interesting findings from the initial interviews or reflective journals. These follow up questions gave participants a chance to elaborate their thoughts on the preliminary findings.

Second, after analysis was complete, I invited participants to review the themes with me. Eight of the 10 participants met with me for 20-30 minutes. Prior to the meeting, I sent each participant their initial- and concluding-interview transcripts, all of their reflective writing, and a summary of the themes I generated. During the meeting, I first asked if they reviewed the transcripts or reflective writing; if they had, I asked if there were any inconsistencies that they found. No participants mentioned any inconsistencies, but most had not read their transcripts and reflective writing.

The majority of the meeting time was spent reviewing themes. The summary I shared in advance is available in Section D. I had grouped the themes by research question[6]:

1. How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?
2. How do course assessment practices affect students' learning of technical writing?
3. In what ways do students hold perspectives that may be barriers to their learning of technical writing?

For each group, I described each theme and gave some examples of participant quotes that supported the theme. I asked participants if what I shared felt consistent or inconsistent with their experiences and if there was anything that they wanted to clarify or augment. Most participants felt that the themes were consistent with their experiences. I have included the contrasting views in the results chapters.

Multiple participants augmented their previous contributions by discussing how the use of ChatGPT (Open AI, 2024) and other Generative AI tools have influenced their practice of learning and doing technical writing. I did not collect enough

[6]The order in which the research questions were presented to participants in the member checks is different than in this dissertation. During my writing process, after member checks were complete, I re-organized the order in which I present results.

data during the member checks to include these contributions in the data analysis. Generative AI tools were not mentioned in the original data collection because ChatGPT was initially released seven days before I completed the concluding interviews for this study, therefore, ChatGPT and other Generative AI tools were not in common use at the time of data collection.

Collecting Data from Multiple Sources

I collected data from each participant via interviews and reflective journaling and had a minimum of nine data collection opportunities for each participant. These data sources allowed me to gather both written and verbal contributions from participants.

Peer Debrief

I debriefed with my friend and colleague Ashley Welsh, PhD, during the study. She works at UBC's Centre for Teaching, Learning, and Technology as a Faculty Liaison for Science. Her PhD and Masters degrees are from the Department of Curriculum and Pedagogy at UBC and her current work is broad, spanning teaching and learning projects across the Faculty of Science. Her expertise and experience with post-secondary education and educational research and willingness to ask hard questions made her a valuable critical friend who questioned my thinking and helped me see additional points of view.

3.6.2 Quality Considerations of my Reflexive Thematic Analysis

Two key reflexive thematic analysis quality considerations raised by Braun and Clarke (2020) are:

1. The analysis procedures must be consistent with the "paradigmatic and epistemological assumptions" (p. 331) and those assumptions must be articulated.

2. Researchers must explain their use of and orientation to reflexive thematic analysis.

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

Braun and Clarke (2020) also provide a set of 20 questions that can be used to evaluate the quality of thematic analysis. I don't present the full list here, but invite readers to use their criteria when considering the quality of this work and the quality of other reflexive thematic analyses. I am confident that I conducted this reflexive thematic analysis in a principled way and that I have provided sufficient information for readers to evaluate the quality of this work.

3.7 Limitations

All research has limitations; researchers' interpersonal skills, subjectivity, and research skills necessarily affect and limit the outcomes. Qualitative research is "saturated with more concealed forms of power than quantitative and experimental research" (Brinkmann, 2018, p. 589); therefore, qualitative researchers must be especially aware of power relations and their influence on the research project. In this section, I discuss the limitations of the data collection methods that I used as well as the limitations of reflexive thematic analysis.

As a frame of reference, it's important to recognize that "[t]hematic analysis – and qualitative research more generally – is never definitive. Findings remain always tentative and emergent. There will always be more that could be said." (Finlay, 2021, p. 114) This is not a limitation to be mitigated, although is it sometimes interpreted as such. Instead, it is the intended outcome of interpretive qualitative research.

The students who volunteered to participate in this research 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. Further, my findings are dependent on my ability to build trust, maintain relationships, respect norms of reciprocity, and sensitively consider ethical issues (Marshall and Rossman, 2016) while interacting with the study participants. Another researcher may have elicited different responses from participants, but my particular perspective is a resource that I bring to this project.

3.7.1 Limitations of Interviews

While the qualitative research interview is an effective method of data collection, it is "not an unproblematic, direct, and universal source of knowledge, but is rather a historical arrangement that builds upon and reinforces a number of ideological constructs, some of which might be problematic" (Brinkmann, 2018, p. 591). Every interview, as a social interaction, is embedded in its social context. This social context may have influenced and limited the information that was shared in the interviews. As with any data collection method, the limitations of interviews must be carefully considered and understood.

As with all qualitative research, the findings from interviews are limited by the interpretation of the data. Fontana and Frey (1994) state that "data must be interpreted and the researcher has a great deal of influence on what part of the data will be reported and how it will be reported" (p. 370). Further, data can be "problematic, and, at times, contradictory" (Fontana and Frey, 1994, p. 372), which implies that the researcher may need to reconcile those contradictions. Problematic or contradictory data, when considered alongside the researcher's subjectivity, mean that different people conducting the analysis could come to different conclusions. The "tremendous, if unspoken, influence of the researcher" (Fontana and Frey, 1994, p. 372) must be considered by the reader, and ideally, explicated by the researcher as transparently as possible when they disseminate the findings. In my study, I have attempted to share the contradictory findings and highlight the multiple realities shared by my participants.

Interview research is limited by the interviewer's skills; every aspect of the research study is affected by the researcher's abilities. Interviewers need "superb listening skills (or language skills, e.g., in local languages, symbols, abbreviations, or signs)" (Marshall and Rossman, 2016, p. 151) and they must be "skillful at personal interaction, question framing, and gentle probing for elaboration" (Marshall and Rossman, 2016, p. 151). An interviewer's ability to listen will affect their ability to understand the interviewee, guide the interview, and ask follow up questions.

I believe that I am an active and empathetic listener and am able to understand and interpret participants' intended meaning from their responses. However, as a novice interviewer, I am continually enhancing my ability to offer adequate context for my questions and consistently pose probing follow-up inquiries.

Every interview, as a social interaction, is embedded in its social context. This social context influences and limits the information that is shared in the interview. For example, Seidman (2019) states that if the interviewers and interviewees have different genders, the results will be different than if they were of the same gender. It's probable that my interview data depended on many social factors of myself and the interviewee including and beyond our genders.

Interviewing, as any social situation, is influenced by power dynamics between participants. Qualitative researchers must be especially aware of power relations and their influence on the research project. The interviewer has significantly more power in an interviewing relationship than the interviewee as they plan and run the interview, ask the questions, use the results to meet their research goals, and have control over data interpretation (Brinkmann, 2018; Seidman, 2019). The interviewee does have some potential control in that they can choose to evade questions or elide information (Brinkmann, 2018), but this is far less power than the interviewer holds.

The power relationship may be further influenced by societal inequities. Interviewers must be "sensitive to the way these issues may be affecting the participants" (Seidman, 2019, p. 105) at all stages of the research project. Interview research is serving a researcher's goals and is fraught with potential exploitation of interviewees (Seidman, 2019). Seidman (2019) states that "it's a constant struggle to make the research process equitable" (p. 13), and interviewers must maintain a focus on equity throughout the study. If the interviewer is unable to provide an equitable research environment, due to their own unwillingness or because the inequities are beyond their ability to resolve, the interviewee's desire to share their authentic story or the interviewer's capacity to learn from and understand the interviewee could be limited.

The particular social context in which this research was conducted frames the findings. I have been transparent in my positionality and my research design in order to bring my subjectivity to this project as an asset rather than as a deficit that should be mitigated. I believe it's important to consider the limitations of interviews in tandem with the idea that the particular context of this project is an asset; the knowledge generated is directly influenced by me, my participants, and the way that we interacted in the study.

3.7.2 Limitations of Reflective Journaling

Reflective journaling has limitations in common with interviews and other qualitative methods. Journals are written in relationship between participants and researcher, and must be considered within this social context. The self-reported nature of journal entries could be considered a limitation by some (Phelps, 2005), but with my interpretive perspective, I am attempting to understand reality "as [they] understood both [their] own experience and [their] subjects' portrayal of theirs" (Charmaz, 2000, p. 523). Phelps (2005) states that "[r]eflective journals are an ideal tool to document this shared reality." (p. 49)

Power relations between the researcher and participant may limit the participant's level of comfort in sharing their reflections and may pose risk or perceived risk to the participant (Paterson, 1995). There may be social inequities that lead to power imbalance, but even when this is not the case, the researcher inherently holds more power in the relationship. The participant may consciously or subconsciously attempt to write what they think the researcher wants to read (Moon, 2013; Phelps, 2005). Further, we cannot assume that all participants are comfortable sharing their personal experiences (Rodriguez, 2017). Participants may worry about being judged negatively on their reflections, or they may be less confident in their writing or language skills (Hayman et al., 2012). When we ask participants to write honest reflections that may leave them feeling vulnerable, we are asking them to take a risk (Greiman and Covington, 2007). We are asking them to trust the researcher with their reflections. In the consent form I was clear that quotes from the reflective journals may be made public (Hubbs and Brand, 2005; Moon, 2013); these plans to include quotes in a published research article may limit a participant's willingness to share honest reflections.

The reflective journal responses were also limited by participants' capacity to meet the expectations for frequency and number of journal entries (Janesick, 1999).

Two participants did not complete the reflective journal entries and others did not complete them weekly and instead batched them and completed more than one entry at a time. Participants' discipline, motivation to write, or available time may have limited their ability to meet these expectations. Further, the limited number of 12 journal entries that I could ask participants to write constrained the number of topics or prompts that could be delivered, thereby limiting the data that was generated via the journals (Lutz and Paretti, 2019).

Participants' responses were limited by their ability to think reflectively; Epp (2008) found that "learning to write reflectively was a learned skill that developed over time" (p. 1386) and not all individuals will develop this skill at the same rate. I attempted to mitigate this limitation by gradually increasing the reflective nature of the prompts, which gave participants time to develop their reflective skills (Moon, 2013). Even with growth in this skill over the study, individuals' capability to reflect varies; not all participants will have the same level of reflective thinking skill (Moon, 2013). In addition to varied reflective thinking skills, contextual factors will also affect the quality, honesty, and vulnerability of the reflections (Moon, 2013). The participants in this study varied in their abilities to think reflectively. Most showed evidence of deep reflection by providing extensive, nuanced answers, and others' reflective responses ranged in depth.

Reflective journals are "not just about gathering data but of actively prompting change, by challenging individuals to reflect on new ideas, concepts and theories and to engage in action" (Phelps, 2005, p. 43). This personal growth and reflection affected and changed the situation that I was trying to understand. This study provided a learning experience for participants who were able to reflect on and learn from their technical writing experiences. The findings from this study represent participants' beliefs and understandings during a four-month data collection window and reflect learning that occurred during this time.

3.7.3 Limitations of Reflexive Thematic Analysis

Reflexive thematic analysis is a flexible approach to qualitative data analysis (Braun and Clarke, 2019); the researcher must make choices about how to enact reflexive thematic analysis in a way that is consistent with their theoretical beliefs and their research questions. Its flexibility could be interpreted as a weakness, however, when approached as Braun and Clarke (2019) intended, it need not be. Braun and Clarke's (2019) approach "require[s] reflexivity, theoretical knowingness and transparency" (p. 592) that justifies a researcher's methodological choices. I have provided a theoretically-informed and transparent description of my use of reflexive thematic analysis that will allow a reader to understand how I generated the findings of this study.

The particular lens and personal subjectivity that I brought to this research could be viewed as a limitation or a resource, depending on a reader's theoretical position. As Braun and Clarke (2019) state "[a]ssumptions and positionings are always part of qualitative research" (p. 595). I have made my assumptions and positionings clear throughout this dissertation and view my subjectivity as a resource in this project.

3.7.4 Limitations in the Context of this Study

I have discussed the limitations of interviewing and reflective journaling as data collection methods and of reflexive thematic analysis in the context of a qualitative study. Interviews allow deep exploration of a topic and lead to rich, detailed data that "allow the researcher to understand the meanings that everyday activities hold for people" (Marshall and Rossman, 2016, p. 150). Reflective journals provide an opportunity to gather data about participants' experiences, thoughts, opinions, and reflections (Paterson, 1995) that may be difficult to collect via other methods. Reflexive thematic analysis is an analysis method that is transparent about its foregrounding of the researcher's subjectivity.

No research method is without limitations; what is important when choosing methods is understanding the limitations, mitigating them when possible, and being transparent about the limitations when disseminating the case study so that readers can come to their own conclusions about the findings. In this dissertation, I have been transparent in my descriptions of my positionality, data collection methods, analyses methods, and findings.

Chapter 4

Results, Discussion, and Implications

4.1 Chapter Overview

This chapter shares the results, discussions, and implications of this study. It is divided into six sections; this section provides an overview and describes the structure of the chapter.

During data analysis, I generated eight themes. Much of the participants' descriptions of technical writing was initially included in the 'Students reflect thoughtfully on their or others' technical writing' theme. I separated these descriptions from the remainder of the analysis in order to present the context of participants' definitions of technical writing and their reflections on their own technical writing before sharing the remaining themes I constructed. Section 4.2 shares participants' and my definitions of technical writing in computer science. Section 4.3 describes participants' reflections on their own technical writing.

Each of Sections 4.4 to 4.6 focuses on a research question and its related themes and contains results, discussion, and implications. The discussions and implications are each relevant to a particular research question, so I placed them immediately following the research question's results rather than in standalone chapters.

The themes, organized by their related research questions, are:

- Research Question 1
  - Students learn technical writing implicitly via examples
  - Students have a keen awareness of audience for technical writing
  - Students reflect thoughtfully on their or others' technical writing
  - Students appreciate diverse community support
  - Students believe that technical writing is a social and collaborative endeavour
- Research Question 2
  - Technical writing is less important than other technical skills in computer science
  - Negative computer science stereotypes affect perspectives about technical writing
- Research Question 3
  - Students have a conflicting desire to learn technical writing

4.2 Definitions of Technical Writing

For this study, I used a broad definition of technical writing in order to capture students' experiences learning about a wide range of writing activities. Before I shared my definition with participants in the initial interviews, I asked them, "Can you tell me what you understand technical writing in computer science to be?", because I was interested in learning how they defined technical writing.

Some of the definitions that participants shared are shown in Table 4.1. Participants often shared examples of technical writing as they discussed its definition. The complete list of examples they mentioned is: formal email, assignments, papers, essays, instructional materials, research papers, manual pages, Java documentation, technical documentation, user documentation, Frequently Asked Questions (FAQs), and support guides. After each participant stated their definition, I shared

Table 4.1: Participant Definitions of Technical Writing

Name	Definition
Adrian	"any writing that appears in a formal or academic setting with the intent to instruct, educate or inquire for information"
David	"formal business writing"
Jason	"writing about computer science for technical people. And writing it in a way that's concise and that you're using the language of the trade."
Nico	"anything in which you communicate what you're doing in computer science, such that it's concise and the other person can understand what it is. So it sort of depends on who your audience is going to be."
Phyllis	"writing about technical subjects or for a technical audience"

my broad definition of technical writing which is "writing to convey technical content". I gave examples of emails, commit messages, reports, user documentation, and issue trackers.

Overall, participants and I had similar definitions for technical writing at the beginning of the study. In general, we all considered broad uses of writing in technical contexts and a wide variety of examples for varied audiences, including student peers, instructors, teaching assistants, users of software that they are developing, managers, and work colleagues.

4.3 Reflections on Technical Writing

Participants reflective depth varied, but they shared their beliefs about what makes technical writing good and their reflections on doing technical writing.

4.3.1 Qualities of Good Technical Writing

Participants raised several key qualities of technical writing. Across participants, they agreed that good technical writing is concise, organized, and easily understood. Some participants felt that good technical writing should be interesting or unique in order to avoid boring the reader.

