Embedding and Scaling Writing Instruction Across First- and Second-Year Computer Science Courses

Lisa Zhang
University of Toronto Mississauga
Mississauga, ON, Canada
lczhang@cs.toronto.edu

Bogdan Simion
University of Toronto Mississauga
Mississauga, ON, Canada
bogdan@cs.toronto.edu

Michael Kaler
University of Toronto Mississauga
Mississauga, ON, Canada
michael.kaler@utoronto.ca

Amna Liaqat
University of Toronto Mississauga
Mississauga, ON, Canada
a.liaqat@mail.utoronto.ca

Daniel Dick
University of Toronto Mississauga
Mississauga, ON, Canada
daniel.dick@mail.utoronto.ca

Andi Bergen
University of Toronto Mississauga
Mississauga, ON, Canada
andi.bergen@utoronto.ca

Michael Miljanovic
Ontario Tech University
Oshawa, ON, Canada
michael.miljanovic@ontariotechu.ca

Andrew Petersen
University of Toronto Mississauga
Mississauga, ON, Canada
andrew.petersen@utoronto.ca

ABSTRACT

Writing skills are often considered unimportant by computer science students and were under-emphasized in our curriculum. We describe our experience embedding CS-specific writing instruction at scale in most of our large, core, first- and second-year Computer Science courses, each with 300-800+ students. Our approach is 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. We outline the institutional support and organization crucial to a project of this scale. In addition, we report on a survey collecting student perception of the writing instruction/assessment. We reflect on quantitative and qualitative evidence of success, as well as the challenges that we faced. We believe that many of these challenges will be common across institutions, particularly those with large courses.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**; **CS1**.

KEYWORDS

written communication, WAC, WID, WTL, CS Education

ACM Reference Format:
Lisa Zhang, Bogdan Simion, Michael Kaler, Amna Liaqat, Daniel Dick, Andi Bergen, Michael Miljanovic, and Andrew Petersen. 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 V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569729

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

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569729

1 INTRODUCTION

Instructors and practitioners agree that writing is important in computer science, and the ACM's Computing Curricula 2020 contains 6 (of 88) communication competencies [8]. CS graduates write documents from a variety of genres, including software specifications, resumes and cover letters, and bug reports [3, 5]. Nevertheless, students struggle with technical writing [11, 26, 29]. Convincing students of its importance is challenging [3], and developing writing skills requires class time that reduces technical content [1].

We describe our experience embedding writing instruction and assessment in five first- and second-year computer science courses at a large, research-intensive North American university. Our effort is informed by the Writing Across the Curriculum (WAC) [2, 10, 24] and Writing In the Disciplines (WID) [12] approach to writing instruction. Each course and assessment introduces students to a genre that they may encounter in their career, including documentation, user guides, take-home interview email responses, and proofs. Our curriculum change supplements the typical approach of dedicated writing courses taught by writing specialists and scholars. Supplementing such instruction helps students to learn discipline-appropriate ways to communicate technical content. Moreover, we hope that having a community of Computer Science instructors will help convince students that *all of us* believe writing to be important.

Our contribution is novel, to the best of our knowledge, due to its scale: 5 courses with enrollments ranging from 300 to 800+ students, supported by 9 instructors, a writing specialist, and 13 writing teaching assistants. We outline the institutional support and project structure that supports this scale in Section 3. We summarize the evidence of success in Section 4, including a survey aimed to understand the student perception of the curriculum change. We analyze the challenges of introducing writing instruction/assessment at this scale. We believe that many of these challenges will be common across institutions, particularly those with large courses.

2 RELATED WORK

Since the 1980's [13, 30] and 1990's [17, 23, 31], computer science instructors have used a variety of approaches in discipline-specific writing instruction [14]. Falkner and Falkner [6] describe a CS1/CS2 communication skills course that uses the WAC model, blending a mixture of Writing to Learn (WTL) and Writing in the Discplines (WID) activities. French [9] describes a similar communication-intensive CS course. We moved away from a "CS communications" course with the introduction of an institution-wide first-year "Academic Writing" course, and—like Fell et al. [7], Anderson et al. [1], Pollock [28] and Burge et al. [4]—build upon communication-related learning objectives already in the core CS courses. Hoffman et al. [15], Jones et al. [18], and Isomöttönen et al. [16] have also proposed writing activities using a WTL model to help students learn CS concepts. Instructors have also used creative, discipline-specific tools to improve written communication skills, including Jupyter Notebooks [32] and agile development [22].

