ARTICLE

Sustainable Collaboration: A Co-taught, Client-based Course Sequence Integrating Computer Science and Technical Communication

Rebecca E. Burnett*
Georgia Institute of Technology
Andy Frazee*
Georgia Institute of Technology
Amanda K. Girard*
Georgia Institute of Technology
Liz Hutter*
University of Dayton
Halcyon M. Lawrence*
Towson University
Olga Menagarishvili*
Metropolitan State University

**Names are listed alphabetically to convey shared first-author roles in composing and editing.*

Abstract. This case study characterizes a client-based, capstone program integrating computer science (CS) and technical communication (TC). The interdisciplinary CS-TC program began with 50 students and 2 faculty; the current program involves 500–600 students, 8–9 collaborative faculty (half CS and half TC), and a full-time coordinator, and culminates in an end-of-semester public expo for the display and demonstration of student teams' client-based projects. A summary of the three periods of programmatic development is based on observations and a review of documents related to administration, students, faculty, and clients. The case focuses on issues related to stakeholders, collaboration, interdisciplinarity, and sustainability.

Keywords: Linked Courses, Integrated Courses, Client-based Projects, Computer Science, Technical Communication, Stakeholders, Collaboration, Interdisciplinarity, Sustainability

Contact author: rebecca.burnett@lmc.gatech.edu

More than a decade ago, the Division of Computing Instruction (DCI) and the Writing and Communication Program (WCP) at Georgia Institute of Technology (Georgia Tech) tackled a problem: Alums in the College of Computing, one of the university's fastest-growing colleges, expressed the strong opinion that a single, required service course in Technical Communication (TC) was not sufficient to prepare students for the intense communication that would be expected of them in the workplace. The alums wanted Computer Science (CS) students to have more experience with written, oral, and visual communication, and they wanted that TC education better integrated into the work of computer science.

In response to alum concerns, DCI proposed collaborating with WCP to re-envision teaching TC for CS students. Our collaborative, cross-college curricular innovation dismantled two stand-alone, three-credit courses and created four new courses, team-taught over two semesters to link CS and TC. The goal was to give students more experience in TC as well as experience directly and immediately relevant to their major.

This now well-established program provides a fertile space for a case study that examines factors influencing programmatic success and sustainability. Through this case study, we demonstrate that integration of TC and disciplinary instruction is marked by complex relationships with numerous stakeholders; collaboration among students, instructors, administrators, clients, and others; an interdisciplinarity that connects TC and disciplinary instruction; and a sustainability that is integrated into a virtually seamless whole.

This article is about how the whole has been created, explaining our methodology and then documenting the initiation and evolution from a linked to an integrated, interdisciplinary program. We use our program's history to explore four factors: stakeholder commitment, collaboration, curricular interdisciplinarity, and sustainability.

Case Methodology

We offer a case study of a 10-year program, from its inception through to its current success. Case studies are useful because, as Kay de Vries (2020) explained, they permit "description, exploration, and understanding of phenomena" in context (p. 42). Case studies not only synthesize information across time and space, but they also typically include rich description and analysis that might be constrained in other kinds of reporting (Alpi & Evans, 2019; Flyvbjerg, 2006, 2011).

Our case study brings together voices involved in innovation and decision-making throughout the program's development. As stakeholders and authors of this article, we represent many of the attitudes and actions as they occurred, not just our recollections of programmatic development. Our case study reflects a consistent subject and object (Thomas, 2011), developed from our collection and analysis of data and our reflection about the program. The subject of this case study is a co-taught, interdisciplinary, multimodal program. As stakeholders, we have extensive knowledge about the program that might be useful for other institutions. The object is our analytical framework—a complex network of actors involving people as well as concepts, organizational units, and actions, all characterized by changes over time (Aka & Labelle, 2021; Latour, 1996, 2005).

Case studies sometimes raise methodological concerns (Lindgreen, Di Benedetto, & Beverland, 2021), specifically about validity (accuracy) and reliability (consistency). We have addressed these concerns by, for example, documenting program development in a timeline (CS-TC Program Timeline, 2022), using multiple data sources, using quantitative and qualitative data, basing descriptions and examples on verifiable records, and triangulating data. We believe that a carefully conducted case study can offer insightful, multi-faceted understanding of a complex situation.

We recognize that no program can be lifted from one institution and adopted directly by another. We intend readers to adapt our experiences to their own situations. Colleagues creating their own integrated program not only need to consider a range of factors (e.g., type and size of their institution, their institutional culture, organizational and disciplinary structure, curricular flexibility, faculty willingness to engage in planning, support staff cooperation, and institutional legal support; Burnett et al., 2019) but also reference national discussions about creating programs with partners outside the academy (Bridgeford & St.Amant, 2017; Lancaster & Yeats, 2016).

We have selected representative examples, cross-checking each other to use consistent terms and to create a comprehensive description of this program's development from a range of artifacts:

- observations of administrators, program coordinators, and faculty
- documents to describe the program and its development, including, for example, administrative documents (e.g., course proposals, meeting agendas and minutes, and policy documents); student documents (e.g., syllabi, living schedule, and website); curricular documents (e.g., assignment sheets and rubrics); faculty documents (e.g., onboarding materials, course schedules, and correspondence); and client documents (e.g., a memo of client

expectations, project proposals, and project descriptions)
- official agreements, including memos of understanding (MOUs), intellectual property (IP) agreements, and nondisclosure agreements (NDAs)
- scholarly presentations and publications related to the program

This case study puts 10 years of programmatic history in perspective and offers observations that could be adapted by other institutions considering an integrated, interdisciplinary program. As with all case studies, this one is not all-inclusive. Instead, it depends on available documents, on observations and recollections of stakeholders, and, in this project, on our self- and peer interviews, triangulated because we have had three or more stakeholders recalling events at each of the three periods of the program.