Good Technical Writing is Concise

Participants uniformly described good technical writing as concise. Appreciation for concise technical writing was expressed across contexts, such as in course work, in documentation, and in other technical documents. When asked about good technical writing, conciseness was the first quality that many participants mentioned, for example, "I'd rather have technical writing that is concise, straight to the point" (Nadir) and "I feel like I'm writing well when I can say what I mean in relatively few words. One of the most important things to me in my writing is being concise." (Jason). David agreed and said, "Write less. A lot less." Wui discussed conciseness as a writing goal that she is still working to achieve: "something that I feel like I haven't learned yet is how to concisely and clearly communicate your ideas in terms of design, architecture, and software decisions, especially in a team setting." Krish emphasized the value of concise and clear technical writing: "[i]f the idea is conveyed in a concise manner and in a clear manner, then that is pretty good technical writing".

Good Technical Writing is Organized

Participants consistently expressed that good technical writing is organized and well-structured. This sentiment was expressed across contexts. Phyllis and Wui both focused on structure: "I think the structure is very important" (Phyllis) and "the ability to structure is very useful in technical writing" (Wui). Victor didn't use the word structure but shared a similar thought: "we need some systematic way of explaining what's going on in the document."

Good Technical Writing is Easily Understood

Participants also consistently expressed that good technical writing should be easily understood. Nadir and David draw from their perspective as a reader to illustrate their view, when they said "if I don't have to re-read it to understand the main objective, it's a well written document" (Nadir) and "I understand ... the purpose ... within a few moments" (David). Etson and Nico focused on ease of understanding in their comments; "the way I choose ... suitable material for me to ... learn from is ... I can fully understand from this material" (Etson) and "[i]t's easy to read or understand and can still communicate the message clearly" (Nico).

Good Technical Writing is not Boring

Some participants noted that they try to keep their technical writing unique and avoid boring the reader. For example, Nico said, "I tried to make my writing not too boring, but I don't think that's necessarily a requirement for a lot of technical writing". In discussing technical writing that he has found useful as a reader, Jason mentioned his appreciation of a colloquial style:

> it doesn't have to be 100 percent serious and sometimes being a little bit more colloquial can be really, really effective in helping to convey something that is maybe, actually, really technical (Jason)

4.3.2 Students Reflect Thoughtfully on Their or Others' Technical Writing

Participants shared reflections on doing technical writing. In this section, I present findings related to students' reflections on their individual writing. In Section 4.4.1, I present the remainder of the findings from this theme.

Participants were generally confident in their writing abilities and also found technical writing to be difficult. Many participants found that the act of writing helped them organize their thoughts. I describe these reflections in more detail in the following sub-sections.

Participants are Confident in Their Technical Writing Abilities

Most participants expressed confidence in their writing abilities, although their confidence varied across contexts. Most were fairly confident in their general writing abilities even though some were less confident in their technical writing abilities. Victor, Jason, and David all expressed their confidence: "I think of myself as a strong writer, and I think I've always thought that about myself" (Victor), "don't have any concrete evidence of that other than just my own confidence in my ability" (Jason), and "I'm really good at communication" (David). Krish felt confident in his abilities to write in English as a non-native speaker; "while writing

I have a good enough vocabulary to present my ideas well". Adrian shared situations in which their communication skills have helped them: "also being able to write and speak well have all been strengths that have helped me in other spaces like interviews."

The participants uniformly expressed a growth mindset (Dweck, 2006) with relation to technical writing. They all believed that it was a skill that they could improve.

Technical Writing is Difficult

Participants consistently described technical writing as difficult and discussed different areas of difficulty. Adrian found it difficult to explain abstract computer science concepts in writing. Nico commented more generally that writing is difficult for him. Victor focused on co-writing and trying to bring multiple ideas together, he said, "stitching all that together is hard, I think, especially when you're writing with multiple people on the same document." Etson commented that he finds it challenging to "write out an explanation or a proof in a way that I think is very clear and concise."

When discussing the difficulty of technical writing, participants often described accompanying emotions, such as frustration and confusion. For example, Wui was frustrated when she wasn't sure what was expected of her: "I felt quite annoyed and a little bit frustrated because we didn't know how much we should explain or what was too detailed and what is too vague". Phyllis expressed similar emotions:

> I was pretty frustrated and confused during most of the writing. I usually am just because I stress a lot about how well it's structured and how well the message is coming across. (Phyllis)

Technical Writing Helps Organize Thoughts

Some participants described the technical writing process as one that helped them organize their thoughts or verify their written answers. For example, Nadir spoke about writing serving to solidify ideas and uncover disconnections in his argumentation: "with any kind of problem ... once you start writing it ... you start seeing that then your main point also has flaws in it". Victor specified that technical writing helps him conceptualize ideas: "the ways technical writing can actually shape the way you conceptualize different ideas and the way you can communicate them". Jason indicated that the writing process boosts his confidence in his solutions: "I usually don't fully believe that I've gotten it right until I have written it out and then look at my answer written out next to the problem".

4.4 Finding 1: Computer Science Students' Communities Influence Their Learning of Technical Writing

This is the first of three sections that explores findings related to one of the research questions. Here, I discuss Research Question 1: "How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?"

When I planned the interview questions and reflective journal prompts, I emphasized questions and prompts that would lead to participants' discussion of their perspectives and experiences related to learning technical writing *in community*. I consider Lave and Wenger's Situated Learning Theory and Social Theory of Learning (Lave, 2009; Lave and Wenger, 1991; Wenger, 2009) and their notions of legitimate peripheral participation (Lave and Wenger, 1991) and communities of practice (Lave, 1991) when discussing how computer science students' communities influence their learning of technical writing. These theories are described in detail in Section 2.1.3.

4.4.1 Research Question 1: Results

Computer science students' communities of practice influence their learning of technical writing in many and varied ways. The study participants' experiences and perceptions of their experiences were not homogeneous; in this section I share the themes I constructed in relation to considering how students' communities influence their learning of technical writing. The five themes related to this research question are "Students learn technical writing implicitly via examples", "Students have a keen awareness of audience for technical writing", "Students reflect thoughtfully on their or others' technical writing", "Students appreciate diverse community support", and "Students believe that technical writing is a social and collaborative endeavour".

Students Learn Technical Writing Implicitly via Examples

Study participants learn technical writing implicitly; they evaluate their own and others' technical writing, determine the strengths and weaknesses of the writing, and emulate the strengths in their own writing practice. Across participants and across contexts, this implicit evaluation and emulation was consistent. Participants were aware of the different contexts for their writing, for example, in a particular course, or at a particular job, and would look to identify appropriate writing samples for the different contexts.

Instructors' Expectations are Unstated or Unclear In coursework, technical writing expectations were often not stated, leaving participants unsure of their instructors' expectations. Nadir discussed an example and said, "it always seemed like the students were either supposed to already know how to make a design document or figure it out while writing it." This sentiment was shared by other participants who felt like they were assigned writing tasks but were not taught how to write the documents. For example, Nico stated, "I didn't receive any formal instruction on how to do this kind of technical writing."

Participants appreciated opportunities to practice technical writing, but overall didn't believe there were enough opportunities to practice. Victor felt like the practice opportunities on their own weren't sufficient - that some instruction was missing: "the curriculum gives you a lot of opportunities to practice this like practice speaking, practice communicating, but you're not taught how to do it". Phyllis echoed this feeling when she said, "it's like one of those skills that are important and you have to use a lot, but nobody really bothers to teach it.

When instructions were given, participants sometimes felt that they were too vague to be understood: "instructors always try to steer us in the right direction by saying write clearly and thoroughly but I would like to have more context" (David).

Participants would prefer more explicit instruction; this sentiment was universal across the participants. Victor expressed awareness of the pedagogical intent behind vague instructions, but believed that further instructions would be helpful. When discussing a particular course project, he said, "they give you very sparse instructions for a lot of the milestones on this project. Supposedly, it's for your own learning good, but I think it's just like under-baked, slightly." Jason also expressed his preference for more explicit instruction: "we don't get taught, in my experience, I haven't been taught anything about technical writing explicitly in my courses. I do think it's important."

Some participants suggested resources that they would like to have in their courses. David "would really appreciate if they gave us guidance and examples of technical writing" and Nadir suggested "an outline of what the students are expected to have in their writing. This could be a brief section in a syllabus or extra resources." Phyllis would have liked to have access to an example when she was working on an analysis section of a report with a partner. She said, "I also disliked that we were not given an example of what a good analysis section might look like, so neither of us were certain our approach was correct." Adrian wished for guidelines to follow in their technical writing: "a set of steps to always follow it becomes a lot easier ... Whereas, otherwise, it can get a little confusing."

Phyllis expressed a desire for examples and less-abstract descriptions of what good writing looks like. She found it difficult to determine whether her writing met abstract criteria like 'well structured' with 'arguments linked in a logical way'.

Victor mentioned that it might benefit more people, and in particular the people who have less experience or less interest in technical writing, if more explicit instructions were given. In discussing people who are less comfortable communicating, he said: "if this is taught explicitly maybe more individuals could benefit from the opportunities".

Students Find Lack of Explicit Instruction Stressful Some found a lack of explicit instruction stressful, especially for course assignments. In courses, participants felt confused when they weren't given explicit instructions or examples for writing they had to complete. Phyllis and Nadir describe feeling confused and lost: "the guidelines for that report also were not particularly clear ... so I think a lot of people were confused about it when I got to office hours" (Phyllis) and "[w]hen it came to writing the user studies, I felt quite lost. I was pretty familiar with writing research questions for life sciences, but writing research questions for a software project was very foreign" (Nadir).

This was less of a concern at work or in smaller academic groups because, in general, participants were more able to ask clarifying questions and ask for templates from their managers, advisors, or colleagues. Victor will ask for examples when he needs to do a new kind of technical writing. He recalled a situation when he was preparing a presentation for a specific conference, he asked his professor for examples and explained that he wanted to "emulate their style and see what I like about their style? Mix and match, remix. I definitely am a work by example person."

Phyllis was able to find past examples at work: "I remember being very unsure about how much detail to go into, so I looked through closed PRs[1] for examples of past interns' work and tried to match their style."

However, even at work, sometimes participants felt that expectations were unclear which caused them stress. Wui encountered situations in which she was unsure about how much detail her documents should include. She "felt stressed because I didn't know how much technical knowledge the next person carrying my project might have, and so I felt a lot of weight ... to make sure it was well documented."

Emulating Examples The practice of emulating examples of good technical writing, and sometimes good writing in general, was discussed often. For example, Victor "grab[s] a lot of writing inspiration from reading other people's writing." Participants were generally reflective about what they were reading and emulated a variety of aspects of others' writing, including sentence structure, higher-level structure, vocabulary, level of detail, the inclusion of figures, style, and tone.

Nico learns from more senior academics about how to write and tries to emulate their notation. He thinks "the people working in academia at the moment know more than I do. So more often than not, I'll try to bring my work up to the current standard than try to invent my own process."

Some participants noted that while they learned from community members and

[1]PR is an abbreviation for pull request.

tried to emulate elements of community members' writing, it is difficult to emulate. It is challenging to generate abstract "rules" for writing from examples and challenging to apply those "rules" to a writing task. Victor described these challenges when he said, "there are no rules, right? I mean, maybe there are, but they're hard to explain" and "[c]ombining all the different ways people explained concepts was another challenge, but at least it gave me a foundation for tooling my own reports."

Learning Community Communicative Norms Participants were intentionally learning about and reflecting on community communicative norms, both in academic and work settings. Their quotes indicate that they are implicitly learning about communicative norms, attempting to infer the norms from reading documents and interactions with community members. Adrian attempts to understand community norms and goals; they said, "most people have an idea of what they want, even if it's not explicit." Nadir focused on collaboration when he said that he "tr[ies] to collaborate and see how your team members are approaching the task and I don't think I've ever asked anyone, what are your goals for technical writing? Rather you kind of interpret them implicitly through their writing." Victor suggested "adapt[ing] to other people's technical writing styles because that's going to pay a major dividend in the future." David's mentor had stated that he doesn't spend a lot of his time writing code; David inferred that his mentor must be doing a lot of technical writing and was planning to ask his mentor for more information. He wondered "what he [a mentor] writes all day and what he does all day. And so I'm going to continue meeting him ... he must be doing lots of writing reports and things like that."

Krish's approach to learning communicative norms involved learning from existing documents as well as explicitly asking questions about how the documents were created. He suggested two ways that he learns community norms: "one way would be to review the existing technical writing and how it's written down, but another way would be to ask questions about what already exists and how the procedure works."

Wui noticed which team members were doing technical writing; the fact that senior engineers were doing technical writing was meaningful to her because she didn't think it was common. She said that the stereotype is that the product manager writes documentation and thought "whoa" when she noticed a senior engineer working on documentation.

Explicit Feedback was Rarely Received Participants rarely received explicit feedback on their writing and instead tried to interpret community members' reactions as a form of feedback. Etson noticed when there are few follow up questions asked about his written documents and interpreted that as positive feedback that his writing was understood. As an example of interpreting lack of follow up questions positively, he said, "when ... I distribute to all departments in the company ... and not many people get back to me to ask about what is the meaning of this step or what is the meaning of this diagram." However, as Wui describes, it's difficult to be confident in how to interpret a lack of feedback. She wondered, "if there are no questions, then maybe they either don't understand or it was very clear, which one?"

Technical Writing is not Salient as a Work-Related Task When discussing tasks that they performed at computer science-related jobs, no participant mentioned a writing task until further prompted, although all nine who had technical work experience had done some small or large writing tasks at their jobs. Technical writing is integral to all the technical jobs that participants had experienced, but did not immediately come to mind as a task that they completed at work. This below-conscious view of technical writing is another indication that it is implicit, unconsidered, and perhaps often unexamined. When prompted, however, participants quickly listed and discussed the technical writing tasks that they had completed.

Variations in Participants' Responses Participants were quite consistent in expressing that they learned technical writing implicitly in their computer science courses (and in their tech-related jobs) and that they would prefer to have more explicit instruction.

There was variation in the specific writing skills that they discussed. Some were focused on vocabulary and sentence structure, some were more focused on document structure, and some discussed varying contexts.

In some cases, participants had received explicit instruction for a specific assignment or in a specific course and those participants expressed appreciation for the instructions. For example, Krish appreciated when "the requirements were laid out pretty clearly so it wasn't that hard" and Wui found "[i]t was also a positive experience to have the opportunity to setup a meeting with the TA and get feedback on how to improve our documentation or planning and clarify the things that they were confused about."

Students Have a Keen Awareness of Audience for Technical Writing

Participants clearly, consistently, and uniformly expressed an understanding that the intended audience is important in technical writing. Adrian stated that "your writing changes based on who you're writing to or what ways you want to write" and Jason was similarly focused on his audience: "I try to really focus on approaching it from like the perspective of whoever is going to read it". Nadir and Nico had specific audiences in mind when they said, "it is very different; communication between the developers should be concise and to the point. Between users should be more simple, simplified in a sense." (Nadir) and "one is going to be geared towards a math audience and the other one's going to be geared towards a ML audience." (Nico)

Participants were aware that any document may be targeting a single audience (e.g., a teammate) or multiple audiences (e.g., a non-technical manager and a teammate, or a teaching assistant and a teammate). For example, Nadir said, "[t]here could be people who only understand the frontend, there could be people who only understand backend. So the design documents should be able to be understandable by both parties."

Participants expressed that writing tasks were difficult when they did not have a clear understanding of their audience. In these cases, they were unsure about what the audience would need from their writing. Phyllis wondered how her audience would understand her writing: "I didn't have a handle on how much an audience completely unfamiliar with it would understand my descriptions" while Victor asked himself how he could explain something to an audience who lacked a deep technical understanding of how a system worked:

> if I want to explain how a file system works to somebody so they can understand, how come I lost my file, or didn't recover something. It's like, how do I explain this to you without explaining how assembly language works under the hood? (Victor)

Wui was unsure about what a project manager would want from her writing as their work is very different from hers. She found it hard to put herself "in the shoes of someone who's managing multi-million dollar projects, what they would think of my technical writing?"

Students Reflect Thoughtfully on Their or Others' Technical Writing

The participants in the study were reflective about their and others' technical writing. Partial findings from this theme were presented in Section 4.3.2; the remainder of the findings are presented here.

The Genre of Technical Writing is Important Participants demonstrated an understanding of genre, where genre means category of writing characterized by the style, form, or content (Merriam-Webster, 2023). Both Nadir and Victor discussed how writing differs between academic areas, for example literature or the humanities and technical writing. Nadir discussed the goal of each genre: "there is a difference between general writing, for example, in literature, vs technical writing. They're trying to achieve two different things", whereas, Victor discussed how the way he approaches the writing varies by genre: "the way you write documentation is so different from the way you would write a short story, an essay, or an opinion piece."