Kortsarts et al. [21] and Zilora and Hermsen [33] both describe collaborating with English faculty members to integrate writing into computer science courses. Likewise, Pesante [27] provides suggestions for CS instructors teaching writing, including "talk to members of the English department" to establish new relationships.

Like these works, we aim to be integrative, collaborative, and disciplinarily appropriate. However, our contribution differs in the scale at which we operate. Our report summarizes the resource and logistical aspects of running a project at scale. It also collects evidence from a variety of sources including a student survey, an independent writing assessment, and instructor assessment.

3 WRITING INSTRUCTION AND ASSESSMENT

This section outlines the curriculum change involving additional writing instruction and assessment. We summarize the institutional support and roles (Section 3.1), the shared learning objectives and rubric (Section 3.2), and outline the writing instruction and assessment in each course (Section 3.3). Finally, we describe a survey we conducted to understand student perception of the curriculum change (Section 3.4).

3.1 Institutional Support and Roles

Our institution's Writing Development Initiative (WDI) program provides financial and other forms of support for projects to enhance the use of discipline-appropriate writing in core courses across the curriculum and across the disciplines. The initiative supports projects that involve enhanced feedback on writing, creating new writing assignments, and/or providing enhanced instruction in writing. The financial support from the initiative gave us 847 hours of teaching assistant time (15-20min/student/course) to provide more writing instruction and assessment across our courses. Writing TAs received training to understand WID, WAC, and how to give effective writing feedback.

As part of this institutional support from the WDI, we were supported by a **writing specialist**, a faculty member from the university's Writing Centre. The writing specialist critiqued the initial proposal, set up the writing TA training referred to above, assisted with development of rubrics and assignment criteria, helped facilitate "benchmarking" or "norming" sessions for the TAs as part of the marking process, held a small number of writing office hours before key deadlines, oversaw the outside assessment of writing produced in the courses, and generally served as a point of contact for input regarding writing pedagogy issues.

The **teaching assistants** assessed student writing and provided writing feedback. Writing TAs were shared across various courses, and received 7 hours of paid training on WAC and writing instruction by the writing specialist. Due to our scale, our department relies on undergraduate teaching assistants in most of our courses. We foresaw the challenge of hiring and training teaching assistants who can provide quality writing feedback to CS students. We hired 3 computer science writing TAs in the Fall term and 10 in the Spring term. Of these TAs, 1 spring term TA was a graduate student in Computer Science and Education, and the remaining TAs were all undergraduate Computer Science students. Sharing writing TAs across multiple courses helped us maintain consistency through the writing grading across our courses and centralize TA management.

The project and resources were centrally managed by **project coordinators** who are Computer Science faculty. The coordinators proposed the project with instructor and departmental support, secured funding, and managed the CS writing TAs. The coordinators also provided support to instructors in developing teaching materials and assessment rubrics to make the project cohesive.

The individual **course instructors**, who are Computer Science faculty, were responsible for developing instructional materials for discussing writing in the context of their course. They were invited to the same training on WAC and writing instruction as the TAs. The course instructors developed the assignments, adapted the marking scheme to their courses, and provided grading instructions.

3.2 Learning Objectives and Grading Rubric

We identified the following learning objectives as communication skills required of a junior software intern or a junior research assistant. By the end of second year, students should be able to:

(1) Communicate effectively the intended and actual software behavior,
(2) Communicate effectively the approach to solving a computational problem,
(3) Structure writing in effective and expected formats, and
(4) Use appropriate terms and tone for the audience.

We used these objectives to guide both the genres that students are expected to write and the grading rubrics used across all our courses. The instructors worked together, with a writing specialist, to create a shared rubric targeting three areas of concern:

(1) **Structure and Organization**, which includes document and paragraph level organization. (LO 3)
(2) **Writing Mechanics and Clarity**, which includes issues with clarity, and other sentence-level issues, with a focus on CS-specific issues. (LO 1, 2)
(3) **Audience Expectations**, which involves issues related to the reception of the text by its target audience, including genre-based expectations. (LO 3, 4)

We deliberately kept the rubric focused to make assessment scalable for our TAs. The rubric items were also informed by our prior work Munir et al. [25]. The specifics of each of these rubric items depended on the particular assignment in each course and were

decided by the instructors. For example, using personal pronouns is an "Audience Expectation" issue when writing a *man* page, but is reasonable in an email to an employer. Having paragraphs with topic sentences is a good organizational tool in some genres, while headings and sub-headings were expected in others.