Program Evolution

Our perspective about the evolution of an integrated CS-TC program is situated within communication in the professions (CIP), an approach with one foot in the classroom and one foot in the workplace. We discuss the impetus for the program and its three stages of development.[1]

Our focus on communication in the professions comes from our belief that students learn to be better communicators when they have an actual context, a defined audience, an explicit purpose, and multimodal options (e.g., Bourelle, 2015). For us, such communication includes written, oral, visual, collaborative, and nonverbal interactions, whether face-to-face, print, or digital, whether local or international. Our position is supported by colleagues who note the appeal of hiring STEM students who are effective communicators: "As workplaces become more interdisciplinary, team-based, and cross-cultural, communication competencies valued by industry and expected in entry-level employees [continue] to grow… [including] individuals who can comfortably interact with clients" (Hora et al., 2019, pp. 2222–2223). Various CIP actions demonstrate the CS-TC program:

[1] Prior to 2013, students across a number of disciplines at our institution were required to take a 300-level technical communication service course. Though students in these courses were sometimes assigned projects related to their major as a way to establish relevance, most of our technical communication service courses were not linked to a specific program in the university's other colleges. Only two programs (in business and in construction engineering) had permanent sections officially devoted specifically to their majors, with assignments focused on their majors' professional expectations. Periodically, units would request discipline-specific sections (including aerospace engineering, industrial design, industrial systems engineering, pre-health, and ROTC), but these were not fully developed and ongoing programs.

- Challenge misconceptions about the instrumental nature of professional communication.
- Address complex considerations including cognitive and psychological capabilities of audiences, ethics, and global boundaries with practical considerations such as technology, budget, and schedule.
- Demonstrate that clear objectives are helpful in creating and assessing activities and assignments (Cross & Wills, 2001).
- Present assignments that are "socially and culturally situated, necessarily rhetorical, and subject to critique" (Jones, Moore, & Walton, 2016).
- Reflect rapidly changing workplace practices, affecting the ability of professionals to transfer and adapt "practice knowledge" (Schreiber, Carrion, & Lauer, 2018, p. 2).

The resulting CS-TC program has had three phases of development: (1) the Early Years involved planning, piloting, and implementing linked courses; (2) the Middle Years involved fully integrating the curriculum and implementing a public-facing expo; and (3) the Established Years started with hiring a full-time coordinator to stabilize and sustain the integration. (See the CS-TC website for a table with details of the program's phases; CS-TC Program Timeline, 2022).

The Early Years

The initial planning committee included the Director of DCI, the instructor of the CS capstone course, and the WCP Director. This committee started exploring possibilities in spring 2012. In fall 2012, the WCP Associate Director and a new TC instructor joined the committee. An agenda from August 2012 shows that the goal was simple. CS requested discipline-specific sections of the existing TC service course (with some co-teaching as well as shared readings, assignments, and assessment). This goal rapidly evolved into a plan that made clear the committee was talking about developing new courses, not just offering variations of existing courses. These five colleagues met weekly during the 2012–13 academic year to design a new program (CS-TC Planning Committee and Program Coordinators, 2022).

The committee strategized ways to combine CS and TC. During part of the planning year (spring 2013), the CS instructor and the TC instructor piloted two joint CS-TC assignments (final project team presentation and post-project review) in one section of a standalone TC service course. The instructors co-created the assignments, the TC instructor introduced the assignments in class with the CS instructor present, and both assessed the resulting work. The goal was to create models and a process for shared assignments and assessment criteria. At the same time, the administrators worked to create programmatic policies and procedures:

- Sought approval for the four new courses by school, college, and university curriculum committees.
- Consulted with the registrar to schedule a common cohort of students and created a process to register students as teams for the second semester of the course.
- Considered criteria for hiring and pairing instructors.
- Negotiated TC faculty workload (moving from a 3:3 load with 60–75 individual students to a 2:2 load with 100 students on 20 teams).
- Agreed on policies such as attendance and participation.
- Informed upper-level administration and support personnel about the new program.
- Managed the program, including WCP's request for the College of Computing to fund a full-time coordinator.

Initially, WCP proposed seamlessly integrated courses in which all assignments fulfilled both CS and TC objectives and all tasks contributed to learning in both disciplines. However, CS saw complete integration as radical and risky. So we compromised: all major assignments were jointly CS-TC, but some smaller assignments were CS-only (introduced and evaluated by the CS instructor), and some were TC-only (introduced and evaluated by the TC instructor). Major joint assignments included project proposals, project specifications, feasibility reports, progress reports, and oral presentations. All joint assignments were evaluated by both instructors. Thus, the course sequence at the beginning of the program was one-third CS, one-third TC, and one-third integrated. This linked approach took several years to evolve to a fully integrated approach, as the benefits of such integration became clear.

To reflect the linked structure, some days became "CS-TC teaching days" (both instructors responsible for in-class activities), some days became "CS teaching days" (the CS instructor responsible for in-class activities), and some days became "TC teaching days" (the TC instructor responsible for in-class activities). The schedule was color coded to reflect the three types of teaching and assignments. However, both instructors were present in class on all teaching days.

During these Early Years, we implemented strategies that emphasized CS-TC balance:

- Both CS and TC parts of the linked courses were introduced in the first class session.