Nico and Victor have some research experience and both discussed the genre of research papers. Nico mentioned his expectation of a research papers readers' knowledge: "[i]f you're writing research papers for people in the computer science community, you would expect that they already have some knowledge of the current notations going on in the community".

Victor was even more specific, discussing the need to learn how to write a research paper for a specific sub-field of computer science. He has research experience in two computer science subfields and mentioned both. When discussing his undergraduate student peers from the systems group, he said, “they’re also interested in learning how to write better, how to write a systems paper”. In relation to his Human-Computer Interaction (HCI) work, his focus was on positionality and learning about the genre from his mentors: “when you’re doing ... HCI work, or stuff that requires thematic analysis, ..., a big part of it is your positionality as a writer. And so I talk a lot with ... my prof and my postdoc about my writing style.”

Phyllis discussed computer-science specific genres when she compared the genre of writing for GitHub with the genre of a report or paper. She said, “if you’re in a GitHub thread, you don’t necessarily need a perfect document structure but for something a bit more formal, like a report or some kind of paper, I think the structure is very important”.

Communicating to Non-Computer Scientists is Important Participants recognized the importance of communicating to non-technical people. For example, Jason said, “being able to explain what you’re doing to non-technical people is extremely valuable”. Participants also believed that it is more challenging to write for a non-technical audience. For example, Nadir finds that “writing for project members who don’t have any experience in the field might be a lot harder.” Krish and Victor expressed similar sentiments and described some difficulties in detail when they said, “it would definitely be harder to write for someone with a non-technical background to explain the concept to them clearly because you’d have to use less technical words or ... avoid technical jargon related to that team” (Krish) and “to ... my peers, who are very smart, capable people. But I don’t think [they] have the exposure to what happens underneath the hood to realize that this isn’t magic, this is human built technology with all the human flaws” (Victor).

Reflection on their Experiences in this Research Study The participants reflected on their participation in this study and recognized that their participation likely influenced their experiences of learning technical writing. For example, Krish said, “as I did the whole study, I felt the importance for it, and it made me pay particular attention to when I was doing it and when I wasn’t. So that was also an educative experience for me.” Nadir felt that the study helped him understand different genres of writing: “[Before the study it] just felt like technical writing was a subset of humanities, but now it feels more, there’s not a divide but rather they’re two separate fields”. Phyllis shared that her participation in the study helped her notice misconceptions and made her aware that writing is difficult enough to be worth of study: “it’s helped normalize the fact that writing is difficult enough to warrant people to come and see how it can better be taught.”

Participants were regularly asked to reflect on their experiences, which raised their overall awareness of the ways in which they were learning and practicing technical writing. For example, the reflection made Jason more aware of the writing he did: “writing the journal entries made me more cognizant of the writing I was doing throughout the semester” and Victor appreciated the opportunity to reflect on his writing practice: “I think it’s let me reflect on a lot of things about how I learned ... it’s nice to be in a study where I’m forced to be reflective.”

Some expressed that the mere existence of the study indicated that technical writing is an important part of computer science. For example, David feels “much more validated that technical writing is very important, even though it might not be highlighted in the particular courses I’ve taken.”

Students Appreciate Diverse Community Support

There was broad evidence that the participants valued writing with and receiving feedback from people who have different perspectives than they do. Jason explicitly mentioned his appreciation for working with others who have different perspectives: “it’s easier to write with other people who are, I think, different from you is too broad, but who can provide an additional perspective”.

Participants also valued working with collaborators who have different work and academic backgrounds than themselves. Wui mentioned appreciating feedback from business and technical perspectives: “having someone to look over your writing, both from a technical point of view and a business point of view, would be beneficial.”

Participants Valued Having a Support Network Many participants expressed that they valued having a support network in their communities. They appreciated knowing that others want them to succeed and being able to ask for feedback. David said, "everybody really wants everyone to succeed and they're there to help, even the little steps. And I thought that was really reassuring". Etson discussed community support from peers in his academic program; he finds his cohort friendly and willing to provide quick advice and insight when he asks questions.

Victor appreciated a mentor's support when she looked over his resume: "really thankful to have so many avenues for guidance otherwise I would probably be swimming in doubt about what to include."

Participants' Experiences Collaborating with People who are English Language Learners Most participants expressed positive feelings about working with collaborators who are English Language Learners (ELLs). Victor shared that working with ELL students as a TA helps him learn how to communicate more effectively with all of his students. Once he is able to explain something to a student who is an English Language Learner, he believes he has a better explanation to use with all students. He thinks that "working with the English language learners has made my teaching much, much better in a way that I would never have forced myself to do if I was just speaking with native speakers."

Some participants believed that working with non-native English speakers is a benefit because the non-native speakers may write in less complex ways, which can make the writing easier to understand. As one example, Nadir finds that ELL students "often they try to write things a bit more simply" and "that's good because it makes the writing easier to understand".

When working with English Language Learners, most participants focused on whether they could understand their colleagues' writing rather than on grammatical correctness. They understand that writing is about meaning making (Byrnes, 2013). For example, Adrian said, "[y]ou learn the means of communication that are most necessary or useful to the highest of utility". In the context of his TA work, David expressed that he should "judge them [students] on the content of their message" and "don't even mark them down for spelling mistakes or grammar or things like that, because that's not what we're looking for". Nadir and Jason agreed: "it might not be as grammatically correct, but I can still understand what they are trying to say" (Nadir) and "[e]ven if they didn't have the strongest English skills, their qualification for their jobs meant that they could still effectively communicate and understand correspondence regarding any technical aspects of the work" (Jason). Phyllis believes she has a coworker who is a non-native English speaker. She said, "their writing sometimes has grammatical mistakes, but it's still readable, so I don't feel the need to correct them." Victor discussed his experience writing with a research professor and a postdoctoral fellow who are non-native English speakers: "I don't think this hampered their ability to write technical English too significantly because they were both academics, save the minor grammatical issues that I often fixed in our manuscripts."

Although most participants focused on meaning making, multiple participants expressed an awareness of negative bias towards English Language Learners and how their writing is perceived. Adrian felt "there is very much like a slight bias in how we approach people based off of their English writing ability" and Nico agreed: "mostly I did the editing because they wouldn't know how to edit properly or their edits would just have bad grammar." David learned more about his own unconscious biases when he worked as a teaching assistant and described his growing awareness of his biases when working with a student who was an English Language Learner: "I guess I have less patience, less leniency, and judge them harsher based on their technical writing ... that is unfair."

Adrian is not a native English speaker but speaks English fluently. They noticed that their language and communication skills affect the way that they are treated by others. Their ideas are readily accepted by others and they believe that is because of their fluency with English.

> I think the fact that I don't speak with an accent in itself benefits me in a way that isn't a good benefit. I'm not happy about it. But it does. Stuff like that, I've noticed, helps me a lot more because I speak so fluidly, quickly and concisely and my pronunciation, most people don't even question half the things I say. (Adrian)

Participants are Open to Feedback Participants valued receiving direct and explicit feedback from peers and mentors, although, as discussed above, they do not feel they receive explicit feedback often. They were welcoming of feedback, appreciated that feedback made their writing better, and demonstrated an openness to learning from others. When discussing feedback, Adrian said "[t]hat's really helpful; these tiny technical things that you don't notice that might make a world of difference." Both David and Jason explicitly discussed their openness to feedback. David believed that his previous experience receiving feedback at work increased his comfort level: "with my background, I realize, I'm much more comfortable receiving hard feedback and providing feedback because of my work as a newswriter." Jason described his openness to feedback as an avenue to improve his work quickly: "the attitude of just being prepared to be wrong or like open to being corrected by anybody was really helpful for just fixing things fast without getting too stuck or afraid of getting feedback."

The writing tasks in which participants appreciated receiving feedback varied from small tasks such as writing a commit message to large, complex tasks like writing a research paper. Nadir discussed a small writing task when he said: "let's say my commit message is not good so my supervisor tells me that I can do it in this way." Krish was similarly appreciative for feedback he received on a longer document: "I received feedback that I should add more visual aids to my documents such as images and graphs to make it easier for the reader to understand the procedures to be followed." Victor mentioned that feedback he received on a research paper improved his writing much more than anything he learned in a course.

Participants were reflective on the feedback they received and were able to see how it impacted their growth as writers. This was particularly evidenced in Victor and Phyllis' quotes. Victor found that feedback helped improve his writing immensely: "having my professor and my postdocs' insight into how to write for that kind of audience and write in that style was really important and helped me improve my academic writing tenfold." Phyllis described a situation in which she and her project partner planned a meeting to discuss the feedback they received. This allowed them to feel more confident in their understanding of the TAs' feedback. She said, "[m]y partner and I scheduled a retrospective meeting to discuss our feedback and which parts weren't done well, which helped me understand what the TA meant."

Students Believe that Technical Writing is a Social and Collaborative Endeavour

Participants commonly discussed the social and collaborative nature of software development in general and the technical writing that is involved in software development. Victor articulated this idea clearly:

> you'd be hard pressed to find a computer scientist who thinks what we're doing is like discovering the laws of physics. Everything we write, our operating systems, hardware, how to structure code, ..., machine learning programs. This is all very much a human endeavour (Victor)

Jason, Nadir, and Victor discussed the necessity of effective communication with team members. Nadir focused on the role communication plays in creating the intended software: "[i]f you don't communicate or get proper communication from other developers, you can't really code what you need to." Victor took the stance that code needs to be understood in order for others to use it: "you can write the most beautiful code in the world, and if no one understands it, no one's going to use it. So it's key to the whole enterprise, I think, because no one really writes software in isolation." Jason discussed the importance of communicating with team members more broadly and believes that communication skills provide value: "99.9% of the time any meaningful software work is going to be done with other people to some extent ... And I think there's actually very limited value in having really good technical skills, but no communication skills."

David noted that companies and teams within the same company vary in communicative culture: "[t]hey all have different culture, and they all seem to speak differently about their work."

Participants Thought the Technical Writing of Others was Poor Participants complained about the quality of their classmates' technical writing, often in relation to group projects. For example, when discussing groupmates' writing, Nico said, "I think the writing is lacking. It could just happen that they have poor English, that happens a lot of times in my experience."

Participants had varied approaches when they thought others' technical writing wasn't good enough. Some chose to redo the writing, for example, Adrian evaluated someone else's draft to be "of such subpar quality" that they asked the author to redo it. When it wasn't re-written, they redid it themself. Phyllis, in one case, chose to prioritize group harmony over pushing for a higher-quality written product. Nadir gave his groupmates some feedback but held back feedback that he believed was beyond the course's expectations for writing: "in that case, I don't really give any feedback ... since they're still hitting the main points of the assignment."

Three teaching-assistant participants found their students' submissions to have poor technical writing. Victor said, "I've been TAing comp sci 320 this semester, which is the algorithms class, and there's some bad technical writing out there". David described a report for a computational thinking course in which he believed students' writing was unnecessarily verbose. Similarly, Adrian had difficulty grading proofs from a first-year course; they said, "I've seen people prove things in insane ways and these are really relatively simple proofs and they've gone a huge circle."

Other forms of technical writing were also described as poorly written. In an operating systems class, students are required to read research papers and one participant commented that they were poorly written. Technical documentation, for example, of a programming language, was mentioned as well. Commit messages were frequently given as an example of technical writing that is done poorly. Participants complained about teammates writing uninformative commit messages and admitted to doing the same, either because they were pressed for time or because they were unsure about what context would be relevant for a reader. For example, Wui said "their commits say 'update' and when you look at the history, they're all 'update'. ... the context, I think, is super important" while Victor described a scenario in which it's difficult to write a useful commit message, and despite knowing that a more complete message is expected, he chose to write a brief message.

> I don't know what information is useful to my teammates. ... And don't want to expend even more brain power thinking about what you need to know about my code. So I'm just going to take the easy way

> out because I have no heuristic framework for even approximating what you might want to know. (Victor)

Some, but not all, participants also described their own technical writing as lacking. For example, Victor knew that he took writing shortcuts and believed that such practices contributed to an atmosphere of poor writing: "I definitely am guilty of cutting corners in my technical writing and not using and putting into practice the things that I've learned, it's easy to learn all these things, but it's hard to put them in practice."

Participants Believe Collaborating on Technical Writing is Difficult Participants discussed many reasons why collaborating on technical writing is difficult. They discussed challenges in co-creating a document when groupmates prefer different styles or tone, or have mismatched expectations for the time or effort to spend on the document.

David and Adrian both mentioned socially-related situations that made collaborating difficult. David discussed language barriers, possibly caused by an age gap or different work experiences. He felt like the only person in his group who didn't understand parts of the writing, and found himself asking questions like, "what are these abbreviations and can you just write it out?" Adrian had difficulty collaborating in a group that made inappropriate comments about their identity. They talked to the professor and were moved to a different group that also had challenges, but they chose to stay with the new group because it was not prejudicial. In their own words,

> I'll take the group where I'm not hearing specific kinds of language or anything about my identity. And then in that group no one really worked, and I was like, I'm just going to take the W[in] here, it's better than the others. (Adrian)

Documentation is Necessary when Collaborating Participants expressed a belief that written documentation and communication were important for software development groups. For example, Adrian said, "if you're coding in a large group, I think all of those development philosophies require some amount of communication and indicating what work has been done" and Etson believes that communication skills are important for tech employees collaborating on large software projects. Phyllis shared this opinion. She believes that writing is important to programmers "as the way the majority of people will interact with and understand the code you've authored".

Krish and Wui both shared a challenging experience when they found documentation lacking. These experiences helped them learn about the value of technical documentation. Krish and a project partner weren't writing comments and found that they got stuck. Wui found it difficult to start working on a project that wasn't documented well. She said, "[i]t was incredibly hard to pick up after someone else's work."

Some said that documentation was important when the size of the collaborating group was large. For example, Nadir found that "documentation is very important whenever you work on any project that will be contributed by more than I would say 20 people."

4.4.2 Research Question 1: Discussion

In this section, I discuss the results of Research Question 1 in relation to the theoretical framework of this dissertation. I focus on how students' experiences learning technical writing were consistent with Lave and Wenger's (1991) Situated Learning Theory. There was varied and substantial evidence to support the idea that participants' learning is socially situated and influenced by their communities' communicative norms. Overall, participants' descriptions of their learning is consistent with learning through legitimate peripheral participation in communities of practice.

As described in Chapter 2, Wenger (1998a) states that a community of practice defines itself via its "joint enterprise as understood and continually renegotiated by its members" (p. 2), mutual engagement, and "shared repertoire of communal resources (routines, sensibilities, artifacts, vocabulary, styles, etc.)" (p. 2). I did not ask participants to identify specific communities of practice, but the way they described their learning in school and work communities meets this definition. A variety of evidence supports that participants' communities had joint enterprises. For example, Victor has been a member of two research groups and discussed how his writing goals differed between groups. Multiple participants implied that teams had joint enterprise when they discussed the teams' communicative goals within a course or company. Mutual engagement was evidenced by participants' discussion of their relationships with peers, instructors, teaching assistants, mentors, managers, and others. For example, Krish discussed asking colleagues about prior writing practices and Wui appreciated feedback she received from community members. Participants described shared repertoires of routines and styles; they learned technical writing implicitly by evaluating the writing of others and emulating the strengths in their own writing.

As discussed in Chapter 2, legitimate peripheral participation describes the process of becoming a "full participant in a sociocultural practice" (Lave and Wenger, 1991, p. 29) and moving from being a newcomer to the community and practice towards being a core, fully contributing member of the community. The study participants described their learning of the sociocultural practice of technical writing. The lack of explicit teaching by experts coupled with the learning that students experienced is an example of newcomers learning through their participation and highlights the "fundamental distinction between learning and intentional instruction" (Lave and Wenger, 1991, p. 40). Participants rarely received explicit feedback on their writing and instead tried to interpret community members' reactions as a form of feedback, demonstrating that learning happens even in an absence of explicit instruction. They were observing communicative norms in each community – for example, understanding a workplace's preference for the style and structure of a report, or a research discipline's conventions – and learning how to become more expert in those communities. Participants were looking to respected community members and learning from their writing practices. In the absence of explicit feedback, participants tried to match their writing to that of the experts in the community whether that was more senior colleagues or academic mentors or instructors. Participants' implicit learning of technical writing is an example of their "negotiation and renegotiation of meaning in the world" (Lave and Wenger, 1991, p. 51); students are attempting to understand and reflect on how technical writing is done in their communities. It's notable that participants wished that they had explicit instruction or concrete examples to learn from, but these were not provided. Technical writing in these communities of practice is not explicitly taught, perhaps because it is a professional skill that is assumed.