We decided not to explicitly focus on grammatical issues; they were considered, but only in the context of their effects on clear expression of the author's ideas. We adopt this approach to help enhance the usability of the feedback for our students who lack grammatical competence (e.g., English Language Learners, amongst others). It also encourages TAs, who are not necessarily trained in formal grammar, to give more feedback on higher-level issues affecting clarity, particularly CS-specific writing issues. The writing specialist supporting our project was supportive of this approach; specifically, they have found that ELL students will not infrequently feel that the feedback they get is more directed at sentence level issues than that received by their native-speaker colleagues. By rooting the feedback firmly in a CS context, we hoped to avoid this.

TAs provided a grade for each rubric item, as well as two other kinds of feedback: inline annotations describing an issue in a particular word, sentence, or paragraph, and—for courses that require a resubmission—a summary of what aspect of the writing the student should focus on for resubmission. TAs could choose to write annotations themselves or use an already written annotations from a comment bank that we generated for use in each of the three areas. We phrased these predefined annotations from the perspective of a reader, rather than as prescriptive commands [19]. Some examples of items from the comment bank include: "This doesn't seem to be the correct transition expression for what you are expressing here (Structure/Organization)", "I'm not sure what this pronoun refers to (Writing Mechanics)", and "This technical term needs to be defined when it is first used (Audience Expectations)".

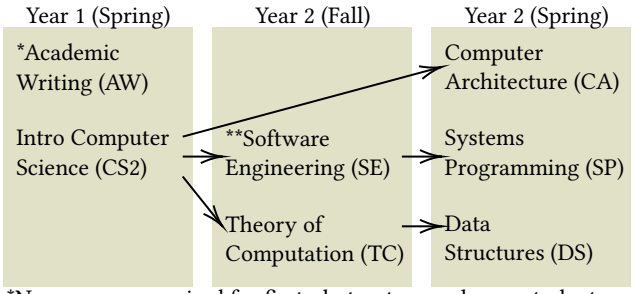
In the first-year course (CS2), the resubmitted work was regraded using the same 3 rubric categories. In the second-year courses that required a resubmission, the resubmitted text was graded for substantial completion, i.e., whether the student addressed the concerns described in the feedback. The resubmission provides an incentive for students to review the feedback provided by the TAs.

3.3 Courses and Assessments

Our core first- and second-year CS courses are shown in Figure 1. We added writing instruction in the following courses:

Year 1 Introduction to Computer Science (CS2), Spring 2022. We used 30 minutes of lecture to discuss structure and organization, transitions, unclear pronoun antecedents, descriptive verbs, and the expectations of a technical reader. Students wrote documentation intended for a technical audience for three Python classes, received TA assessment and feedback, and submitted a revision.

Year 2 Theory of Computation (TC), Fall 2021. We created a video module describing the steps in writing a proof, targeting issues like defining terms, using transition expressions, avoiding unclear pronoun antecedents, and refining one's work. In a lecture exercise, students identified logical and writing issues in sample proofs. Then, TAs assessed and provided feedback on the writing in one question in each of the three problem sets.



*New course; required for first-, but not second-year students
**No writing instruction/assessment this year

Figure 1: Core First- and Second-Year Courses. Arrows indicate and pre-requisites.

Year 2 Data Structures (DS), Spring 2022. We used 50 minutes of lecture to discuss logical ordering of ideas, use of topic sentences, sentences reading well "left-to-right" [20], and annotation of symbols with their "types" [20]. Students wrote a response to a take-home interview problem in the form of an email intended to be read by a technical audience. Students received TA assessment and feedback, and used this feedback to submit a revision.

Year 2 Systems Programming (SP), Spring 2022. We introduced the format of a *man* (manual) page, so that students were familiar with the genre and the expected content, organization, and tone of such documents. Students then wrote a *man* page. Students received TA assessment and feedback, and used this feedback to submit a revision in the form of a *diff* file.

Year 2 Computer Architecture (CA), Spring 2022. We ran a 20-minute collaborative exercise in a lab setting to identify common elements of user documentation and to illustrate the importance of writing for a target audience. Students then wrote a user manual, intended for a general (non-technical) audience, for a piece of software that they built.