- Linking the courses required a shared physical classroom. Both instructors regularly discussed ways to share in-class time guided by schedule, topics, and assignments.
- Because the linked courses had to satisfy two sets of requirements, using separate syllabi was seen as necessary.
- Students' main source of information about the course was a living schedule created in Google Docs—easily and regularly updated. The living schedule presented day-to-day activities, including dates, course topics, readings, and links to assignment descriptions and other materials.
- Each course had its own LMS for submitting assignments, providing feedback, and assigning grades. An online platform (Piazza) was used for discussions and peer review of assignment drafts.
- To make connections for students during class, TC instructors often used CS-based examples to illustrate TC concepts.
- To reinforce CS-TC links, instructors used the same grading rubric to give students feedback on assignments.

Initially, all students were in face-to-face sections, but the instructors experimented with online and hybrid teaching. Two hybrid sections and one online section taught in years 4 and 5 were the focus of an IRB-approved mixed-methods study to compare face-to-face and hybrid sections (Burnett, Menagarishvili, & Frazee, 2019; Kmiec, Menagarishvili, & Longo, 2017a, 2017b; Menagarishvili, 2018; Menagarishvili, Frazee, & Burnett, 2022a, 2022b).

As the number of sections grew, we standardized onboarding of new instructors. In year 5, we began regular meetings for all continuing and new instructors (weekly meetings during the first month of the academic year; monthly meetings after that). A Google Drive folder with a template living schedule, sample assignment sheets, and sample agendas for every class meeting was used to discuss the courses. Discussions during these meetings facilitated interactions among all CS-TC teaching pairs. Finally, a CS-TC Course Context document was created to describe the goals, history, and content of the courses as well as recommended pedagogical approaches.

The Middle Years

Until 2015–16, the demarcations between CS and TC course content and instructor responsibilities limited both the ability of instructors to fully collaborate and students' understanding of the courses' integrated purpose. These demarcations communicated mixed messages, reinforcing students' assumptions about TC and CS work and the value of that work.

By year 5, the CS instructors came to recognize the enormous benefits from full integration, something they saw as risky 5 years earlier. Full integration resulted in a number of pedagogical and administrative changes. Though this change was strongly supported by the CS instructors, the TC postdoctoral fellows coordinating the program largely designed and implemented the curricular changes such as these:

- Course learning objectives were revised to reflect fully integrated expectations, leading to a new single syllabus.
- CS- and TC-only designations were eliminated, all assignments were now shared, and all class sessions were integrated.
- Existing assignments and assessment criteria were revised, and new assignments were introduced to reflect evolving industry standards. For example, a formal usability module was developed.
- Instructors negotiated more effective ways to share class time, teaching responsibilities, and assessment.

Because the institution's LMS had evolved, we could merge the separate CS and TC course sites into one, presenting students with a unified place for communication. Additionally, we established shared course policies about attendance, client feedback, team charters, and peer evaluations. As the program grew, WCP continued to request that the College of Computing fund a full-time coordinator.

In year 6, TC instructors proposed a public-facing expo to replace the second-semester software demos that teams previously presented to peers, faculty, and clients. In spring 2018, working with the College of Computing's Assistant Dean for Outreach, Enrollment and Community, the postdoctoral fellow coordinating the program created the first full-day Computer Science Junior Design Capstone Expo (thereafter referred to as the Expo). This Expo was intentionally different from existing capstone expos on campus so that it more closely reflected software industry events. Teams staffed their booths, demonstrating their project and answering questions from attendees. (For further details about the Expo, see KellyAnn Fitzpatrick, 2019. Fitzpatrick was a former program coordinator who interviewed another program coordinator and an event manager about the details of this now semi-annual event.)

Once the Expo had a visible presence on campus and with the workplace community, the College of Computing finally agreed that a permanent program coordinator be hired. A job description was agreed on by WCP and DCI; these units also drafted, revised, and their colleges signed an MOU designating responsibilities of each in relation to the new position. The person hired for the position was a TC instructor who knew the program well.

The Established Years

The current version of the program reflects more than a decade of development, leading to a well-established steady state: 500–600 students, typically 9-12 CS-TC faculty, and 100+ clients. Maintaining close relationships with CS faculty has led to regular updating of curriculum to match industry standards.[2] With each new instructor or new instructor pair, new insights are brought into the program.

The program's newly established, permanent, full-time coordinator (with a PhD in rhetoric, theory, and culture), funded by the College of Computing, balances pedagogical and administrative responsibilities. Pedagogically, the coordinator develops curriculum, applies disciplinary and workplace standards to courses, orients new instructors, and co-teaches several sections of the courses. The coordinator is employed by DCI but maintains close connections with WCP, being responsible, for example, for onboarding and professional development of new TC hires. Administratively, the coordinator assists with scheduling, organizes and facilitates stakeholder meetings, updates legal requirements, and plans and implements the Expo. Additionally, the coordinator recruits and vets clients and organizes the registration permit process. The coordinator takes a proactive approach to attracting clients and drawing attention to the CS-TC program by creating a website where projects are submitted. She also works with the university's legal office to streamline non-disclosure agreements for clients, students, and instructors. Further, she has sought other resources on campus to aid students with their IP rights. Finally, the coordinator maintains an MS Teams site as a virtual space for instructors and student teams in the course series.