Participants' discussions of what they learned from instructors, teaching assistants, student peers, managers, colleagues, and workplace peers is consistent with Lave and Wenger's (1991) notion that learning occurs between experts and newcomers as well as between peers and near peers in communities. Further, the learning that students took from and between different course or workplace communities is consistent with Masika and Jones's (2016) view that communities of practice are fluid and overlapping. Knowledge and learning flows between communities as community members engage with multiple communities or move between communities. As Masika and Jones (2016) describe, there was also learning across and between communities of practice. In particular, students brought what they had learned at a workplace to their experience in their UBC courses. As one example, Nadir learned about writing design documents during his experience with an open source project and applied that learning to his course work.

Students' descriptions of how they learned implied that they viewed each course and each workplace as a separate community of practice. For example, participants discussed communicative norms and expectations on a per-course or per-workplace basis. Participants reflected on differences and similarities between their communities of practice; they often wished that what they learned in one community more directly applied in another but were aware of the different considerations in different contexts.

Participants' keen awareness of audience is consistent with a community mindset. Participants were conscious that they needed to understand who they were writing for and they would tailor their approaches to the audience. Participants' belief that technical writing is a social and collaborative endeavour is further evidence that they view technical writing as a community-based and socially-situated task. Further, participants want to learn about technical writing from community members who have diverse work and academic backgrounds. They see value in learning from diverse perspectives that exist in the community.

In some cases, in particular for participants who have identities that have been historically marginalized or excluded, there was evidence that their identity affected their participation in groupwork. For example, Adrian, a non-binary participant, had their participation in groupwork affected by negative comments that groupmates made about their identity. The negative comments caused them to change groups and although working with the new group wasn't a positive experience either, they chose to remain in the group because at least their identity wasn't being undermined. As another example, students who are English Language Learners can face marginalization in English-language institutions. Adrian and David both acknowledged instances of bias towards English Language Learners. These examples serve as reminders of how identity interacts with the social situatedness of learning communities and how some individuals are affected more than others.

4.4.3 Research Question 1: Implications

In this section, I discuss implications for research and practice in relation to Research Question 1.

Implications for Research

To my knowledge, this is the first study on computer science students' learning of technical writing conducted with the theoretical framing of Lave and Wenger (1991). The findings from this study provide evidence that participants learn technical writing via legitimate peripheral participation in their communities of practice. Further study is warranted to better understand how computer science students learn collaborative, communicative skills in community.

Participants were members of multiple communities of practice at one time and as they moved through their degree or through work experiences their membership in communities of practice was fluid. With participants belonging to workplace and student communities, it would be interesting to explore how and why practices and learning transfer across these communities. Does the learning tend to move in one direction from industry to academia or vice versa? Does the movement of ideas and practices depend on other relationships between communities or between community members?

As discussed in Section 3.6.1, when I conducted member checks, many participants mentioned that their use of ChatGPT (Open AI, 2024) and other Generative AI tools has influenced their technical writing practice. This study was conducted before these AI tools were released; now that the use of these tools is growing in popularity (Prather et al., 2023), future research could explore students' or professionals' use of Generative AI writing tools in relation to learning from community members. How do computer scientists use Generative AI tools in their technical writing practices? What are they learning from these tools? What are they still learning from community members? Has the introduction of Generative AI tools influenced computer scientists' perspectives about learning technical writing?

Implications for Practice

As discussed in Chapter 2, some institutions integrate writing and communicative skills into a cross-section of curriculum while others teach a dedicated course. While both have benefits, the findings of this study support the practice of integrating teaching and learning of technical writing across the curriculum. Participants want to learn technical writing from technical experts. Perhaps more importantly, they are learning technical writing from community members regardless of whether the community members are intending to teach technical writing. Therefore, more attention should be paid to teaching technical writing in computer science courses.

However, it can be challenging for computer science faculty members to teach technical writing. They may not feel that they have appropriate expertise to teach technical writing or other communicative topics. When Victor was discussing teaching writing skills and said, "there are no rules, right? I mean, maybe there are, but they're hard to explain", I believe he gave voice to a common concern among computer science educators; computer science faculty members and teaching assistants are not generally trained to teach writing. A long-term, sustained collaboration with writing experts could allow the integration of teaching technical writing skills into computer science classrooms effectively. It's challenging for non-language experts to teach writing and provide effective feedback, but students want to receive feedback from computer science instructors. Further, students would benefit from learning how to give constructive feedback on peers' technical writing and writing experts could facilitate that learning.

Zhang et al. (2023) share a successful example of embedding a writing specialist into a project focused on teaching technical writing in computer science classes. The writing expert trained teaching assistants who assessed the writing, guided the assessment process, and provided input to computer science course staff about writing pedagogy. It can be challenging to initiate interdisciplinary collaborations, but over time, as collaborators learn from each other, an effective teaching and learning environment can be built.

A sustained collaboration would allow computer science experts to define priority learning outcomes for technical writing and other communicative skills. As Burge et al. (2014) state, research supports "having each individual department define the communication outcomes and pathways for its students, rather than leaving these responsibilities to another department" (p. 580). Communicative priorities vary across disciplines (Burge et al., 2014) so more explicitly embedding teaching of technical writing and other communicative skills into computer science curricula would allow for priority skills to be taught and learned.

4.5 Finding 2: Computer Science Students Hold Perspectives that may be Barriers to Learning Technical Writing

This is the second of three sections that explores findings related to one of the research questions. Here, I discuss Research Question 2: "In what ways do students hold perspectives that may be barriers to their learning of technical writing?"

Student perspectives towards an academic discipline influence their learning of the discipline (Lewis et al., 2010; Perkins et al., 2005). Perkins et al. (2005) studied student perspectives about physics and learning physics in a university setting and found that "[w]hat students think about a discipline—its structure, usefulness, how it is learned—plays an important role in shaping how they approach it." (Lewis et al., 2010, p. 78)

The themes related to students' perspectives about learning technical writing will be helpful in understanding how students approach their learning.

4.5.1 Research Question 2: Results

This section documents the perspectives about learning technical writing in computer science courses contributed by participants during the study. These perspectives were not necessarily held by participants; in some cases participants shared their own perspectives and in other cases they discussed perspectives that they think are held by others.

The themes related to this research question are “Technical writing is less important than other technical skills in computer science” and “Negative computer science stereotypes affect perspectives about technical writing”.

Technical Writing is Less Important than Other Technical Skills in Computer Science

All participants expressed a clear belief that technical writing is important for their careers, regardless of whether their desired career path is in academia or industry. However, some participants shared that the attitude that technical writing was less important and less difficult than other technical skills. The study participants did not necessarily hold these views themselves, but in general, believed that these views were widely held. Adrian stated that there’s a belief that “anyone can write” while it “takes an academic mind to solve math”.

Wui described a belief that people who are technically skilled may not appreciate the need to share their ideas with others: “they’re just so technical that they don’t see the importance of learning how to convey your ideas to people who might not understand it.”

Phyllis interprets a lack of discussion of technical writing as a belief amongst her peers that it’s not that important. Nico and Etson shared similar sentiments; they said there is a heavy focus on technical correctness. For example, Nico said, “there doesn’t seem to be as much emphasis on the writing as getting the correct answer on the assignment” and Etson said, “[f]or the answer is focused on the correct meaning and the correct explanation and not care about the grammar and how student structure the sentence, is all about the correctness.”

Participants interpreted salaries, job ads, and recruiting practices as indicators of how much a company values technical writing or other communicative skills in comparison to other technical skills. Participants believe that salaries are lower for jobs that focus on technical writing. For example, Jason's experience indicates that technical jobs pay significantly more than technical writing jobs and he believes that candidates will be drawn to the higher salaries: "it makes a lot more sense to get a technical job that will pay you like ten times more money".

Some participants discussed a negative attitude in Science, Technology, Engineering, and Mathematics (STEM) or computer science towards skills or knowledge that isn't considered to be highly technical. They discussed a computer science culture in which code is believed to be the most important output. I include a variety of their quotes to fully illustrate the ideas they expressed. Jason noticed that "the most technical people don't tend to put any value on technical writing". David discussed a code-centric focus: "everything is code-centric and we just want to get it to work and everything else icing on the top" and found that "documentation is more like an after thought." Phyllis found a lack of focus on technical writing meaningful: "nobody I've talked to ever focuses on technical writing. Like they don't even mention it, really, it's always about the technical difficulty of whatever they're working on." Victor discussed a lack of attention to technical writing and implied that this is an intentional choice, where people are de-emphasizing technical writing. He stated: "there's a bit in the culture where people want to play down how important communication is in computer science because they think it's detracting from the real stuff, like the hard technical material."

In contrast, some participants expressed that communication and writing skills were as important or more important than programming skills or, more broadly, technical skills. In general, participants believed that technical writing was important, despite their perception, discussed above, that others did not believe it was important. For example, Nadir believes communication is important in team work: "if you have a good communicator and an average coder, that will be a lot more useful in a team setting" and Phyllis thinks communication is key to career success: "learning to communicate effectively is as key to succeeding in this field (and probably life) as technical competence". Jason shared his opinion that a combination of skills can be valuable: "you can be really impactful by having solid technical skills, but also really good writing skills. And I think that you can ... provide a disproportionate amount of value with that skill set."

Writing is More Valued in Other Fields than in Computer Science Participants believed that students learned writing in non-computer science programs and courses at UBC. UBC has a communication requirement for first-year Science students which requires students to take six credits (equates to two courses) of communication courses, chosen from a First Year Seminar in Science, courses in Writing Research and Discourse Studies, or courses in English. Participants enumerated the courses that fulfill the communication requirement, courses in the Cognitive Systems program, and courses in business as courses that teach writing. Two participants said that technical writing was a skill that students were assumed to have learned elsewhere, so it would be practiced but not taught in computer science courses. For example, Nadir said, "None of the classes that I have attended looked closely at technical writing. It was always something that the students should already know about and practice throughout the term." Phyllis found it embarrassing to need to learn writing skills: "I feel like [computer science] professors often expect us to come with writing skills, from high school for example, so expressing a need to be taught how to write is almost a bit embarrassing".

One UBC computer science class requires students to write multiple essays and does not involve programming, proofs, or other non-writing technical tasks. Its focus is on computers and society. David implied that the essay writing in this course, Computer Science 430, indicates that it's not a computer science course. He corrected his original statement, but when he said "430 which is not really computer, which I guess is, it is a computer science course but we mostly just write essays for that one" the underlying question of whether technical writing is considered to be part of computer science is evident.

Both Jason and Phyllis stated that technical writing is not part of computer science content. Jason said, "it's also just sort of outside of the domain of any of the classes that I've taken". In relation to whether technical writing should be assessed in computer science courses, Phyllis said, "[t]he practical challenge is that none of these are directly related to the content of most [computer science] courses, so I could see some complaints of sense of unfairness about being assessed on such skills."

Both Jason and Wui mentioned that other fields place a higher value on technical writing. Jason shared that communication and writing is core to non-STEM fields: “most things outside of STEM, it’s a core, it’s built into the culture that you want to share understandings of things, and that’s established, but to me, it feels like in STEM, that’s not very well established at all”. Wui compared her perception of business-driven companies and STEM. She believes that business-driven companies “value technical writing a lot more because it’s how people communicate” whereas in STEM “they’re mostly focused on code quality or code comments, but not really about how clear documentation should be.”

Participants describe community members’ beliefs that technical writing is not important in computer science. They expressed these beliefs in a variety of ways. Krish focused on computer science instructors’ pedagogical responsibilities when he said, “I think computer science professors or instructors have done a great job of teaching technical knowledge, and helping students improve their writing is not their primary responsibility.” Jason believes that other community members don’t value technical writing; he said, “a lot of people who think that technical writing is like a big waste of time” and “I feel like that’s just something I notice across, even outside of UBC, because the most technical people don’t tend to put any value on technical writing.” Phyllis notes that technical writing is not a focus in computer science courses: “nobody ... pay[s] attention to it.”

Wui expressed that her opinions on technical writing were changing after participating in the reflection prompted by this study when she said, “I’m learning to challenge my beliefs that technical writing is not a [computer science] skill”.

Negative Computer Science Stereotypes Affect Perspectives about Technical Writing

Some participants discussed computer science stereotypes and how they interrelate with community members’ views on technical writing and other communication skills.

Participants were aware of, and sometimes in agreement with, the stereotype that computer scientists cannot communicate well or are uninterested in communicating. Victor commented on the negative stereotype explicitly: “I think that stereotype is always, computer science, engineering adjacent, we don’t know how to communicate properly. And ... sometimes the stereotype holds up.”. Whereas, Krish noticed a disinterest in communication compared to peers from other programs: “but not a lot of them are ... willing to engage in conversation as other majors.”

Victor believed that there was an attitude of disregard towards technical writing; perhaps that some computer scientists want to present as highly technical and remove themselves from demonstrating communicative skills. He said, “[b]ut because the cool thing is, ... I don’t care about these things, to prove that I am more CSy than you”.

In different contexts, both Adrian and Wui discussed the perception that computer science is a “boys’ club”. Adrian discussed the historical patterns of computer science participation and how that created or affected negative mentalities. They noted: “computer science culture, I do think it’s a bit of a boys club ... I definitely think there’s a few negative mentalities here that ... people need to be more open to having change.”

Wui described a situation in which a group of mostly-male students spent a lot of time working together on computer science assignments and she did not feel included in the group. She shared that “there’s a certain group of people who like to grind together; they’re mostly dudes so I don’t really relate. It’s like they just spend every waking hour on [computer science] assignments. I can’t really break into that one, unfortunately.” Her use of the word ‘unfortunately’ implies that she wishes she was able to interact with these peers.

In discussing misogyny in people’s attitudes towards writing, Adrian said, “I think there’s definitely an undertone as to what people are capable of doing or what they’re able to say or be like.” Adrian also mentioned the relationship between systemic and non-systemic issues and how we associate creativity with feminity:

> I think systemic and non-systemic issues interplay each other quite a lot. I think this is one of those things where we see arts or writing as this creative field and we associate creativity to femininity. And then it becomes this weird separation for some reason. (Adrian)

Participants described a divide between STEM fields and more writing-heavy fields such as arts or the humanities. Two participants shared a belief that non-STEM fields are easier or less rigorous. In discussing some of their peers, Adrian focused on their negative attitude towards arts classes when they relayed the statement, "arts classes are easy and this is BS." Krish shared the belief that STEM fields drive innovation and are more rigorous and more difficult to transition to than the arts and humanities.

4.5.2 Research Question 2: Discussion

In this section, I discuss the results of Research Question 2 in relation to the study's theoretical framework and related work that explored student perspectives.

Situated Learning Theory

As described in Section 2.1.3, the Situated Learning Theory (Lave and Wenger, 1991) and the Social Theory of Learning (Wenger, 2009) provided the theoretical framework for this study. Therefore, in my reflexive thematic analysis, I was influenced by these theories and was conscious of the theories when reading and analyzing participants' transcripts. I found evidence that both theories provide a suitable framework for understanding how the study participants' perspectives were influenced by community members and by participants' legitimate peripheral participation with multiple and overlapping communities.

Lave and Wenger (1991) describes situated learning as placing "emphasis on comprehensive understanding involving the whole person rather than "receiving" a body of factual knowledge about the world; on activity in and with the world; and on the view that agent, activity, and the world mutually constitute each other" (p. 33). Participants in this study described the way they learned about technical writing not as the "receiving" of knowledge, but instead as a process that involved themselves, their reflections on their learning, their peers and mentors, and the broader communities in which they study and work. For example, both Wui and Phyllis discussed their peers' beliefs that technical writing is not important and multiple participants shared an opinion that technical writing is not as valued as other technical skills. These opinions were influenced by academic communities; for example, values implied by academic assessments that focus on correctness. While these peers' beliefs did not convince study participants that technical writing is unimportant, the study participants nonetheless have an awareness of experts', peers' and near-peers' perspectives and their behaviours that are influenced by those perspectives. Over time, perspectives in communities of practice evolve as membership changes and as the social situation in which the communities are embedded evolves. Perhaps as computer science becomes even more interdisciplinary and new community members who place higher value on technical writing join communities, these communities' attitudes will change. Since community members "are engaged in the generative process of producing their own future" (Lave and Wenger, 1991, p. 57), the study participants and others who hold similar beliefs are likely to influence their current communities and the ones that they will join in the future.