3.4 Student Survey

At the end of term, we conducted a student survey to capture student perception on the writing initiative. Aside from logistics questions such as which writing intervention courses they were enrolled in, and whether they submitted (and resubmitted) their work, we used a combination of Likert-scale and free-form questions, which aimed to assess the student perception on the following:

(1) **Feedback Utility and Actionability**: whether the writing feedback was helpful for improving their writing, what the student felt that they learned, and which aspects of CS writing the student became aware of as a result.
(2) **Communication Improvement**: the extent to which the student perceived that the writing instruction and writing assignments improved their ability to communicate (Likert scale). This was broken down into four questions corresponding to the four learning objectives.
(3) **Writing Improvement**: how students rated their writing abilities going into and after the course (Likert scale).
(4) **Future Improvement Opportunities**: what students would like instructors to change in teaching writing in CS.

Table 1: Student writing grade across the courses and assessments. Here, n is the number of submissions. The Structure, Writing Mechanics, and Audience Expectation scores are each out of 4 points. Total writing grade is out of 12 points; substantial completion grade is presented as a percentage.

Assessment	n	Struc.	Mech.	Audience.	Total
CS2	601	3.03	2.95	3.22	9.20
CS2 (Resubmit)	294	3.59	3.47	3.80	10.87
TC PS1	424	3.43	3.12	3.60	10.15
TC PS2	404	3.50	3.21	3.67	10.38
TC PS3	402	3.65	3.63	3.60	10.87
DS	241	3.31	2.99	3.44	9.74
DS (Resubmit)	206	-	-	-	85%
SP	315	-	-	-	8.55
SP (Resubmit)	237	-	-	-	77%
CA	199	-	-	-	9.23

We received 86 survey responses from first-year students and 49 responses from second-year students. Of the 86 first-year responses, 82 respondents reported submitting the writing assessment and 45 reported resubmitting after receiving feedback. All second-year student respondents submitted at least one writing assessment.

We performed thematic analysis on the survey responses. Three coders coded approximately 15% of the responses together to develop a codebook. The coders then independently coded the same subset of responses, which were compared for agreement. Disagreements were resolved through discussion. Once the codebook was stable, a fourth coder evaluated the codebook to to check consistency of codes and standardization. The remaining responses were split between the three coders for independent coding. A final review and discussion was performed to ensure consistency.

4 EVIDENCE AND PERCEPTION OF SUCCESS

In this section, we discuss quantitative and qualitative evidence of success due to the curriculum change. While some evidence of quantitative measures of improvement such as grades are present, we do not present this as the sole indicator of success. Rather, we wish to provide a holistic picture of the curriculum change impact both in terms of writing improvement evidence, as well as the perceived experience of students and instructors.

4.1 Assessment of Student Writing

Student Writing Grades Table 1 summarizes writing grades across courses. Within the same course (CS2 and Theory) the grades improved. We initially suspected that this is due to weaker students dropping the course, but this trend persisted amongst students who submitted all assessments in the same course. In "Theory", the average writing grades among the 378 students who submitted all three problem sets were 10.17, 10.42, and 10.94. In "CS2", the average writing grades among the 262 students who submitted both an initial version and the resubmission were 9.07 and 11.10. The student grades provide some evidence, from the TAs perspective, that student writing improved.

Independent RA Assessment Our institution's WDI program hired a writing Research Assistant (RA), supervised by the writing specialist, to independently assess the changes in student writing. Rubrics used were based on those used in the courses and were created by the writing specialist. These rubrics were used to assess samples of the writing produced in the various courses: samples were drawn from roughly 10% of the students (randomly selected) from each course and consisted of the original submissions of the main writing assignment and the resubmission (or later assignment). In addition to the completed rubric, the RA submitted an overview of their observations of the changes in student writing between the initial submissions and resubmissions for each course.

The RA assessment showed that the in-class interventions, alongside TA feedback, had a demonstrable impact:

- *Colloquial/overly personal language was largely eliminated* in the second submission across all courses. For example, several of the initial submissions seen in the "Theory" course contained direct addresses to the reader (e.g., "...you can reverse the moves you just made..."). In contrast, the style and content of the second submissions shifted away from this colloquial writing style. Furthermore, writing conventions specific to computer science (e.g., the usage of "we" in place of "you" or other direct addresses to the reader) were better employed across the majority of submissions.
- *Students used a greater variety of transition expressions* following TA feedback in all the courses examined. This was particularly common in "Systems", wherein the overall readability and organization of a number of submissions was improved due to improved use of transition expressions.
- *Issues related to clarity of pronoun antecedents were generally fairly rare*, occurring less often than in comparable courses where this issue was not highlighted by the instructor(s). Students tended to provide necessary context when using terms like "it" or "them".
- *TA feedback had a noticeable impact on aspects of overall organization and clarity.* Post-feedback submissions contained a greater range of informative subheadings, and larger sections/run-on sentences were broken up.
- *Grammatical issues were prevalent and did not show consistent improvement across submissions.* Most commonly, students struggled with the plural and possessive forms of nouns. Spelling issues were largely restricted to technical and/or unfamiliar words (e.g., delimiter). While a subset of students in each course fixed some of the observed spelling/grammatical issues between submissions, no consistent pattern emerged across submissions.
- *Students often used "pseudocode" style writing.* In assignments involving explaining code (particularly CS2 and "Systems"), students occasionally wrote in a "pseudocode" style. That is, most of their submission was written as a series of disconnected statements articulating what each line of code does, usually of the format "The [function] does [functionality]." (e.g., "...insert() works by getting the initial index *index* by hashing the key *k*.").

These RA perceptions corroborated discussions with TAs about student writing. Two TAs in the "Theory" course also identified that students were using more transition phrases. TAs in the Spring

Table 2: Student rating of their writing abilities before/after the course(s). Both figures collected at the end of term.

Year	V. Weak	Weak	Competent	Strong	Excellent
Y1 before	15%	18%	32%	24%	11%
Y1 after	3%	5%	34%	35%	24%
Y2 before	2%	14%	39%	41%	4%
Y2 after	0%	6%	29%	59%	6%

term noted that students were diligent in making changes suggested by the TA, and noticed genuine effort in incorporating feedback.

4.2 Student Perception of Success

Feedback was largely helpful (with caveats, see Section 5.2). In the student survey, 75% of first-year and 72% of second-year students answered positively (selected "Somewhat" or "Very Much") to the question, "To what extent did the feedback on your initial submission help to improve your writing in the resubmission?" More specifically, between 59% - 80% of respondents responded positively to the question about each of the four learning objectives (e.g., "To what extent did the instruction/assignment improve your ability to communicate intended and actual software behavior"). In the open-response questions, positive comments about the feedback discussed its role in improving the student's grade ("I followed all the issues pointed out from the submission of my first draft, and resubmitted and received an almost perfect score") and improve the draft ("After fixing these issues, I feel that my assignment had much better flow.", "The feedback helped me edit and rewrite a better overall version of the problem set."). We discuss the issue of feedback quality more thoroughly in Section 5.2.

Students perceived improvements in their writing. Table 2 shows the student rating of their writing abilities. About 48% of first-year and 35% of second-year students had a higher ranking of their writing abilities after the course(s) compared to before the course(s) with the writing instruction/assessment. Although the improvement appear to be reduced in the second-year, we noticed that the survey responses from second-year students appear to be more nuanced and show a greater depth of understanding of the challenges of writing. A second-year student writes, "I learned that it is more difficult than I thought it is to explain CS in non-CS related terms. You truly have to understand how your code works to be able to do it, alongside being clear and concise."

Students' responses suggest a change in the perception of writing. The survey provides some evidence that student's perception of writing is changing. Students reports that writing was challenging ("I learned that documenting code is both very useful but also relatively difficult to do, especially when you finished a large coding project and just want to be done with it"), and two students reported that writing was enjoyable ("personally enjoyed the writing portion, although I know others who might think otherwise"). Documentation was a new skill for 12 students; they reported positive attitudes towards learning this skill, and ten students described the importance of documentation skills in computing. In particular, 10 first-year students shared that the writing process helped them better understand their code and other CS concepts [16, 18] ("I learned that explaining our code is actually really important part of the debugging process because I realized that my algorithm was a bit off when I started putting it in words"). Finally, 7 students respondents linked developing writing skills with professional growth ("I was able to learn how to write emails professionally, which is great as I am currently applying for internships and would be a very useful skill").

Students' descriptions of what they learned aligned with our learning objectives. In particular, the "Audience Expectations" learning objectives resonated overwhelmingly. About half of the student responses reflected on the expectations of a reader/audience in one way or another ("I learned that I try to squeeze too much information and content into my sentences and that can ruin the readability of my work.", "It's important to make it organized in the aspect of how reader or user would refer to when they want to find out something", "I have to be extremely specific, and use pronouns as little as possible, and avoid using informal language").