This much-abbreviated narrative of the CS-TC program's development provides the foundation for our analysis of factors that webelieve provide insight about the success of the program: stakeholders,

[2] We recognize accreditation and assessment as important for programmatic creation, evaluation, and evolution. The Georgia Tech College of Computing (COC) has decided to not assess the CS-TC courses for accreditation purposes because CS undergraduate students are not required to take these specific courses to satisfy junior design degree requirements. Other course options to complete the degree include Create-X, VIP, and a research project. The COC assessment focuses on the CS3001 course, "Computing, Society, and Professionalism," a requirement that all CS majors must take and a prerequisite to the CS-TC Junior Design Capstone course series.

collaboration, interdisciplinarity, and sustainability. We situate each of these briefly in the literature before providing details that we hope effectively document this case study as well as provide direction for other institutions.

Stakeholder Commitment

Stakeholders (human actors) are involved in a complex programmatic network, including physical and digital work/display spaces, curriculum, instructional technology, institutional policies, and legal compliance. In this case study, we analyze the roles of and committed relationships among stakeholders and institutional actors. However, because of space constraints, not all roles and responsibilities of individual stakeholders and institutional actors are included in our discussion; rather, we represent major decisions, responsibilities, and historical underpinnings. As the program has evolved, stakeholders have shifted—not the categories but the people in the categories and the relative roles and influences of the categories. For example, when the coordinator became a permanent position, the day-to-day roles of the DCI and WCP directors diminished. As the number of clients increased and projects became more complex, the role of the Office of Legal Affairs increased from occasional contact to a regular supporting role.

All stakeholders have a definable commitment to the program, participate in some way, and bring value (which is to say they influence and have articulated responsibilities; Pirozzi, 2019). Although all stakeholders are important in this program in that they "are both recipients and (co)creators of value" (Freudenreich, Lüdeke-Freund, & Schaltegger, 2020), they nonetheless do not have the same level of understanding of, involvement in, or commitment to the program. They sometimes have roles that barely or indirectly interact; their roles or goals may even appear contradictory because each stakeholder has "a different understanding of what constitutes value" (Freudenreich, Lüdeke-Freund, & Schaltegger, 2020)—even though they all want the program to succeed. Here we characterize the individual stakeholders and institutional actors.

Individual Stakeholders

Stakeholders influence and are influenced by the other actors, directly or indirectly. In this case study, stakeholders include faculty, students, clients, and the program coordinator.

Faculty. Faculty assume a number of roles with other stakeholders in the network (e.g., helping students understand workplace expectations). Additionally, faculty coordinate with their teaching partner-about assignments, classroom activities, assessment, and project/course evaluation. Finally, in the early phases of the program, faculty had a relationship with clients through their outreach to and recruitment of new clients; however, since 2019, this role has been assumed by the coordinator.

Given the central role the faculty play as stakeholders, program administrators have focused particular attention on hiring criteria. Since the beginning of the CS-TC program, the courses have been taught by faculty with advanced degrees and workplace experience. The computer science instructors who have taught in the CS-TC program have been PhD lecturers, PhD graduate students, or academic professionals, some with industry or military experience. CS determined that faculty with a PhD, near completion of a PhD, or with significant work experience were qualified to teach in the program. Similarly, the TC instructors who have taught in the CS-TC program have been postdoctoral fellows in WCP, nearly half with PhDs in rhetoric/TC/composition and the others with PhDs in other areas of English studies, with the additional requirement of industry experience. (See the CS-TC website for tables summarizing the program's faculty, including the numbers each year of the program and the advanced degree of each faculty member: CS-TC Instructors by Year, 2022; CS-TC Faculty Education, 2022).

During the process of approving the new courses, experienced tenure-track TC faculty insisted that the CS-TC instructors have a PhD in TC to establish TC expertise within the interdisciplinary collaboration. Over time, requiring a PhD specifically in TC softened, though the instructors appointed to teach these courses continue to be selected according to specific rigorous criteria.

Once faculty were hired, administrators attended to faculty professional development. Prior to teaching the course, all faculty participate in a week-long orientation, including sessions with the CS-TC coordinator. During these sessions, new instructors learn about the curriculum, meet their teaching partner, and begin collaborating with them. All TC instructors participate in the Technical Communication Postdoctoral Seminar during their first semester. All CS-TC instructors take part in further orientation and mentoring through the auspices of the CS-TC coordinator.

At the heart of instructor onboarding, orientation, and professional development—and of transmitting course knowledge from one generation of CS-TC instructors to the next—is the extensive set of resources overseen by the CS-TC coordinator (resources currently on an MS Teams site where stakeholders can also participate in chats and virtual meetings). These resources include details about curriculum, syllabus templates, assignment sheets, assessment criteria and rubrics, class presentations, and other materials that provide a structured framework for new instructors.

Students. The CS-TC students can be accurately described as learners with entrepreneurial spirits. The student teams are expected to address "problems with innovative solutions—solutions that could involve new combinations of products, services, processes, or principles" (Spinuzzi, 2017). Though the core problems have already been defined by the clients, the student teams refine the problems, conceptualize them, and develop software solutions to resolve them. Some students have already gained industry experience through co-ops and internships. Many are just beginning to develop an entrepreneurial mindset and to understand the resourcefulness necessary to learn the ever-changing platforms and coding languages of their discipline.

Because each student team is working with a client, the team works together in a consultancy role. The CS-TC program helps students navigate the change from acting as students (in which problems have right and wrong answers, a clearly defined timeline, and an expectation of coding languages to be used based on what they are learning) to acting as workplace professionals. Moreover, the interdisciplinary collaboration empowers CS students to think of their work beyond its creation and in situ with TC (Johnson-Eilola, 2004). CS students soon realize that they cannot just give clients what they ask for because clients seldom have the same level of technical knowledge. Instead, teams explore new ways of creating and communicating. They find that every choice requires a conversation with teammates and their client. The course sequence provides students opportunities to fail as well as succeed. The students sometimes fail at teamwork, at satisfying course expectations, and, even, at meeting client expectations. However, when this happens in their student role, they do not lose their livelihood, and they learn new ways to navigate CS industry experience before entering the workplace.