Although participants' perspectives were learned through interactions with community, they expressed a desire to be the "receivers" of direct instruction about how to do technical writing. This apparent contradiction is interesting. Further research could investigate students' preference for direct instruction.

Participants in this study expressed that learning technical writing will be important in their careers, but few expressed a belief that it would be important for their undergraduate studies. This is another indication that their beliefs about technical writing are nuanced and dependent on context.

Wenger's (2009) Social Theory of Learning is based on four premises, as presented in Section 2.1.3. They are:

> [w]e are social beings;
> [k]nowledge is a matter of competence with respect to valued enterprise;
> [k]nowing is a matter of participating in the pursuit of such enterprises, that is, of active engagement in the world;
> [m]eaning – our ability to experience the world and our engagement with it as meaningful – is ultimately what learning is to produce (p. 210)

Participants perceived negative perspectives towards technical writing in computer science academic and industrial communities. These negative perspectives affect students' perception of the value of technical writing and therefore their ability to make meaning in their work. The study participants held strong beliefs that technical writing was important, but this was contradicted by their interpretations of their communities' values and beliefs. The study participants were high-achieving, reflective individuals who were fairly senior in their undergraduate programs. Given these characteristics, these students may have been particularly well-suited to perceive their experiences with technical writing in community as meaningful, yet, even they expressed contradictory thoughts.

There was evidence that participants' identities influenced their perception of communities. As Lave and Wenger (1991) state, "learning involves the whole person" (p. 53). Seven men, one woman, one non-binary person, and one person who described their gender as "queer; somewhere between woman and non-binary" participated in this study. UBC only has binary gender data available, although as an institution it is improving its demographic data collection practices. The participants' gender profile is reasonably consistent with the gender data that is available for UBC computer science undergraduate students. Each year, the population of computer science undergraduate students is approximately 70% men. Of the 10 study participants, two of the three non-men discussed misogyny and the belief that computer science is a "boys' club" in relation to learning technical writing. Their quotes illustrated an awareness of how the social context affects perspectives about learning technical writing. The absence of similar quotes from male participants supports the idea that an individual's identity is central to their perception of a social situation. In a population where non-male genders are underrepresented, it is important to seek out their experiences and perspectives to ensure that we are learning from a breadth of interpretations of the communities of practice.

Student Perspectives in Computer Science

To the best of my knowledge, there is limited research that investigates computer science students' experiences and perspectives about learning technical writing.

There is a substantial body of research that explores how students' attitudes towards a scientific discipline align with expert attitudes. The Colorado Learning Attitudes about Science Survey (CLASS) was designed and validated for use in Physics (Adams et al., 2005) and has been adapted for Chemistry (Adams et al., 2008), Biology (Semsar et al., 2011), and Computer Science (Elliott Tew et al., 2012). These instruments focus on general attitudes towards a discipline, for example, attitudes towards problem solving in a discipline, and do not include questions about writing or other communication skills. Studies that use these instruments have found that attitudes and learning correlate (Milner-Bolotin et al., 2011; Perkins et al., 2005) or that attitudes influence learning (Cahill et al., 2018; Jeffery et al., 2016). Other scholars have examined student attitudes or experiences in computer science (Hewner and Guzdial, 2008; Lewis et al., 2022; Simon et al., 2013) or programming (Leifheit et al., 2019; Malmi et al., 2020b) but none of these projects considered technical writing.

The perspectives identified by my study provide an initial view on how students may approach their learning of technical writing. Students may be consciously or subconsciously influenced by peers who don't believe technical writing is important or by companies who don't appear to value communicative skills in their recruiting ads or interview processes. However, study participants all placed high value on learning to be strong technical writers and communicators, and some argued that these skills are more important than other technical skills.

Participants noted that writing is more core to other fields; perhaps more explicit technical writing instruction and assessment in computer science courses would help students understand how communication plays an important role in software development and other subfields of computer science. Sustained focus on technical writing and communication skills in undergraduate programs may affect students' attitudes. We see evidence from this study that students are conflicted in their perspectives. While they believe technical writing is important for themselves as computer scientists, they don't feel like they were taught how to do it, and they don't think it's valued by their communities. Wui articulated this tension when she said she was learning to challenge her belief that "technical writing is not a [computer science] skill".

The stereotypes that participants mentioned may act as further barriers to learning technical writing. Victor shared the perspective that it's 'cool' to not care about technical writing. Students' communities and their perception of what is valued by community members influences their learning.

4.5.3 Research Question 2: Implications

In this section, I discuss implications for research and practice in relation to Research Question 2.

Implications for Research

This study gathered perspectives towards learning technical writing in computer science courses from 10 participants who study at UBC. In the future, a research project with more diverse participants, participants attending a different post-secondary institution, or a longer data collection period, would allow further and deeper investigation of the range of perspectives held. A broader understanding of the beliefs held by diverse computer science students across institutional settings may allow us to develop a survey instrument to measure perspectives of a wider population. As an example, Byrne et al. (2020) developed and validated a survey instrument to measure engineering students' attitudes towards learning professional skills, which they defined as "leadership in teams, communication, civic and public engagement, cultural adaptability, and innovation" (p. 1417). If new pedagogical or curricular techniques are developed to address learning of technical writing in computer science, future research should assess their impact on student learning as well as on their perspectives (Dorn and Tew, 2015). The design and validation of a survey instrument to measure perspectives about learning technical writing or professional skills more broadly would allow such assessment.

Future research could investigate the influence of multiple and overlapping communities of practice on students' perspectives. For example, it may be interesting to study specific course-based or workplace-based communities to see how these particular communities affect individuals' perspectives.

Many factors may influence students' perspectives related to learning technical writing in computer science courses. An individual's English language fluency – when learning in an English-speaking institution – is worthy of further exploration. Participants in this study discussed biases that they believe English Language Learners face. The one participant in this study, Etson, who self-identified as not yet fluent in English shared his perceptions and perspectives about learning technical writing. It would be beneficial to hear from more English Language

Learner students.

Implications for Practice

Many of the perspectives that students shared would be detrimental to learning technical writing in computer science courses. In particular, if students don't believe that learning technical writing is important, they are unlikely to prioritize learning this skill. Computer science faculty who believe that learning technical writing is important should find explicit ways to communicate their belief. If faculty members express that they value technical writing and other communicative skills, students' perspectives may be affected, which could then influence their learning.

At the outset of this study, participants generally thought that technical writing was important, despite their perception that technical writing is undervalued in computer science communities. The reflections that they engaged in throughout the study reinforced its importance and some participants stated that the study encouraged them to reflect on their learning of technical writing. Faculty members could further support students' learning by including reflective prompts about students' writing experiences and the feedback that they receive on their writing in computer science courses. For example, the prompts used in this study could be included in weekly assignments or as an exit ticket exercise (Fowler et al., 2019) in lecture. Students' responses to reflective prompts could provide further data for faculty members to improve their courses or to conduct additional studies on students' experiences and perspectives.

Participants commented on their multiple roles as communicators and technical writers and the varying genres of writing that they do. Their broad view of technical writing was made clear by the varied examples that they gave. As just a few examples, they mentioned writing code that is easy to read, writing clear commit messages, and writing research papers for particular audiences. Faculty members could support student learning by drawing attention to the varied writing activities that students already do.

4.6 Finding 3: Automated Assessment Practices Affect Students' Perspectives Towards Learning Technical Writing

This is the third of three sections that explores findings related to one of the research questions. Here, I discuss Research Question 3: "How do course assessment practices affect students' learning of technical writing?"

In Section 4.5.1, I presented participants' general perspectives about the importance of technical writing. In relation to Research Question 3, in Section 4.6.1, I discuss the conflict between students' perspectives that technical writing is important and the (often low) priority that they assign to technical writing tasks in their course work. Further, I discuss the ways in which course assessment practices influence students' prioritization. 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 must decide what to include and exclude in their courses. Participants expressed that they wished technical writing was assessed more often and more thoroughly; with the growing use of autograders for assignments and exams, participants prioritized the aspects of an activity that would be graded.

4.6.1 Research Question 3: Results

The theme related to this research question is "Students have a conflicting desire to learn technical writing"; they believe technical writing is important but prioritize learning activities that will be more heavily assessed.

Participants Believe Technical Writing is Important for their Future Careers There was overwhelming 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.

Technical Writing Creates Career-Related Opportunities One participant explicitly stated that they think they get more opportunities than their peers because of their communication skills. They didn't necessarily think this was fair, but recognized that they may be preferred by employers or collaborators than others who have weaker communication skills.

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."

In addition, Jason believed that technical writing is a transferable skill; he said, "it's a transferable skill, so it's helpful if I decide I don't want to do computer science anymore."

Strong Technical Writing Allows Your Work to be Recognized A number of participants expressed that technical writing is a method for communicating their ideas to peers, colleagues, and managers and having their work recognized. Phyllis' quote illustrates this point: "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 explained how communicating 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?”

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: “it still helped us to improve it because 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 shared that she 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."

Participants Desire More Curricular Attention but Recognize Curricular Tradeoff
In individual participants and across the group of participants, there were conflicting views about whether there should be 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 that she found embarrassing, Wui reflected and 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 course 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 discussing 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 class time. She said, “he would even tell us, this is the importance of comments, like good comments does this. He was very like, OK, what does this line mean in a comment? we spent 30 minutes talking about comments.”

Technical Writing Was not Assessed Frequently or Deeply Enough Participants consistently held the opinion that their technical writing was not assessed often enough 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 “the only technical writing, it was very small, like 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.”

Krish discussed the assessments he experienced in Computer Science 304 (Introduction to Relational Databases), and believed that the lack of assessment of technical writing meant that what was important was whether the code works. He said, “304 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 discussed two computer science courses in which she had learned about technical writing. She was assigned a reading on how to write for HCI, 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: "[n]ot specifically on the quality of the writing itself. ... all of those written assignments are basically, to my understanding, just being graded on correctness." Etson also finds that feedback and assessment focuses on correctness: "[i]f there are questions that require writing, only about 4-5 sentences are maximum and TA only grades on the correctness of the answer. Therefore, I have not received feedback on grammar, meaning, or organization."

Phyllis explicitly stated that her perception of her learning was influenced by the lack of assessment. She said, "[s]ince the mechanics of writing were not assessed or a focus, I do not feel I learned much there."

Want and 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 on their written work: "ordinary written assignments don't mark technical writing, only content, and I'd like to get feedback for improvement" (Nico) and "I think that most of the things that we submit in school ... we don't really get a lot of feedback on the technical writing part." (Wui)

When Wui did have work assessed on technical writing, 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."

Participants Prioritize What is Assessed Participants expressed that assessment practices influence their learning and the way they prioritize course activities. Despite the consensus that technical writing was important for their careers, 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 so we can be comfortable with writing whatever works for us."

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 (PrairieLearn, Inc., 2024), an automated assessment tool that is used in many UBC computer science courses. The tool allows manual grading by instructors or teaching assistants, but is most commonly used as an autograding system where instructors create assessments (that are often randomized) and can programmatically assess students' 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, the autograder, ... 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, "[w]hile this would probably bring down people's grades, 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 losing these marks 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 spent time trying 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."

4.6.2 Research Question 3: Discussion

Here, I discuss the results of Research Question 3 in relation to the related literature. I will consider participants' opinions on the importance of learning technical writing in relation to the findings from other studies that gathered students' perceptions and I will discuss the finding that students' prioritize what is assessed in relation to other studies on automated assessment tools.

Zhang et al. (2023) and von Briesen (2023) found that computer science students believe that it is important to learn technical writing. Zhang et al. (2023) 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 (even if they did not always prioritize technical writing learning opportunities) 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 large, research-intensive computer science departments in 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 (2023) 135 survey respondents reported that writing helped them better understand both their code and the related computer science concepts that they were learning. These findings are consistent with my findings that students believe technical writing is important and that it can help them understand technical concepts. However, participants in my study also believed that others undervalue technical writing. Zhang et al.'s (2023) study did not explicitly ask all students about the importance of technical writing, so it does not provide evidence that the majority of students think that writing is unimportant. Further research to understand perceptions of a full student body is warranted. Further research could also explore the disconnect between my participants' beliefs and their perception of their peers' beliefs.

von Briesen (2023) 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) and that 7 of 18 students expressed their wishes 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 provides evidence that they are disincentivized from doing the work necessary to achieve this learning when they are primarily assessed and receive feedback via automated tools.

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" (Nicol and Macfarlane-Dick, 2006, p. 204). Much of the feedback that post-secondary students receive is in response to graded assessments. In computer science courses, interest in automated assessment tools is growing (Paiva et al., 2022). 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, automated assessment tools therefore cannot provide feedback that helps a student assess their current performance and improve their writing towards 'good performance'. Students can over-rely on automated assessment tools (Baniassad et al., 2021) and spend their effort trying to get their solution to pass all of the autograder tests (Luxton-Reilly et al., 2023) rather than thinking more holistically about the solution they are designing.

Faculty members should consider including 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 (Tims et al., 2023); large class sizes and limited resources lead to higher adoption of automated assessment tools because manual assessment becomes unmanageable (Souza et al., 2016). 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.

4.6.3 Research Question 3: Implications

In this section, I discuss implications for research and practice in relation to Research Question 3.

Implications for Research

Automated assessment tools are becoming more commonly used in computer science courses, but the implications of the use of these tools on student learning and students' experiences needs to be further explored. Scholars have studied the impact of automated assessment tools on some student behaviours, such as submission patterns, and some student learning, such as ability to write functionally correct code. However, a holistic view on students' learning has not been conducted. Future research could investigate whether students are achieving the full range of expected learning outcomes, including professional skills such as communication.

The participants in my 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 of learning technical writing by gathering data from a wider group of participants.

Implications for Practice

Computer science instructors should consider integrating technical writing into their courses. Findings from this study suggest that assessing technical writing in a meaningful way will encourage students to prioritize it as a learning activity.

In this study, participants expressed a belief 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 the aspects of assessments that will be more heavily graded. Automated assessment tools have benefits for computer science students and instructors, including greater consistency in marking, reducing workload in grading, and identifying academic misconduct (Luxton-Reilly et al., 2023). 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. A variety of assessment types rather than relying on a singular type of autograded assessment would encourage students to meet a broader set of learning outcomes. These varied assessments could be embedded in courses across the curriculum and co-exist with autograded assessments.

Chapter 5

Conclusion

Technical writing and other communication skills are important to the success of computer science undergraduate students (Vivian et al., 2013). To the best of my knowledge, limited research has examined students' experiences and perspectives about learning technical writing in computer science courses. This study will begin to fill the research gap.

This study was guided by three research questions:

- RQ1: How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?
- RQ2: In what ways do students hold perspectives that may be barriers to their learning of technical writing?
- RQ3: How do course assessment practices affect students' learning of technical writing?

My goal was to answer these research questions and provide implications for practice and further research.

With the theoretical foundation of Situated Learning Theory (Lave and Wenger, 1991) and the Social Theory of Learning (Wenger, 2009), I conducted an interpretive, qualitative study to answer my research questions. To collect data, I interviewed 10 upper-year computer science undergraduate students twice, once at the beginning of a semester, and once at the end of a semester, and asked them to write a reflective journal entry in response to a prompt offered once a week. I transcribed the interviews and used reflexive thematic analysis (Braun and Clarke, 2020) to generate themes from the data. I grouped the themes relevant to each research question together.

In this chapter, I summarize the results and corresponding implications for research and practice of each research question, synthesize results across research questions, and summarize the contributions of this dissertation. A summary of implications for research is provided in Table 5.1 and a summary of implications for practice is provided in Table 5.2.

5.1 Summary of Results and Implications

RQ1: How do Students Describe Their Experiences Learning and Doing Technical Writing with Others? What are Their Reflections on These Experiences? Participants' experiences learning and doing technical writing with others was influenced by their communities. They often learned implicitly from community members' writing; participants reflected on the writing of others and emulated aspects they found effective. Participants often wished for more explicit teaching and found lack of instruction stressful. They were keenly aware of the importance of audience in their technical writing and discussed difficulties in understanding needs or expectations of audience members who have different academic or work backgrounds than their own. Participants also expressed a deep appreciation for working with and being mentored by community members with varied academic and work experiences. Finally, they appreciated and discussed the social nature of technical writing.