4.3 Instructor Perception of Success

No student push-back. Going into the project, there were several instructor concerns regarding student buy-in and pushback. However, in the "Data Structures" course, students were engaged when writing was covered during lecture. In the "Architecture" course, students were able to effectively mimic similar documents and integrate visual aids into their documents. All instructors noted few or no questions about writing in the course message boards, and little or no student feedback on course evaluations.

No excessive regrade requests. Instructors were initially concerned about possibly high volume of student requests to regrade the writing work, but there were only a handful (<10) of remark requests in each course. In most of these cases, instructors agreed with TA grading and feedback.

5 (REMAINING) CHALLENGES

5.1 Teaching Assistants

TA Staffing Our department generally hires undergraduate students, and hiring undergraduate students to be writing TAs is particularly challenging. We coincidentally mitigated this challenge by supervising undergraduate research students on the topic of writing in Computer Science the year prior to the curriculum change, thus having some more experienced TAs than we otherwise would have. While TA training is provided by the writing specialist and the WDI, we note that experienced TAs tend to provide better feedback.

Grading Consistency With our need to hire undergraduate and new TAs, grading consistency was an issue that we anticipated. To mitigate this issue, the writing specialist ran "benchmarking" sessions where writing TAs discussed the grading of a few submissions. Benchmarking was effective enough at ensuring grading consistency that we had only a small number of remark requests.

Grading Turnaround Grading turnaround time for writing TAs was a challenge, especially as new writing TAs learned to provide writing feedback. This was a challenge the project coordinators did not fully anticipate. Since grading consistency was not an issue in our case, we intend to explore trading off having more unique TAs and thus fewer grading hours per TA.

5.2 Writing Feedback Quality

While most of the students found the writing feedback to be "Somewhat" or "Very" helpful (Section 4.2), students also identified the actionability and type of feedback to be areas for improvement.

Feedback does not suggest a fix. 10 respondents reported that feedback was confusing or unclear. Students were unsure of how to act on the feedback to improve their writing ("[The feedback] told me what I did wrong, but it had very little information as to how to fix it"). We noticed that some of our pre-written annotations, particularly those related to structure, point out an issue without a description for its fix. Although writing office hours with the writing specialist were available for students, these office hours tended to be under-used, particularly in the resubmission stage.

Feedback focuses on sentence-level issues. 19 respondents reported that feedback they received focused on sentence level errors in their writing rather than a holistic assessment ("I generally received more feedback on grammar than the explanation of my logic"). We corroborated this observation by counting the number and type of predefined annotations used by TAs. In CS2, 36% of predefined annotation used were for writing mechanics issue and 57% for audience expectations issues, compared to only 7% for structure and organization issues. In the second-year courses (TC, DS, and SP), 53-68% of the pre-defined annotations use were for writing mechanics and 14-21% for audience expectations issues, compared to 18-26% for structure and organization.

5.3 Responding to Subjective Feedback

In two second-year courses, instructors noted student disagreement with TA feedback. Instructor discussions with students during the term suggests some level of student resentment toward feedback that they consider "subjective". We see 2 instances of this in the survey responses ("my documentation got the job done, even though it was not what was expected", "TAs snipped lots of marks for grammar at places where they did understand the point I was trying to convey like capitalisation errors and and simple stuff like connecting two paragraphs or using points"). We believe there is an opportunity to provide support to students in receiving, addressing, and learning from subjective feedback that they may disagree with.

5.4 Writing Evaluation Structure

Students prefer more, smaller, integrated writing assessments. Students overwhelmingly preferred having more integrated writing assessment throughout a course. In the survey, 20 respondents requested more frequent practice and assignments that were integrated throughout the course. Second-year students pointed to the "Theory" course as a successful model, where there were multiple writing assessments throughout the course.

Students appreciate being able to resubmit their work. Further, students appreciated the opportunity to resubmit their work. One first-year respondent said that the opportunity for resubmission, "...helped reduce the stress and allowed me to experiment with certain communication styles."