Clients. Prior to developing the CS-TC course sequence, the standalone CS capstone course used actual clients—stakeholders with software development needs the student teams could address. The CS-TC program reflected DCI's disciplinary preference for client-based, project-based curriculum. Fortunately, the client base has continued to grow because many previous clients propose additional projects or recommend the program to others. In the network, clients act as stakeholders who are looking for a complete project solution. Clients act as stakeholders not only based on the outcome of the project but also in their contribution to the students' role as learners. As mentors, clients shape the students' understanding of time commitments, communication, and feedback in a professional environment.

The client must agree to provide guidance to the student team(s) throughout the two-semester project. Though students are responsible for maintaining contact, each client is expected to respond to student correspondence in a reasonable time. The client must also agree to provide feedback about the performance of the team(s) to instructors. As initiators of the project, clients are also expected to communicate regularly with students about the scope and the development of the project, as well as discuss resources through a flexible partnership with students (Hea & Shah, 2016). At the end of the project, clients may continue to work with student teams, but no guarantee exists for further support for the software solution.

CS-TC Program Coordinator. From its beginning, the CS-TC program has required regular collaboration and coordination among the actors (Duin, Tham, & Pedersen, 2021). The CS-TC program has had three evolving phases related to the coordinator role.

During the pilot year (during the Early Years), CS and TC administrators worked with the two instructors to plan and organize what was needed. When the CS capstone was a stand-alone course, interaction with project clients was managed by the CS instructor; that practice continued when the CS-TC courses were linked.

Once the pilot year was completed, a TC postdoctoral fellow was asked to coordinate the program day to day. Responsibilities included a range of pedagogical and administrative tasks: orienting faculty to assignments; facilitating faculty inter- action; synthesizing syllabi, assignments, and assessment; updating assignments; reviewing and recommending policy changes; planning the Expo when it was created; and troubleshooting immediate problems. The CS instructor continued to manage interaction with project clients and to organize a registration permit process for students.

At the beginning of the Established Years, the College of Computing, in consultation with WCP, created and funded a position for an academic professional to coordinate/manage the program. The new coordinator assumed responsibility for the tasks described above as well as managing clients and registration permits for students. The coordinator also took on the role of working with the institution's attorneys on the NDA process and other entities on campus to ensurestudents could review IP agreements. The coordinator is employed by

DCI while maintaining close connections with WCP.

Institutional Actors

Although the core of the CS-TC program is comprised of individual stakeholders (e.g., faculty, students, clients, and the coordinator), institutional actors are essential for the program's development and success, including a group of supporting stakeholders as well as the larger institutional offices and institutional vision.[3] Table 1 displays many of these stakeholders and identifies some of their responsibilities. Informing them about programmatic changes, challenges (especially expectations for their time and budget), benefits (e.g., to student learning, to the use of faculty expertise, to programmatic/institutional reputation), and immediate and long-term implications helps these stakeholders. Though these stakeholders are integral to the success of the program, their effectiveness in supporting the program depends upon timely and relevant information.

Table 1. Institutional Actors

Stakeholder Group	Supporting Stakeholders	Selected Responsibilities
Student Classroom Support	• Academic advisors • IT professionals	• Academic advisors guide students about programmatic requirements and advantages of one curricular option over another. • IT professionals advise about, order, install, and maintain classroom and team technologies.

[3] Accreditation is an expected future institutional actor in our interdisciplinary program. The Association for Interdisciplinary Studies (AIS) recognizes the need for the assessment of programs, but also notes that interdisciplinary accreditation is not yet an established practice (see https://interdisciplinarystudies.org/assessing-ids-programs/). Our program follows many of the AIS recommendations, such as program goals, an established curriculum, administrative support, and so on, and continues to work towards targeted assessment. Georgia Tech's accrediting body, the Southern Association of Colleges and Schools Commission on Colleges (https://sacscoc.org/accrediting-standards/), has not addressed the need for interdisciplinary accreditation formally; however, we continue to consult with assessment coordinators and other stakeholders within both the Writing and Communication Program and the College of Computing about ways to best evaluate the CS-TC program.

Table 1. Institutional Actors (cont.)

Depart-ment/ School	• School curriculum committee • Program directors/ associate directors • Course schedulers • School chair	• Department/school curriculum committees approve new courses. • Program directors/associate directors encourage innovation and troubleshoot problems. • Course schedulers can make or break a course and identify pitfalls. • School chairs need to anticipate potential problems, manage political challenges, and know likely benefits.
College	• College curriculum committee • Deans/ associate deans	• College curriculum committees approve new courses. • Deans/associate deans need to anticipate problems, manage political challenges, and know likely benefits.
Institu-tion/Uni-versity	• Institution curriculum-committee • Registrar • Provost	• The institution curriculum committee approves new courses. • The registrar needs to enter new courses as well as manage graduation requirements and scheduling. • The provost needs to know likely benefits.
Work-place	• Client liasons to student teams	• Client liaisons can simplify or deter access, depending on perceptions of the importance/ relevance of regular interaction.