These findings provide evidence that participants learn technical writing via legitimate peripheral participation in their communities of practice. In some computer science programs, some technical writing skills are taught explicitly, while in many contexts, students learn these skills implicitly. It's important to recognize that students would like to receive more explicit instruction, while still acknowledging that they are learning via their legitimate peripheral participation in communities of practice. Faculty members should attend to both kinds of learning. Further study is warranted to better understand how computer science students learn collaborative, communicative skills in community and which technical writing skills might be best suited to explicit instruction.

Participants were members of multiple communities of practice at one time and their membership in communities of practice was fluid as they moved through their degree or through work experiences. With participants belonging to workplace and student communities, it would be interesting to explore how and why practices and learning transfer across these communities. Does the learning tend to move in one direction from industry to academia or vice versa? Does movement of ideas and practices depend on other relationships between communities or between community members?

The evidence that participants were learning technical writing as members of multiple communities of practice in industry and academia provides support for integrating teaching and learning of technical writing across the curriculum. Participants want to learn technical writing from technical experts and appreciate learning from those with different academic and work backgrounds. Perhaps more importantly, they are learning technical writing from community members regardless of whether the community members are intending to teach technical writing. Therefore, I argue student learning would be enhanced if more attention is paid to teaching technical writing in computer science courses. Students should have opportunities to practice technical writing and should receive meaningful feedback on their technical writing. Students would benefit from instruction on how to give peer feedback on technical writing. Learning how to give meaningful, constructive feedback on a peer's or near-peer's writing could enhance learning of technical writing in post-secondary computer science programs and beyond as students graduate and progress in their careers.

Computer science faculty members and teaching assistants are not generally trained to teach writing so may not feel comfortable providing explicit technical writing instruction. A long-term, sustained collaboration with writing experts could facilitate effective integration of teaching technical writing skills into computer science classrooms. It's challenging for non-language experts to teach writing and provide effective feedback, but students want to receive feedback from computer science instructors. A collaborative effort between writing experts and computer science faculty could provide learning opportunities that would not be available from a course focused on either area alone.

RQ2: In What Ways do Students Hold Perspectives that May Be Barriers to Their Learning of Technical Writing? Participants shared perspectives that they hold, or that they think their peers hold, that may be barriers to learning technical writing. For example, participants shared the beliefs that technical writing is less important than other technical skills and that computer scientists are not strong communicators. In addition, participants shared that technical writing is something that is taught in non-computer science courses and is more valued in other fields than it is in computer science. Participants specifically mentioned the humanities and business as fields that place higher value on technical writing.

Future research could investigate the influence of multiple and overlapping communities of practice on students' perspectives. For example, it may be interesting to study specific course-based or workplace-based communities to see how these particular communities affect individuals' perspectives.

Many factors may influence students' perspectives related to learning technical writing in computer science courses. An individual's English language fluency – when learning in an English-speaking institution – is worthy of further exploration. Participants in this study discussed biases that they believe English Language Learners face. The one participant in this study who self-identified as not yet fluent in English, Etson, shared his perceptions and perspectives about learning technical writing. It would be beneficial to hear from more English Language Learner students.

RQ3: How do Course Assessment Practices Affect Students' Learning of Technical Writing? Participants uniformly believed that learning technical writing was important for their career path, regardless of the career path that they intended to pursue. They expressed a desire to learn more technical writing in their computer science courses, but appreciated that curricular tradeoffs exist and recognized that other content would need to be removed. Participants did not think that their technical writing was assessed often or deeply enough in computer science courses.

They shared that course assessment practices, such as heavy use of automated assessment tools that grade for correctness, influence the way that they spend their time. Participants prioritize activities that will be assessed and often ignore those that won't be assessed. Since technical writing is not often assessed, students do not allocate time to learning it. Despite a genuine desire to learn technical writing, the practicalities of being a busy student influence the aspects of a course that get the most attention. However, despite the constraints placed on students by course assessment practices, participants in this study shared a variety of ways in which they are learning about technical writing. They are reflective and thoughtful about the technical writing they read and emulate effective writing techniques in their own writing.

Automated assessment tools (AATs) are becoming more commonly used in computer science courses, but the implications of their use on student learning and students' experiences needs to be further explored. Scholars have studied the impact of automated assessment tools on some student behaviours, such as submission patterns, and some student learning, such as ability to write functionally correct code, but a more holistic investigation of students' learning of skills that cannot be assessed with automated assessment tools has not been conducted. Students would likely benefit from completing a variety of assessments tailored to meet a broad set of learning objectives, including learning objectives focused on technical writing. Further study of the impact of automated assessment practices on student learning would provide a foundation for designing a holistic, program-wide assessment approach.

Computer science instructors should consider integrating technical writing more fully into their courses. Findings from this study suggest that assessing the writing in a meaningful way will encourage students to prioritize it as a learning activity.

5.2 Synthesizing Findings

This research study addressed three interrelated research questions. The findings of the three research questions, when considered together, demonstrate the nuanced perceptions and understandings that the study participants have of learning technical writing. Participants believe that technical writing is important, but don't think

Table 5.1: Implications for Research

Research Question	Implication
RQ1	Future research could explore students' or professionals' use of Generative AI writing tools in relation to learning technical writing from community members.
RQ1	Future research could explore how and why practices and learning transfer across communities of practice.
RQ2	Future research could investigate the influence of multiple and overlapping communities of practice on students' perspectives.
RQ2	Future research could explore additional factors that may influence students' perspectives related to learning technical writing in computer science courses, such as English language fluency.
RQ2 and RQ3	A future research project with more diverse participants, participants attending a different post-secondary institution, or a longer data collection period, would allow further and deeper investigation of the range of perspectives held. A broader understanding of the beliefs held by diverse computer science students across institutional settings may allow us to develop a survey instrument to measure perspectives of a wider population.
RQ3	In courses or programs that heavily use automated assessment tools, future research could investigate whether students are achieving the full range of expected learning outcomes, including professional skills such as communication.

that their peers share this belief. Participants described their reflective practices in consideration of their own technical writing and the technical writing of others, but also stated that computer science course assessment practices cause them to de-prioritize technical writing activities. Participants do not believe that they learn from unassessed course-related writing activities, but also described the mature approaches they use to complete the assignments, indicating that they are likely learning about technical writing despite their perception that they are not. Participants want to be taught explicitly, but demonstrated capacity for learning from examples and emulating good examples that they find.

Legitimate peripheral participation in communities of practice provides a theoretical lens for understanding these apparent contradictions and complications

Table 5.2: Implications for Practice

Research Question	Implication
RQ1	Instructors could consider integrating teaching and learning of technical writing across the curriculum. A long-term, sustained collaboration with writing experts could facilitate effective integration of teaching technical writing skills into computer science classrooms.
RQ2	Instructors who believe that learning technical writing is important could find explicit ways to communicate their belief. If faculty members express that they value technical writing and other communicative skills, students' perspectives may be affected, which could then influence their learning.
RQ2	Instructors could further support students' learning by including reflective prompts about students' writing experiences and the feedback that they receive on their writing in computer science courses.
RQ2	Instructors could support student learning by drawing attention to the varied writing activities that students already do in computer science courses.
RQ3	Instructors could review their assessment practices to ensure that they are assessing technical writing in a meaningful way, which could encourage students to prioritize it as a learning activity.
RQ3	In courses or programs that heavily use automated assessment tools, instructors could consider employing a variety of assessment types which may encourage students to meet a broader set of learning outcomes. These varied assessments could be embedded in courses across the curriculum and co-exist with autograded assessments.

across the findings. Students are learning from community members in their multiple communities of practice. They learn from peers in courses, near-peers as teaching assistants, and 'oldtimers' such as managers, instructors, or mentors who have more technical writing experience. They often learn implicitly through reflection on community norms and what aspects of technical writing are worthy of emulation. Lave and Wenger (1991) wrote that "[l]earning, transformation, and change are always implicated in one another" (p. 57); learning technical writing is influenced by community beliefs and stereotypes, and over time, these beliefs and stereotypes may change as new members, such as the study participants, transform communities' understandings and practices.

5.3 Contributions

The key contributions of this dissertation are:

- A methodologically sound reflexive thematic analysis which I believe is uncommon in computer science education research and may serve as an example for future research.
- Themes that may transfer to other contexts in which students are learning technical writing.
- Pedagogical and curricular implications for UBC and beyond.

5.3.1 Methodologically Sound Reflexive Thematic Analysis

Although reflexive thematic analysis has been used by other computer science education researchers (e.g., Bausch et al., 2024; Bernstein et al., 2024; Björn and Kann, 2023; Denny et al., 2024a,b; Gray et al., 2021; Ikeda et al., 2023; Indriasari et al., 2021; Lusa Krug et al., 2023; Paterson et al., 2022; Prather et al., 2023; Sentance and Waite, 2021), it is not common for these studies to report their use of reflexive thematic analysis as completely or transparently as Braun and Clarke suggest. Braun and Clarke (2019) state, "[r]eflexive TA needs to be implemented with theoretical knowingness and transparency; the researcher strives to be fully cognisant of the philosophical sensibility and theoretical assumptions informing their use of

TA; and these are consistently, coherently and transparently enacted throughout the analytic process and reporting of the research" (p. 594). Most of the computer science education research studies only provide a few sentences that describe their use of reflexive thematic analysis and do not provide their philosophical or theoretical assumptions. Many computer science education research papers are subject to a page limit, so researchers may have elided this information for brevity. Sentance and Waite (2021) provide an example of a study reported with the detail that fulfills Braun and Clarke's expectations. My dissertation provides an additional example of a methodologically-sound, transparently described, reflexive thematic analysis in computer science education research.

5.3.2 Themes That May Transfer to Other Contexts

This study provides a rich understanding of students' experiences of learning technical writing with others, students' perspectives on learning technical writing, and how course assessment practices affect students' learning of technical writing. The eight themes generated via reflexive thematic analysis expand knowledge about students' perceptions of their technical writing experiences in computer science courses and their perspectives related to learning about technical writing. The themes may transfer to other non-computer science contexts in which students learn technical writing.

5.3.3 Pedagogical and Curricular Implications

As summarized in Table 5.2, my dissertation has led to pedagogical and curricular suggestions for computer science educators. These suggestions may improve computer science education at UBC and beyond, and may be relevant for non-computer science educators.

Bibliography

Adams, W. K., Perkins, K. K., Dubson, M., Finkelstein, N. D., and Wieman, C. E. (2005). The design and validation of the Colorado Learning Attitudes about Science Survey. In *American Institute of Physics Conference Proceedings*. American Institute of Physics.

Adams, W. K., Wieman, C. E., Perkins, K. K., and Barbera, J. (2008). Modifying and validating the Colorado Learning Attitudes about Science Survey for use in chemistry. *Journal of Chemical Education*, 85(10):1435.

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education*, 15(2):83–102.

Amazon Web Services (2024). Front end vs back end - difference between application development - AWS. Amazon Web Services, Inc. Retrieved from: https://aws.amazon.com/compare/the-difference-between-frontend-and-backend/.

Andres, L. (2012). *Designing & Doing Survey Research*. SAGE Publications, Ltd.

Baniassad, E., Zamprogno, L., Hall, B., and Holmes, R. (2021). STOP THE (AUTOGRADER) INSANITY: Regression penalties to deter autograder overreliance. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1062–1068. Association for Computing Machinery.

Bausch, G., Ni, L., Thomas-Cappello, E., Martin, F., Feliciano, B., and Mirderikvand, F. (2024). Investigating middle school students' early learning experience of computer science through creating apps for social good. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education Volume 1*, pages 95–101. Association for Computing Machinery.

Begel, A. and Simon, B. (2008). Struggles of new college graduates in their first software development job. In *Proceedings of the 39th ACM Technical*

Symposium on Computer Science Education, pages 226–230. Association for Computing Machinery.

Ben-Ari, M. (1998). Constructivism in computer science education. In *Proceedings of the 29th ACM Technical Symposium on Computer Science Education*, pages 257–261. Association for Computing Machinery.

Bergen, N. and Labonté, R. (2020). "Everything Is Perfect, and We Have No Problems": Detecting and limiting social desirability bias in qualitative research. *Qualitative Health Research*, 30(5):783–792.

Bernstein, S., Denny, P., Leinonen, J., Kan, L., Hellas, A., Littlefield, M., Sarsa, S., and Macneil, S. (2024). "Like a Nesting Doll": Analyzing recursion analogies generated by CS students using large language models. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education Volume 1*, pages 122–128. Association for Computing Machinery.

Björn, C. and Kann, V. (2023). Variables affecting students' success in CS2. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education Volume 1*, pages 257–263. Association for Computing Machinery.

Blunt, T. and Pearson, T. (2021). Exploring the digital identity divide: A call for attention to computing identity at HBCUs. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 640–646. Association for Computing Machinery.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101.

Braun, V. and Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4):589–597.

Braun, V. and Clarke, V. (2020). One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative Research in Psychology*, 18(3):328–352.

Braun, V. and Clarke, V. (2022a). *Thematic analysis: A practical guide*. SAGE Publications Inc.

Braun, V. and Clarke, V. (2022b). Toward good practice in thematic analysis: Avoiding common problems and be(com)ing a *knowing* researcher. *International Journal of Transgender Health*, 24(1):1–6.

Braun, V., Clarke, V., Hayfield, N., and Terry, G. (2019). Thematic analysis. In Liamputtong, P., editor, *Handbook of Research Methods in Health Social Sciences*, pages 843–860. Springer Singapore.

Brinkmann, S. (2018). The interview. In Denzin, N. K. and Lincoln, Y. S., editors, *The SAGE Handbook of Qualitative Research*, pages 576–599. SAGE Publications, Inc., 5th edition.

Burge, J. E., Gannod, G., Carter, M., Howard, A., Schultz, B., Vouk, M., Wright, D., and Anderson, P. (2014). Developing CS/SE students' communication abilities through a program-wide framework. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, pages 579–584. Association for Computing Machinery.

Byrne, Z. S., Weston, J. W., and Cave, K. (2020). Development of a scale for measuring students' attitudes towards learning professional (i.e., soft) skills. *Research in Science Education*, 50(4):1417–1433.

Byrnes, H. (2013). Positioning writing as meaning-making in writing research: An introduction. *Journal of Second Language Writing*, 22(2):95–106.

Cahill, M. J., McDaniel, M. A., Frey, R. F., Hynes, K. M., Repice, M., Zhao, J., and Trousil, R. (2018). Understanding the relationship between student attitudes and student learning. *Physical Review Physics Education Research*, 14:010107.

Charmaz, K. (2000). Grounded theory: Objectivist and contructivist methods. In Denzin, N. K. and Lincoln, Y. S., editors, *The Handbook of Qualitative Research*, pages 509–535. SAGE Publications, Inc., 2nd edition.

Cilliers, C. B. (2012). Student perception of academic writing skills activities in a traditional programming course. *Computers and Education*, 58(4):1028–1041.

Clarke, V., Braun, V., and Hayfield, N. (2015). Thematic analysis. In Smith, J. A., editor, *Qualitative Psychology: A Practical Guide to Research Methods*, pages 222–248. SAGE Publications Ltd., 3rd edition.

Cooper, S., Rodger, S. H., Schep, M., Stalvey, R. H., and Dann, W. (2015). Growing a K-12 Community of Practice. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 290–295. Association for Computing Machinery.

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

CS2008 Review Joint Task Force (2008). *Computer Science Curriculum 2008: An Interim Revision of CS 2001*. Association for Computing Machinery and Institute of Electrical and Electronics Engineers.

Decker, A., Edwards, S. H., McSkimming, B. M., Edmison, B., Rorrer, A., and Pérez Quiñones, M. A. (2024). Transforming grading practices in the computing education community. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education Volume 1*, pages 276–282. Association for Computing Machinery.

Denny, P., MacNeil, S., Savelka, J., Porter, L., and Luxton-Reilly, A. (2024a). Desirable characteristics for AI teaching assistants in programming education. In *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education Volume 1*, pages 408–414. Association for Computing Machinery.