5.5 Writing Pedagogy

Provide more examples. Overwhelmingly, students asked for more examples of quality work. In the survey, 31 participants stated they would like more examples or templates to model their writing ("Maybe give more examples of well-written documentation, or encourage us more to look a python documentation for a better idea of what to do"). In particular, 13 first-year respondents asked for clearer writing criteria. Despite our best efforts in providing instructions and rubric, first-year students described struggling with understanding the expectations of the writing assessment ("not knowing exactly what is to be expected is a bit challenging", "most of us had no experience writing documentation before"). During the term, first-year students asked for templates to use. However, we hesitated to provide one, preferring not to limit students to one particular template. We believed that having a variety of successful prior work would be helpful for students encountering this new genre for the first time.

Demonstrate the writing process. Two first-year respondents and three second-year respondents requested demonstrations of the writing process ("maybe [instructors] could use more examples of code, and how they would go about documenting them", "Demo how to plan writing to ensure that it is concise, precise and follows a relevant logical flow").

Course-Specific Objectives Instructors identified several writing issues in specific courses. For example, in the "CS2" and "Systems" courses requiring code documentation, students tend to write in a "pseudocode" style and directly translate code line-by-line into English. Conciseness was an issue in the "Data Structures" course. In the "Architecture" course, students used variables and tools introduced in the course without understanding these terms to be jargon that needs to be defined or avoided for a non-technical audience.

6 CONCLUSION

Although there are remaining challenges, all stakeholders hold positive feelings about the approach of *embedding writing instruction* across the first- and second-year CS courses and *centralizing project coordination*. However, we disagree on which factor was most essential to this project and that is worth emphasizing. The instructors believe that the central coordination of the project was key—otherwise adding writing instruction and managing writing TAs would be an overwhelming time commitment. The project coordinators feel that the institutional support and collaboration with the writing specialist was key—otherwise training the TAs and being confident about the writing pedagogy would not be possible. The writing specialist feels that the commitment of the community of instructors to prioritize writing was key—otherwise there would be no interest in the project or momentum to sustain it. This circular argument underscores the real takeaway: scaling writing instruction across a program requires commitment from a broad set of experts. As voiced by our students, that commitment enables writing instruction that feels authentic and results in real changes to their writing abilities.

ACKNOWLEDGMENTS

We thank the UTM's Writing Development Initiative, Office of the Dean, the Robert Gillespie Academic Skills Centre, and the Institute for the Study of University Pedagogy for the financial and other forms of support. We appreciate Paul Zhang for his preliminary data analysis, and Andrew Wang for the survey setup.

REFERENCES

[1] Paul V. Anderson, Sarah Heckman, Mladen Vouk, David Wright, Michael Carter, Janet E. Burge, and Gerald C. Gannod. 2015. CS/SE instructors can improve student writing without reducing class time devoted to technical content: experimental results. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. IEEE, 455–464.

[2] Charles Bazerman and Joseph Little. 2005. *Reference guide to writing across the curriculum.* Parlor Press LLC.

[3] Theresa Beaubouef. 2003. Why Computer Science Students Need Language. *SIGCSE Bull.* 35, 4 (Dec. 2003), 51–54. https://doi.org/10.1145/960492.960525

[4] Janet E. Burge, Gerald C. Gannod, Paul V. Anderson, Kara Rosine, Mladen A. Vouk, and Michael Carter. 2012. Characterizing communication instruction in computer science and engineering programs: Methods and applications. In *2012 Frontiers in Education Conference Proceedings*. IEEE, 1–6.

[5] Robert F. Dugan Jr and Virginia G. Polanski. 2006. Writing for computer science: A taxonomy of writing tasks and general advice. *Journal of Computing Sciences in Colleges* 21, 6 (2006), 191–203.

[6] Katrina Falkner and Nickolas J.G. Falkner. 2012. Integrating communication skills into the computer science curriculum. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 379–384.

[7] Harriet J. Fell, Viera K. Proulx, and John Casey. 1996. Writing across the computer science curriculum. *ACM SIGCSE Bulletin* 28, 1 (1996), 204–209.

[8] CC2020 Task Force. 2020. Computing Curricula 2020: Paradigms for Global Computing Education. (Dec 2020). https://www.acm.org/education/curricula-recommendations

[9] Jean H. French. 2012. Evaluating a communication-intensive core course in the CS curriculum. *Journal of Computing Sciences in Colleges* 28, 2 (2012), 197–209.

[10] Toby Fulwiler and Art Young. 1982. *Language connections: Writing and reading across the curriculum.* ERIC.

[11] Ernie Giangrande. 2009. Communication Skills in the CS Curriculum. 24, 4 (April 2009), 74–79.