As our program has grown and matured, the network of stakeholders has been dynamic, responsive, and flexible—to meet the needs of the program. Some stakeholders have been consistent, such as the Office of Information Technology, which regularly helps meet program needs. Some stakeholder's roles expanded; for example, we described above the increasing role of the Office of Legal Affairs to meet the legal requirements of students and clients. Some stakeholder roles have been entirely redefined, as with the formalization of the CS-TC coordinator to manage logistical tasks, supervise and maintain curriculum, facilitate onboarding and professional development, and facilitate stakeholder relationships.

Collaboration

One of the critical factors contributing to the success and longevity of the CS-TC program is collaboration—teaching partners, student teams, client relationships, and a range of other interactions that make the program function. In CS-TC courses and in the program as a whole,

> … collaboration involves substantive interactions between and among people who share goals and exchange information as they work toward those goals in a variety of settings and with a variety of tools, either because the task size or complexity is too great for a single person or because the task will benefit from multiple perspectives. (Burnett, Cooper, & Welhausen, 2013)

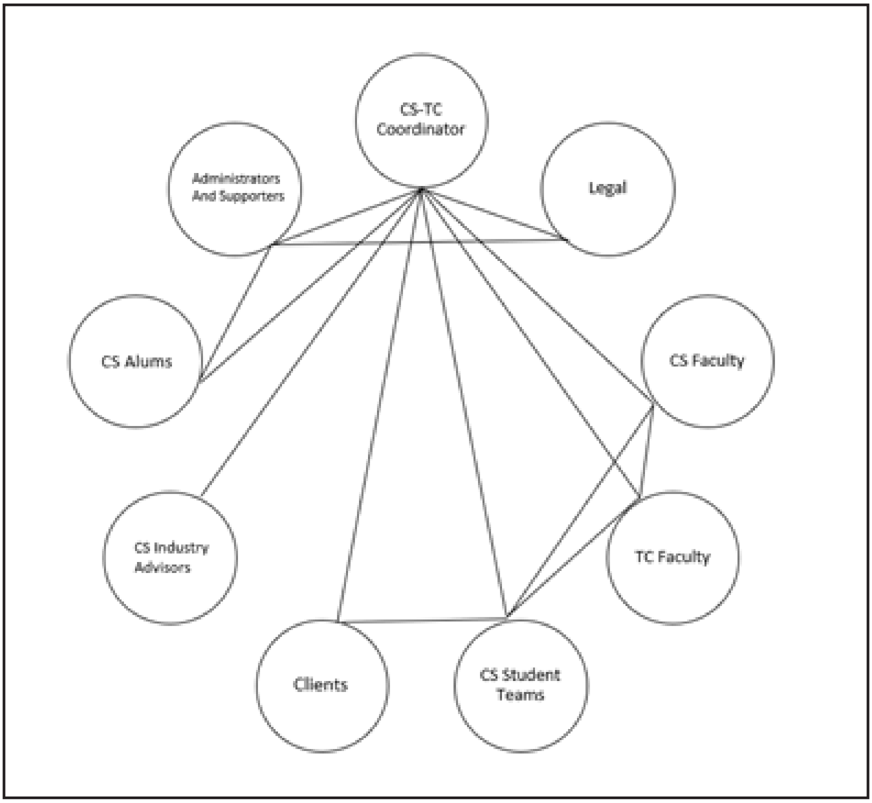
Not only is collaboration good educational practice, but it has long been prevalent in the workplace (Cohen & Bailey, 1997), with the distribution and diversification of teams increasing the frequency of "direct collaboration among individuals who do not share the same kinds of expertise" (Schreiber, Carrion, & Lauer, 2018, p. 2).

Although the practice and value of collaboration among our students and working professionals have always been a priority in the CS-TC program, we note distinct characteristics of the program's collaborative practices among stakeholders. Collaboration grows in complexity as the number of stakeholders increases and a program matures. In the CS-TC program, this complexity is demonstrated, for example, in relationships that emerged among course coordinator, faculty, students, clients, and the Office of Legal Affairs. To navigate these complexities, we have found the following approaches to collaboration to be central to programmatic success:

- Acknowledge and respect interdisciplinary differences and boundaries that shape ways in which collaborators work, teach, think, make knowledge, and make decisions. (We discuss these differences in the following section on interdisciplinarity.)
- Acknowledge power differentials across disciplines and between individual and institutional actors in the network (e.g., relationships between clients and students and relationships between CS and TC faculty pairs).
- Build trust and cultivate interactions that foster psychological safety.

Collaboration in the CS-TC program is decentralized and distributed, so no one actor in the network drives the collaborative efforts. For example, though Figure 1 shows that the CS-TC coordinator is central to the network, the coordinator does not mediate all collaborative relationships in the program.

Figure 1. Distributed collaborative partnerships among some institutional and individual actors in the CS-TC program.



Stakeholder responsiveness is another distinguishing characteristic of the program's collaborative relationships. For instance, in the onboarding process, CS-TC assignments are introduced by the CS-TC coordinator, including learning goals associated with each assignment. As the CS-TC teaching pairs work together to shape their classroom and coordinate their collaborative relationship (Robinson, Dusenberry, & Lawrence, 2016), each pair gleans insights and learns new things to share with the other teaching pairs and the coordinator, taking advantage of "collaboration among individuals who do not share the same kinds of expertise" (Schreiber, Carrion, & Lauer, 2018, p. 2). In the CS-TC program, collaboration is central (Duin, Tham, & Pedersen, 2021), training is essential to "facilitate building a personal awareness of interdependence among team members" (Dusenberry & Robinson, 2020, p.