Denny, P., Smith, D. H., Fowler, M., Prather, J., Becker, B. A., and Leinonen, J. (2024b). Explaining code with a purpose: An integrated approach for developing code comprehension and prompting skills. In *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education Volume 1*, pages 283–289. Association for Computing Machinery.

Dorn, B. and Tew, A. E. (2015). Empirical validation and application of the Computing Attitudes Survey. *Computer Science Education*, 25(1):1–36.

Dweck, C. S. (2006). *Mindset: The new psychology of success*. Random House.

Dym, B., Rockwood, C., and Fiesler, C. (2023). Gaming together, coding together: Collaborative pathways to computational learning. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education Volume 1*, pages 1035–1041. Association for Computing Machinery.

Elliott Tew, A., Dorn, B., and Schneider, O. (2012). Toward a validated computing attitudes survey. In *Proceedings of the 9th Annual International Conference on International Computing Education Research*, pages 135–142. Association for Computing Machinery.

Ellis, C. (2007). Telling secrets, revealing lives: Relational ethics in research with intimate others. *Qualitative Inquiry*, 13(1):3–29.

Emig, J. (1977). Writing as a mode of learning. *College Composition and Communication*, 28(2):122–128.

Epp, S. (2008). The value of reflective journaling in undergraduate nursing education: A literature review. *International Journal of Nursing Studies*, 45(9):1379–1388.

Etlinger, H. A. (2006). A framework in which to teach (technical) communication to computer science majors. In *Proceedings of the 37th ACM Technical Symposium on Computer Science Education*, pages 122–126. Association for Computing Machinery.

Exter, M., Caskurlu, S., and Fernandez, T. (2018). Comparing computing professionals' perceptions of importance of skills and knowledge on the job and coverage in undergraduate experiences. *ACM Transactions on Computing Education*, 18(4):1–29.

Falkner, K. and Falkner, N. J. (2012). Integrating communication skills into the computer science curriculum. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, pages 379–384. Association for Computing Machinery.

Fell, H. J., Proulx, V. K., and Casey, J. (1996). Writing across the computer science curriculum. In *Proceeding of the 27th ACM Technical Symposium on Computer Science Education*, pages 204–209. Association for Computing Machinery.

Fincher, S. and Petre, M. (2004). A preface to pragmatics. In Fincher, S. and Petre, M., editors, *Computer Science Education Research*, pages 9–18. Taylor & Frances.

Fincher, S. and Tenenberg, J. (2006). Using theory to inform capacity-building: Bootstrapping communities of practice in computer science education research. *Journal of Engineering Education*, 95(4):265–277.

Finlay, L. (2021). Thematic analysis: The 'Good', the 'Bad' and the 'Ugly'. *European Journal for Qualitative Research in Psychotherapy*, 11:103–116.

Fontana, A. and Frey, J. (1994). Interviewing: The art of science. In Denzin, N. K. and Lincoln, Y. S., editors, *The Handbook of Qualitative Research*, pages 361–376. SAGE Publications Inc.

Fowler, K., Windschitl, M., and Richards, J. (2019). Exit tickets. *The Science Teacher*, 86(8):18–26.

Galster, M., Mitrovic, A., Malinen, S., and Holland, J. (2022). What soft skills does the software industry *really* want? An exploratory study of software positions in New Zealand. In *Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 272–282. Association for Computing Machinery.

GitHub, Inc. (2024). Github definition. GitHub, Inc. Retrieved from: github.com.

Gray, C. M., Chivukula, S. S., Melkey, K., and Manocha, R. (2021). Understanding "dark" design roles in computing education. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, pages 225–238. Association for Computing Machinery.

Greiman, B. C. and Covington, H. K. (2007). Reflective thinking and journal writing: Examining student teachers' perceptions of preferred reflective modality, journal writing outcomes, and journal structure. *Career and Technical Education Research*, 32(2):115–139.

Groeneveld, W., Jacobs, H., Vennekens, J., and Aerts, K. (2020). Non-cognitive abilities of exceptional software engineers: A Delphi study. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 1096–1102. Association for Computing Machinery.

Große-Bölting, G., Gerstenberger, D., Gildehaus, L., Mühling, A., and Schulte, C. (2021). Identity in K-12 computer education research: A systematic literature review. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, pages 169–183. Association for Computing Machinery.

Gruba, P. and Søndergaard, H. (2001). A constructivist approach to communication skills instruction in computer science. *Computer Science Education*, 11(3):203–219.

Guba, E. G. and Lincoln, Y. S. (1981). *Effective Evaluation*. A joint publication in the Jossey-Bass higher education and social and behavioral sciences series. Jossey-Bass Publishers.

Guile, D. and Young, M. (1999). Beyond the institution of apprenticeship: Towards a social theory of learning as the production of knowledge. In Ainley, P. and Rainbird, H., editors, *Apprenticeship: Towards a New Paradigm of Learning*, pages 111–128. Routledge, London.

Guzdial, M. and Tew, A. E. (2006). Imagineering inauthentic legitimate peripheral participation: An instructional design approach for motivating computing education. In *Proceedings of the Second International Workshop on Computing Education Research*, pages 51–58. Association for Computing Machinery.

Hayman, B., Wilkes, L., and Jackson, D. (2012). Journaling: Identification of challenges and reflection on strategies. *Nurse Researcher*, 19(3):27–31.

Hazzan, O. and Har-Shai, G. (2013). Teaching computer science soft skills as soft concepts. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, pages 59–64. Association for Computing Machinery.

Hazzan, O. and Har-Shai, G. (2014). 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*, pages 567–572. Association for Computing Machinery.

Hewner, M. and Guzdial, M. (2008). Attitudes about computing in postsecondary graduates. In *Proceedings of the Fourth International Workshop on Computing Education Research*, pages 71–78. Association for Computing Machinery.

Hoffman, M. E., Dansdill, T., and Herscovici, D. S. (2006). Bridging writing to learn and writing in the discipline in computer science education. In *Proceedings of the 37th ACM Technical Symposium on Computer Science Education*, pages 117–121. Association for Computing Machinery.

Holmes, A. G. D. (2020). Researcher positionality–a consideration of its influence and place in qualitative research–a new researcher guide. *Shanlax International Journal of Education*, 8(4):1–10.

Hubbs, D. L. and Brand, C. F. (2005). The paper mirror: Understanding reflective journaling. *Journal of Experiential Education*, 28(1):60–71.

Ikeda, B., Hoelscher, J., Alterovitz, R., and Szafir, D. (2023). Guiding the development of undergraduate educational robotics. In *Proceedings of the ACM Conference on Global Computing Education Vol 1*, pages 71–77. Association for Computing Machinery.

Indriasari, T. D., Luxton-Reilly, A., and Denny, P. (2021). Investigating accuracy and perceived value of feedback in peer code review using gamification. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education Volume 1*, pages 199–205. Association for Computing Machinery.

Iniesto, F., Sargent, J., Rienties, B., Llorens, A., Adam, A., Herodotou, C., Ferguson, R., and Muccini, H. (2021). When industry meets education 4.0: What do computer science companies need from higher education? In *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 367–372.

Janesick, V. J. (1999). A journal about journal writing as a qualitative research technique: History, issues, and reflections. *Qualitative Inquiry*, 5(4):505–524.

Jeffery, E., Nomme, K., Deane, T., Pollock, C., and Birol, G. (2016). Investigating the role of an inquiry-based biology lab course on student attitudes and views toward science. *CBE—Life Sciences Education*, 15(4):ar61.

Joint Task Force on Computing Curricula (2001). *Computing Curricula 2001 Computer Science Final Report*. Association for Computing Machinery and Institute of Electrical and Electronics Engineers.

Joint Task Force on Computing Curricula (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery and Institute of Electrical and Electronics Engineers.

Joint Task Force on Computing Curricula (2024). *Computer Science Curricula 2023*. Association for Computing Machinery and Institute of Electrical and Electronics Engineers.

Joy, M., Sinclair, J., Sun, S., Sitthiworachart, J., and López-González, J. (2009). Categorising computer science education research. *Education and Information Technologies*, 14(2):105–126.

Kaczmarczyk, L. C. (2003). A technical writing class for computer science majors: Measuring student perceptions of learning. *SIGCSE Bulletin*, 35(1):341–345.

Kafai, Y., Proctor, C., and Lui, D. (2019). From theory bias to theory dialogue: Embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. In *Proceedings of the 15th ACM Conference on International Computing Education Research*, pages 101–109. Association for Computing Machinery.

Kapoor, A. and Gardner-McCune, C. (2018). Understanding professional identities and goals of computer science undergraduate students. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 191–196. Association for Computing Machinery.

Kapoor, A. and Gardner-McCune, C. (2019). Understanding CS undergraduate students' professional development through the lens of internship experiences. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 852–858. Association for Computing Machinery.

Karatsolis, A., Cervasato, I., Harras, K., Cooper, Y., Oflazer, K., Abu-Ghazaleh, N., and Sans, T. (2011). Getting CS undergraduates to communicate effectively. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, pages 283–287. Association for Computing Machinery.

Kay, D. G. (1998). Computer scientists can teach writing: An upper division course for computer science majors. In *Proceedings of the 29th ACM Technical Symposium on Computer Science Education*, pages 117–120. Association for Computing Machinery.

Kortsarts, Y., Fischbach, A., Rufinus, J., Utell, J., and Yoon, S.-C. (2010). Developing oral and written communication skills in undergraduate computer science and information systems curriculum. *Information Systems Education Journal*, 8(30):3–13.

Lave, J. (1991). Situating learning in communities of practice. In Resnick, L. B., Levine, J. M., and Teasley, S. D., editors, *Perspectives on Socially Shared Cognition*, pages 63–82. American Psychological Association.

Lave, J. (2009). The practice of learning. In Illeris, K., editor, *Contemporary theories of learning: Learning theorists ... in their own words*, pages 208–216. Routledge.

Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.

Lave, J. and Wenger, E. (2002). Practice, person, social world. In Daniels, H., editor, *An introduction to Vygotsky*, pages 155–162. Routledge.

Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., and Ninaus, M. (2019). Development of a questionnaire on self-concept, motivational beliefs, and attitude towards programming. In *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*. Association for Computing Machinery.

Leinonen, J., Denny, P., and Whalley, J. (2022). 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*, pages 885–891. Association for Computing Machinery.

Lethbridge, T. (1998). A survey of the relevance of computer science and software engineering education. In *Proceedings of the 11th Conference on Software Engineering Education and Training*, pages 56–66. IEEE Computer Society.

Lewis, A. N., Politz, J. G., Vaccaro, K., and Minnes, M. (2022). Learning about the experiences of Chicano/Latino students in a large undergraduate CS program. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Volume 1*, pages 165–171. Association for Computing Machinery.

Lewis, C., Jackson, M. H., and Waite, W. M. (2010). Student and faculty attitudes and beliefs about computer science. *Communications of the ACM*, 53(5):78–85.

Li, P. L., Ko, A. J., and Zhu, J. (2015). What makes a great software engineer? In *Proceedings of the IEEE/ACM 37th IEEE International Conference on Software Engineering*, pages 700–710. IEEE Computer Society.

Lincoln, Y. S. and Guba, E. G. (1985). *Naturalistic Inquiry*. SAGE Publications, Inc.

Lingard, L. (2019). Beyond the default colon: Effective use of quotes in qualitative research. *Perspectives on Medical Education*, 8(6):360–364.

Lumivero (2024a). NVivo 14. Lumivero. Retrieved from: https://lumivero.com/product/nvivo-14/.

Lumivero (2024b). NVivo Transcription. Lumivero. Retrieved from: https://lumivero.com/products/nvivo-transcription/.

Lusa Krug, D., Mouza, C., Jones, W. M., Barnett, T., and Shepherd, D. C. (2023). Attracting adults to computer programming via hip hop. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education Volume 1*, pages 528–534. Association for Computing Machinery.

Lutz, B. and Paretti, M. (2019). Development and implementation of a reflective journaling method for qualitative research. In *2019 ASEE Annual Conference & Exposition Proceedings*, pages 1–14. American Society for Engineering Education.

Luxton-Reilly, A., Tempero, E., Arachchilage, N., Chang, A., Denny, P., Fowler, A., Giacaman, N., Kontorovich, I., Lottridge, D., Manoharan, S., Sindhwani, S., Singh, P., Speidel, U., Stephen, S., Terragni, V., Whalley, J., Wuensche, B., and Ye, X. (2023). Automated assessment: Experiences from the trenches. In *Proceedings of the 25th Australasian Computing Education Conference*, pages 1–10. Association for Computing Machinery.

Malmi, L., Sheard, J., Kinnunen, P., Simon, and Sinclair, J. (2020a). Computing education theories: What are they and how are they used? *ACM Inroads*, 11(1):54–64.

Malmi, L., Sheard, J., Kinnunen, P., Simon, and Sinclair, J. (2020b). Theories and models of emotions, attitudes, and self-efficacy in the context of programming education. In *Proceedings of the 16th ACM Conference on International Computing Education Research*, pages 36–47. Association for Computing Machinery.

Malmi, L., Sheard, J., Simon, Bednarik, R., Helminen, J., Kinnunen, P., Korhonen, A., Myller, N., Sorva, J., and Taherkhani, A. (2014). Theoretical underpinnings of computing education research: What is the evidence? In *Proceedings of the 10th Annual Conference on International Computing Education Research*, pages 27–34. Association for Computing Machinery.

Malmi, L., Sheard, J., Simon, Bednarik, R., Helminen, J., Korhonen, A., Myller, N., Sorva, J., and Taherkhani, A. (2010). Characterizing research in computing education: a preliminary analysis of the literature. In *Proceedings of the Sixth International Workshop on Computing Education Research*, pages 3–12. Association for Computing Machinery.

Marshall, C. and Rossman, G. B. (2016). *Designing qualitative research*. SAGE Publications, Inc., 6th edition.

Masika, R. and Jones, J. (2016). Building student belonging and engagement: Insights into higher education students' experiences of participating and learning together. *Teaching in Higher Education*, 21(2):138–150.

Maynard, M. (1994). Methods, practice and epistemology: The debate about feminism and research. In Maynard, M. and Purvis, J., editors, *Researching Women's Lives From A Feminist Perspective*, pages 10–26. Routledge.

Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education: Revised and Expanded from Case Study Research in Education*. [Kindle Edition] Jossey-Bass Publishers.

Merriam-Webster (2023). Genre. In *Merriam-Webster.com dictionary*. Merriam-Webster, Inc. Retrieved from: https://www.merriam-webster.com/dictionary/genre.

Michael, M. (2000). Fostering and assessing communication skills in the computer science context. In *Proceedings of the 31st ACM Technical Symposium on Computer Science Education*, pages 119–123. Association for Computing Machinery.

Milner-Bolotin, M., Antimirova, T., Noack, A., and Petrov, A. (2011). Attitudes about science and conceptual physics learning in university introductory physics courses. *Physical Review Physics Education Research*, 7:020107.

Mitra, J. (2023). 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*, pages 388–394. Association for Computing Machinery.

Moon, J. A. (2013). *Reflection in learning and professional development: Theory and practice*. Routledge.

Moon, K. and Blackman, D. (2014). A guide to understanding social science research for natural scientists. *Conservation Biology*, 28(5):1167–1177.

Morse, J. (2018). Reframing rigor in qualitative inquiry. In Denzin, N. K. and Lincoln, Y. S., editors, *The SAGE Handbook of Qualitative Research*, pages 796–817. SAGE Publications, Inc., 5th edition.

Munir, R., Strafforello, F., Kani, N., Kaler, M., Simion, B., and Zhang, L. (2022). Exploring common writing issues in upper-year computer science. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education Volume 1*, pages 161–167. Association for Computing Machinery.

Nicol, D. J. and Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218.

Open AI (2024). ChatGPT. Retrieved from: https://chat.openai.com.

Paiva, J. C., Leal, J. P., and Figueira, A. (2022). Automated assessment in computer science education: A state-of-the-art review. *ACM Transactions on Computing Education*, 22(3).

Parikh, S. B., Janson, C., and Singleton, T. (2012). Video journaling as a method of reflective practice. *Counselor Education and Supervision*, 51(1):33–49.

Paterson, B. L. (1995). Developing and maintaining reflection in clinical journals. *Nurse Education Today*, 15(3):211–220.