[12] Mary Goldschmidt. 2014. Teaching writing in the disciplines: Student perspectives on learning genre. *Teaching and Learning Inquiry* 2, 2 (2014), 25–40.

[13] Janet D. Hartman. 1989. Writing to learn and communicate in a data structures course. *ACM SIGCSE Bulletin* 21, 1 (1989), 32–36.

[14] Wu He, Harris Wu, Li Xu, and Kurt Maly. 2015. Scaffolding to Improve Writing Skills in Computer Science Literacy Course. In *Society for Information Technology & Teacher Education International Conference*. Association for the Advancement of Computing in Education (AACE), 934–937.

[15] Mark E. Hoffman, Timothy Dansdill, and David S. Herscovici. 2006. Bridging Writing to Learn and Writing in the Discipline in Computer Science Education. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA) *(SIGCSE '06)*. Association for Computing Machinery, New York, NY, USA, 117–121. https://doi.org/10.1145/1121341.1121379

[16] Ville Isomöttönen, Aletta Nylén, and Ville Tirronen. 2016. Writing to learn programming? A single case pilot study. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. 140–144.

[17] Paul M. Jackowitz, Richard M. Plishka, and James R. Sidbury. 1990. Teaching writing and research skills in the computer science curriculum. *ACM SIGCSE Bulletin* 22, 1 (1990), 212–215.

[18] Bryan A. Jones, M. Jean Mohammadi-Aragh, Amy K. Barton, Donna Reese, and Hejia Pan. 2015. Writing-to-Learn-to-Program: Examining the need for a new genre in programming pedagogy. In *2015 ASEE Annual Conference & Exposition*. 26–1779.

[19] Michael Kaler, Jonathan Vroom, and Christoph Richter. 2022. Drawing on Readerly Intuition in Sentence Level Feedback. *Discourse and Writing/Rédactologie* 32 (2022), 73–102.

[20] Donald E. Knuth, Tracy Larrabee, Paul M. Roberts, and Paul M. Roberts. 1989. *Mathematical writing*. Number 14. Cambridge University Press.

[21] Yana Kortsarts, Adam Fischbach, Jeff Rufinus, Janine M Utell, and Suk-Chung Yoon. 2010. Developing Oral and Written Communication Skills in Undergraduate Computer Science and Information Systems Curriculum. *Information Systems Education Journal* 8, 30 (2010), n30.

[22] Clifton Kussmaul. 2005. Using agile development methods to improve student writing. *Journal of Computing Sciences in Colleges* 20, 3 (2005), 148–156.

[23] Gary McDonald and Merry McDonald. 1993. Developing oral communication skills of computer science undergraduates. In *Proceedings of the twenty-fourth SIGCSE technical symposium on Computer science education*. 279–282.

[24] Susan H. McLeod. 1989. Writing across the curriculum: The second stage, and beyond. *College Composition and Communication* 40, 3 (1989), 337–343.

[25] Rehmat Munir, Francesco Strafforello, Niveditha Kani, Michael Kaler, Bogdan Simion, and Lisa Zhang. March 2022. Exploring Common Writing Issues in Upper-Year Computer Science. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*.

[26] Tom Nurkkala and Maria Gini. 2002. *An Assessment of the Writing of Undergraduate Computer Science Students.* Technical Report. Department of Computer Science, University of Minnesota. 20 pages.

[27] Linda H. Pesante. 1991. Integrating writing into computer science courses. In *Proceedings of the twenty-second SIGCSE technical symposium on Computer science education*. 205–209.

[28] Lori Pollock. 2001. Integrating an intensive experience with communication skills development into a computer science course. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*. 287–291.

[29] Carol Stewart, Alison Wall, and Sheryl Marciniec. 2016. Mixed signals: do college graduates have the soft skills that employers want?. In *Competition Forum*, Vol. 14. American Society for Competitiveness, 276.

[30] William Taffe. 1989. Writing in the computer science curriculum. *Writing across the Curriculum* 1, 1 (1989), 17–22.

[31] Henry M. Walker. 1998. Writing within the computer science curriculum. *ACM SIGCSE Bulletin* 30, 2 (1998), 24–25.

[32] Alistair Willis, Patricia Charlton, and Tony Hirst. 2020. Developing students' written communication skills with Jupyter notebooks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1089–1095.

[33] Stephen J. Zilora and Lisa M. Hermsen. 2007. Take a WAC at writing in your course. In *Proceedings of the 8th ACM SIGITE conference on Information technology education*. 167–174.