207), the conception of the problem is constructed and shared (Baker, 2015), and learning to identify and manage types of conflict improves interaction (Burnett, 1993, 1994).

Building and maintaining good will between and among the network's actors helps develop collaborative partnerships. These characteristics of collaboration in the CS-TC program enable the actors to effectively engage with and respond to issues of interdisciplinarity across the program's dynamic and changing contexts (e.g., Hutter et al., 2018; Paretti, 2008; Ritter, 2012).

Program Collaboration

The CS-TC program's full-time coordinator serves as the point of contact for anyone interested in or involved with the program, assuming responsibility for four primary areas of collaboration:

- *College- and university-level conversations* highlight the program and help to establish connections with other capstones. College- and university-level conversations raise awareness about the CS-TC program and make publicity or referrals consistent. The Expo brings attention to the program and promotes recognition from college leadership.
- *Onboarding, orientation, and professional development* are scheduled for everyone new to the program or returning from a hiatus. Additionally, the coordinator typically co-instructs with new DCI faculty.
- *Instructional faculty* in the CS-TC program meet multiple times per semester to connect with each other. The coordinator addresses problems and requests input for curriculum revision. Meetings may also include other stakeholders to explain or demonstrate new procedures or opportunities.
- *New client projects* are arranged by the coordinator, who uses referrals and established relationships built by her predecessors. Clients may meet with the coordinator or just submit a project proposal. However, clients need to understand that their project may not be chosen and that they will be working directly with a student team.

The continuity of the CS-TC program relies not only on current collaboration but also on the connections and collaborations that were able to continue as the program has developed.

Faculty Collaboration

From the beginning, CS and TC faculty negotiated differences in understanding what effective pedagogy entailed. CS faculty were accustomed to lecturing and then expecting the student teams to work independently. In contrast, TC faculty were used to employing active learning that focused on small group problem solving and supported "the process of having students engage in some activity that forces them to reflect upon ideas and how they are using those ideas" (Prince, 2004, p. 160). Active learning was agreed on for all CS-TC classes, and both instructors introduce active learning to students at the beginning of the course. As the courses develop, instructors learn from each other and customize a combination of active learning activities, including discussions, mini-lectures, document analysis, peer review, small-group problem solving, and independent teamwork (with oversight and guidance from the instructors). Whether teaching face-to-face, online, or hybrid sections, instructors see students as participants, not passive recipients, so team activities predominate, and lectures are minimized.

As the program moved from being linked to being fully integrated and the number of instructors increased, instructors acknowledged a broader range of pedagogical approaches. For example, by year 5 of the program, all instructors agreed to use an assignment sheet for each assignment (specifying purpose, audience, design, and so on), and both CS and TC instructors for each section were involved in the grading of all assignments. The courses evolved so that all instructors blended widely accepted pedagogical approaches (e.g., explicitly teaching collaborative strategies) and industry practices (e.g., using iterative Scrums).

Student Collaboration

In the CS-TC program, each class is divided into 10 five-person teams, usually constructed by their instructors; thus, for each course project, an instructor receives 10 artifacts (one from each team) rather than 50 (one from each student). At the beginning of the first semester, each team creates its own team agreement, characterizing the collaborative expectations and responsibilities. This agreement is revisited periodically and can be revised. Though teams are self-governing, instructors typically meet with teams individually and establish policies to reduce problems. Teaching collaborative processes and strategies is part of the curriculum.

Students often express concerns about working in teams. In the study that was conducted in year 5, the two most common categories of team-related concerns were (1) process, including composing in teams; recursive processes in teams; and management of time, efficiency, and schedule in teams and (2) community, including "initiating and engaging in conversation; dealing with individual/team balance, roles, collaboration, and the working environment; managing anxiety or difficulties related to teamwork; and dealing with the stress of interaction" (Burnett, Menagarishvili, & Frazee, 2019, p. 177). Beyond the teamwork, students are engaged in other kinds of collaboration, including active learning, collaborating with faculty, and working with clients.

Interdisciplinarity

Faculty often define interdisciplinarity as integrating disciplines or disciplinary knowledge and, thus, as a "means to increase problem-solving capacity and a working method for reaching a common goal" (Kans & Gustafsson, 2020, p. 5). Faculty exchange ideas and disciplinary knowledge/experience, thereby strengthening their own disciplinary understanding and extending their networks. In fact, interdisciplinary courses have a long history in technical and professional communication, and they can be notably successful if an institution commits systems and resources to such course innovation (e.g., Burnett, Menagarishvili, & Frazee, 2019; Fitzpatrick 2018a, 2018b, 2018c, 2018d, 2018e, 2018f, 2019; Ford & Riley, 2003; Kain & Wardle, 2005; Watts & Burnett, 2012; Williamson & Sweany, 2004). We want to push the definition further by discussing practical negotiations and compromises necessary to create an interdisciplinary course sequence in which the two disciplines see themselves as equal partners.