Paterson, J. H., Adams, J., Foster, D., Baghban Karimi, O., Kazmi, Z., Lennon, R. G., Nazir, S., Stott, L., and White, L. (2022). Motivation and strategies for effective inclusion of cloud solution provider certifications in computing curricula. In *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 235–260. Association for Computing Machinery.

Perkins, K. K., Adams, W. K., Pollock, S. J., Finkelstein, N. D., and Wieman, C. E. (2005). Correlating student beliefs with student learning using the Colorado Learning Attitudes about Science Survey. In *American Institute of Physics Conference Proceedings*, volume 790, pages 61–64. American Institute of Physics.

Peters, A.-K., Berglund, A., Eckerdal, A., and Pears, A. (2015). Second year computer science and IT students' experience of participation in the discipline. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, pages 68–76. Association for Computing Machinery.

Phelps, R. (2005). The potential of reflective journals in studying complexity "in action". *Complicity: An International Journal of Complexity and Education*, 2(1):37–54.

Pollock, L. (2001). Integrating an intensive experience with communication skills development into a computer science course. In *Proceedings of the 32nd ACM Technical Symposium on Computer Science Education*, pages 298–291. Association for Computing Machinery.

PrairieLearn, Inc. (2024). PrairieLearn. Retrieved from: https://www.prairielearn.com on January 9, 2024.

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., and Savelka, J. (2023). The robots are here: Navigating the Generative AI revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 108–159. Association for Computing Machinery.

Radermacher, A. and Walia, G. (2013). Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, pages 525–530. Association for Computing Machinery.

Rao, D. M. (2019). Experiences with auto-grading in a systems course. In *Proceedings of the 2019 IEEE Frontiers in Education Conference*, pages 1–8. IEEE Computer Society.

Rodriguez, H. R. (2017). Using womanism and reflective journaling to understand the academic experiences of Black female undergraduates at a religiously-based PWI: A case study analysis. *Journal of Ethnographic & Qualitative Research*, 12(2):125–141.

Rutz, C. and Grawe, N. D. (2017). How writing program best practices have transformed Carleton College. *Peer Review*, 19(1).

Scaffidi, C. (2018). Employers' needs for computer science, information technology and software engineering skills among new graduates. *International Journal of Computer Science, Engineering and Information Technology*, 8(1):1–12.

Seidman, I. (2019). *Interviewing as Qualitative Research: A Guide for Researchers in Education and the Social Sciences*. Teachers College Press.

Semsar, K., Knight, J. K., Birol, G., and Smith, M. K. (2011). The Colorado Learning Attitudes about Science Survey (CLASS) for use in biology. *CBE — Life Sciences Education*, 10(3):268–278.

Sentance, S. and Humphreys, S. (2018). Understanding professional learning for computing teachers from the perspective of situated learning. *Computer Science Education*, 28(4):345–370.

Sentance, S. and Waite, J. (2021). Teachers' perspectives on talk in the programming classroom: Language as a mediator. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, pages 266–280. Association for Computing Machinery.

Simon, B., Esper, S., Porter, L., and Cutts, Q. (2013). Student experience in a student-centered peer instruction classroom. In *Proceedings of the 9th Annual ACM Conference on International Computing Education Research*, pages 129–136. Association for Computing Machinery.

Souza, D. M., Felizardo, K. R., and Barbosa, E. F. (2016). A systematic literature review of assessment tools for programming assignments. In *Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training*, pages 147–156. IEEE Computer Society.

Stake, R. E. (2005). Qualitative case studies. In Denzin, N. K. and Lincoln, Y. S., editors, *The SAGE Handbook of Qualitative Research*, pages 443–466. SAGE Publications, Inc., 3rd edition.

Szabo, C., Falkner, N., Petersen, A., Bort, H., Connolly, C., Cunningham, K., Donaldson, P., Hellas, A., Robinson, J., and Sheard, J. (2019). 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*, pages 89–109. Association for Computing Machinery.

Szabo, C. and Sheard, J. (2022). Learning theories use and relationships in computing education research. *ACM Transactions on Computing Education*, 23(1).

Tenenberg, J. (2010). Industry fellows: Bringing professional practice into the classroom. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 72–76. Association for Computing Machinery.

Tight, M. (2015). Theory application in higher education research: The case of communities of practice. *European Journal of Higher Education*, 5(2):111–126.

Tims, J. L., Tucker, C., Weiss, M. A., and Zweben, S. (2023). Computing enrollment and retention: Results from the 2021–22 undergraduate enrollment cohort. *ACM Inroads*, 14(4):24–43.

Tracy, S. J. (2010). Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative Inquiry*, 16(10):837–851.

UBC Computer Science Department (2019). Program learning outcomes. In *UBC Computer Science Department website*. UBC Computer Science Department. Retrieved from: https://www.cs.ubc.ca/program-learning-outcomes.

Vihavainen, A., Vikberg, T., Luukkainen, M., and Kurhila, J. (2013). Massive increase in eager TAs: Experiences from extreme apprenticeship-based CS1. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, pages 123–128. Association for Computing Machinery.

Vivian, R., Falkner, K., and Falkner, N. (2013). 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*, pages 105–114. Association for Computing Machinery.

Vivian, R., Falkner, K., Falkner, N., and Tarmazdi, H. (2016). A method to analyze computer science students' teamwork in online collaborative learning environments. *ACM Transactions on Computing Education*, 16(2):1–28.

von Briesen, E. (2023). 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*, 39(5):119–128.

Weidler-Lewis, J., DuBow, W., and Kaminsky, A. (2017). Defining a discipline or shaping a community: Constraints on broadening participation in computing. In *Proceedings of the 48th ACM Technical Symposium on Computer Science Education*, pages 627–632. Association for Computing Machinery.

Wenger, E. (1998a). Communities of practice: Learning as a social system. *Systems Thinker*, 9(5):1–5.

Wenger, E. (1998b). *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.

Wenger, E. (2009). A social theory of learning. In Illeris, K., editor, *Contemporary theories of learning: Learning theorists ... in their own words*, pages 217–240. Routledge.

Wikipedia (2024a). API. In *Wikipedia: The Free Encyclopedia*. Wikipedia. https://en.wikipedia.org/wiki/API.

Wikipedia (2024b). Assembly language. In *Wikipedia: The Free Encyclopedia*. Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/Assembly_language.

Wikipedia (2024c). Code review. In *Wikipedia: The Free Encyclopedia*. Wikipedia. https://en.wikipedia.org/wiki/Code_review.

Zamprogno, L., Holmes, R., and Baniassad, E. (2020). Nudging student learning strategies using formative feedback in automatically graded assessments. In *Proceedings of the 2020 ACM SIGPLAN Symposium on SPLASH-E*, pages 1–11. Association for Computing Machinery.

Zhang, L., Simion, B., Kaler, M., Liaqat, A., Dick, D., Bergen, A., Miljanovic, M., and Petersen, A. (2023). 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*, pages 610–616. Association for Computing Machinery.

Appendix A

Semi-Structured Initial Interviews: Planned Questions

- Can you tell me about yourself, for example, where you did attend high school? When did you start learning computer science?
- What courses have you taken at UBC Computer Science so far? How has it been going?
- How would you describe your community or communities in computer science at UBC? Who do you interact with?
  - Follow up to ask about their connections with peers, near peers (TAs, for example), staff, and instructors.
- Can you tell me about your computer-science related work experiences if you have any?
  - Follow up to ask: Have you done co-op? If not, have you used your computer science skills or knowledge in any jobs?
- What do you plan to do after you complete your computer science degree?
- What do you expect to learn about communication during your computer science degree?

- Do you think that computer science classes should explicitly teach speaking, reading, writing, presenting, or listening skills? Why or why not?
- Can you tell me what you understand technical writing in computer science to be?
  - Ask for a description or definition if they don't give one.
  - Can you give me some examples of technical writing?
  - After they answer, define technical writing for the rest of the interview as "writing to convey technical content", for example, a commit message, email, report, or user documentation.
- What is the value of technical writing for a student who is a computer science major?
- What do you think you should learn about technical writing during your computer science degree?
- In what ways do you think writing will be important for you as a computer scientist?
- What is the UBC computer science student culture around technical writing? By culture I mean values, beliefs, expectations. What about instructors and TAs?
- Do you think you can improve your writing skills? Why or why not?
- Are there things that have helped you improve your writing or things that haven't helped?
- Do you think that writing is easier or more difficult for others than it is for you? Why or why not?
- What have you learned from others about technical writing?
- Who do you learn about technical writing from?
- How do you learn from others about writing? Do you talk to anyone about technical writing? Have you read others' writing?

- Can you tell me about the opportunities you have to collaborate and practice technical writing with others?
- What have you taught others about technical writing?
- Who do you teach about technical writing?
- Do you think your English language skills are a benefit or barrier to you in your technical writing? How and why?
- How do you think you'll use your writing skills in your work after graduation? Do you think writing is valued by companies that you'd like to work for? Why or why not?
- We will be asking some demographic questions and questions about your family background in order to help us understand whether or how your identity affects your experiences learning about technical writing or your perspectives about technical writing.
  - How many languages do you speak? What language do you speak at home? What was the first language you learned?
    - (If English isn't their native language) Tell me about your experiences learning English? How would you describe your level of English fluency in speaking, reading, writing, and listening?
  - Are you a man, woman, non-binary person, or is there another way you'd prefer to describe your gender? Or do you prefer not to answer?
  - How would you describe your cultural background? Is it African, European, East Asian, South Asian, South East Asian, First Nations or Indigenous, Hispanic or Latinx, Middle Eastern, other, or do you prefer not to specify? (If they answer First Nations, Indigenous, or other, ask them to specify if they want to).

Appendix B

Reflective Journal Prompts

Week One

Consider a recent technical writing experience that was related to a computer science course. This could be writing documentation, writing an email to a teammate, writing a report, writing an issue in an issue tracker, or something else.

1. Describe the writing that you did.
2. Do you think that having an opportunity to practice this kind of technical writing in your coursework is valuable or not? Why?

Week Two

Consider a recent time when you did a particular kind of technical writing for the first time; this experience could be related to a computer science course or a tech job. The writing could be documentation, writing an email to a teammate, writing a report, writing an issue in an issue tracker, or something else.

1. Describe the writing that you did.
2. How did you learn how to do this kind of technical writing?

Week Three

Consider a time when you wrote a comprehensive technical document for the first time; this experience could be related to a computer science course or a tech job. The writing could be a research paper, user documentation, developer documentation, an assignment report, or something else.

1. Describe the writing that you did.

2. Please describe the emotions that you felt during the writing experience. For each emotion, if you can recall, please describe why you felt that way. For example, "I felt [an emotion] because ..."

Week Four

Consider a time when you received feedback on your technical writing. The writing could be a research paper, user documentation, developer documentation, an assignment report, or something else.

1. Describe the writing that you did and the feedback that you received.

2. Please describe the emotions that you felt after receiving the feedback. For each emotion, if you can recall, please describe why you felt that way. For example, "I felt [an emotion] because ..."

Week Five

Can you tell me about a time you worked with someone and you thought they didn't do a good job on some shared writing? What was it like? What made it a complicated situation?

Week Six

Thinking specifically about technical writing, what do you wish your computer science professors did differently to help you learn?

Week Seven

Can you tell me about an AHA moment (i.e., a moment of sudden insight) that you had when you realized the importance of technical writing?

Week Eight

Can you tell me about your technical writing experiences when working with writers who are non-native English speakers?

Week Nine

Thinking about some technical writing that was assessed in a computer science class, can you tell me as much as you can remember about that experience?

For example, you may want to write about:

1. who did the assessment (i.e., TA or instructor)?
2. what was assessed (e.g., grammar, organization, structure, technical accuracy, other)?
3. how was the writing assessment weighted vs. other aspects of the assignment/exam/report/other?
4. what feedback did you receive?
5. what (if anything) did you learn from the assessment?

Week Ten

In computer science courses, what assessment(s) on technical writing would aid your learning?

Week Eleven

What are your top three strengths in technical writing skills? How did you learn to master these skills?

What are the top three technical writing skills that you'd like to improve? How do you plan on improving these skills?

Week Twelve

As you reflect on your technical writing experiences, what advice would you give to first-year or second-year computer science students?

Appendix C

Semi-Structured Concluding Interviews: Planned Questions

- What kinds of technical writing have you done in your computer science courses this year? Tell me more about that.
- If no writing is mentioned, ask about some examples to see if they can recall doing that kind of writing (e.g., commit messages, emails, reports, issues, etc.)
- (For each example of technical writing) Can you tell me more about writing X?
  - What was the context for this writing (e.g., individual vs. group, intended audience)?
  - Were you taught how to write this kind of technical document? If not, how did you learn to do this kind of writing?
  - How long did it take?
  - Was it was challenging or not challenging?
  - Were the emotions you had related to this writing experience similar or different to emotions that you feel when doing your other computer science work?

- Thinking about your classmates who had a similar assignment/project/report this term, do you think your experiences writing in computer science are similar to or different than other peoples' experiences? Why do you think this is the case?

- Did you learn any particular technical writing skills this semester?

  – How did you learn that?

  – How do you know how well you've mastered that skill?

  – Was your experience learning that skill positive, negative, a mix of both? Why?

- Do you remember anything in your current computer science course syllabi about technical writing? If so, what?

- Tell me more about a time in one of your CS classes this term when you received some feedback on your written work.

  – What kind of expertise do you think the person giving you feedback had? Was their feedback useful to you? If no, what might have been more useful?

- What kinds of experts would you want to learn technical writing from?

- We just talked about different kinds of expertise, but I'm wondering how do you personally describe effective technical writing? How would you measure or evaluate if a technical document is written effectively? What if you're the author?

- When you join a new community (for example at a new job), how do you learn about their norms and expectations for technical writing?

- Do you find it easier or more difficult to write with some people rather than others than others? Why or why not?

- I'm interested in the social dynamics of writing with others and giving and receiving feedback. Can you think of a time when you gave or received difficult feedback? Or a time when the social aspects of giving feedback prevented you from giving the feedback or changed the feedback that you gave?

- I asked about your experiences doing technical writing with people who are English language learners in a reflective journal and I'd like to know more about that. Can you tell me about a time that you wrote something with someone (or some people) who was an English language learner?
- How do you decide which pieces of writing to emulate?
- Do you find it easier or more difficult to write for some audiences rather than others than others? Why or why not?
- In general, the participants in this study believe that technical writing is important but don't think that their peers think it's important. Is there anything about your experiences or background that you think might lead you to have a different opinion than others?
- Some participants mentioned a divide between that gets widened at school where people who are good technical writers and communicators continue to get better and others don't. Do you think this happens? Why do you think this happens?
- Another thing that some people mentioned when we were discussing whether technical writing is valued and do companies value it, was a divide between the STEM fields and the humanities fields and that society values STEM fields more, and that that might be a reason why technical writing is valued less as part of STEM field. What you think about that?
- Why did you volunteer for this study?
- Do you think you learned anything through participating in this study? If so, what?
- Do you think participating in this study has affected your beliefs about technical writing? Or your technical writing experiences this term?

- Is there anything else you'd like to tell me?
- Is there anything I should have asked about?

Appendix D

Member Checks: Summary of Themes

My research study generated preliminary findings and themes related to the following questions:

1. How do students describe their experiences learning and doing technical writing with others? What are their reflections on these experiences?

   Findings:

   - Students learn from community members' technical writing by reading and emulating examples
   - Students reflect thoughtfully on their own technical writing in relation to writing of peers, mentors, instructors
   - Students understand that diverse community members have different needs for technical writing and appreciate learning from community members who have diverse academic and professional experiences

   Related themes:

   - Students learn technical writing implicitly via examples
   - Students have a keen awareness of audience for technical writing

- Students reflect thoughtfully on their or others' technical writing
- Technical writing is a social and collaborative endeavour
- Diverse community support is appreciated

2. How do course assessment practices affect students' learning of technical writing?

   Finding:

   - Students believe technical writing is important but prioritize learning activities that will be (more heavily) assessed

   Related themes:

   - Students have conflicting desire to learn technical writing

3. In what ways do students hold perspectives that may be barriers to their learning of technical writing?

   Participants shared their perspectives, or the perspectives of others in the field. Commonly held beliefs were:

   - Technical writing is less important than other technical skills
   - Computer scientists are not strong communicators
   - Writing is more valued in non-technical fields than it is in computer science

   Related themes:

   - Technical writing is less important than technical skill in computer science
   - Negative computer science stereotypes affect perspectives about technical writing