Efforts to integrate "perspectives/concepts/theories, and/or tools/techniques, and/or information/data from two or more bodies of specialized knowledge or research practice" (Porter et al., 2006, p. 189) increase insight and productivity. Beyond opportunities to develop competence in coding and communication, students in the CS-TC program develop interdisciplinary competence in areas that defy neat categorization: developing proposals, managing projects, interacting with clients, and testing usability. For example, students' "Detailed Design" assignment requires integration of technical and rhetorical knowledge in these ways:

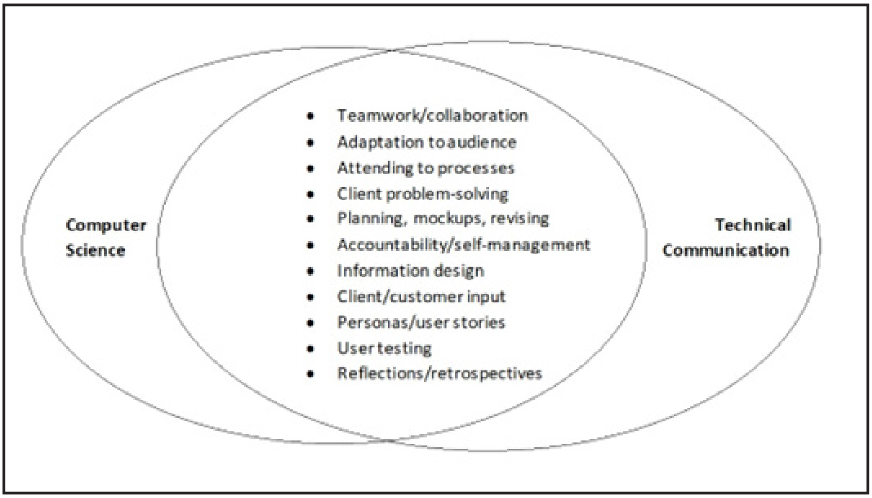
- Communicate an architecture to all interested parties.
- Support the tasks of architecture creation, refinement, and validation.
- Represent hierarchical detail including the creation of substructures by instantiating templates.
- Support the analysis of the architecture.

The kind of interdisciplinarity in the Detailed Design assignment is foundational for the CS-TC program; experiences from both CS and TC help students become more observant, insightful, and effective professionals, more functional in diverse workplace situations, and more responsive in addressing complex problems. Not only do students learn that success as a CS professional depends on more than their ability to code, but faculty learn that they have a co-equal, mutual interdependence with their disciplinary partner in which they jointly address challenges that include negotiating curricular interdisciplinarity and creating an interdisciplinary, public-facing expo (e.g., Burnett, Menagarishvili, & Frazee, 2019; Nardone, Strubberg, & Blackburne, 2020; Fitzpatrick 2018a, 2018b, 2018c, 2018d, 2018e, 2018f, 2019, 2021; Ford & Riley, 2003; Watts & Burnett, 2012).

Curricular Interdisciplinarity

The curriculum integrates theory, research, and practice from both CS and TC, as Figure 2 illustrates. As disciplines collaborate and move toward an integrated approach, their commonalities define some of what is shared. Figure 2 illustrates the kinds of intersections that anyone might make in integrating technical and/or scientific communication with another discipline.

Figure 2. The intersection of CS and TC defines some of what is shared between the disciplines.



One example of interdisciplinarity is using Agile (an approach borrowed from industry) with what are called Scrums to manage complex software and product development. The student teams are responsible for defining/refining the client problem, deciding how to do their work, considering options, and developing a solution. An Agile approach using Scrums is especially suitable to CS-TC because it depends on concepts, processes, and artifacts familiar to both CS and TC faculty:

expectation of regular communication, concern for client/customer input, development of personas and user stories, story mapping, articulated criteria, planning, regular meetings to share information, iterative product and user testing, drafts/mock objects, and reflections/retrospectives (see Figure 2).

The interdisciplinarity of an Agile approach encourages CS and TC faculty to work toward the same broad goals: students who are better written, oral, and visual communicators in their academic and professional lives; who understand communication as a process within intertwining networks; who better understand the social, psychological, political, and ethical aspects of communication; who are better able to communicate their technical ideas; and who are more competent and confident in communicating with classmates, colleagues, and clients. The challenge has been accomplishing these goals. Despite commonalities, we have to work out the mechanics of functional interdisciplinarity: what's an ideal class size, how course credits should be allocated, and how an assignment should be presented, assessed, and evaluated.

Class Size. Everyone on the planning committee agreed that the classes needed to be small in these courses; however, administrators and faculty expressed dramatic disciplinary differences about what constituted "small." For a typical CS class, 50 students per section is considered small; for a TC class, 50 is immense. We compromised: students would work on five-member teams, so although a class section has 50 students in it, those students work collaboratively and submit their work as a team. All assessment focuses on team artifacts, each one reflecting interdisciplinary competencies.

Course Credit. We started with two 1-term, 3-credit courses. We developed an interdisciplinary program that extended over two terms but continued with the same number of credits, equally divided—3 for CS and 3 for TC (see Table 2).

Credit allocation reflects the emphasis of the interdisciplinary deliverables student teams complete in each term. The first term focuses on activities that contribute to the development of a software prototype. The tasks require understanding and practice of TC knowledge such as establishing client relationships, defining a problem, conceptualizing a software solution, and testing a prototype. The second term shifts the emphasis to coding through students' development of the prototype created in the prior term.

Table 2. CS-TC course credits

	ORIGINAL Independent Courses*	REVISED Course Sequence	
Term	**Term 1 OR Term 2**	**Term 1**	**Term 2**
CS	CS 4911 (3 credits)	CS 3311 (1 credit)	CS 3312 (2 credits)
TC	TC* 3403 (3 credits)	TC 3432 (2 credits)	TC 3431 (1 credit)
Credits	Total 6 credits	Total 6 credits	

*Throughout the article, the technical communication courses are labeled as TC to be consistent in the discussion. In the actual institutional catalog, a different signifier identifies the courses.

Assignments. During the Early Years of the CS-TC program, assignments called attention to a disciplinary difference in introducing and explaining assignments to students. CS faculty were accustomed to giving assignments orally and not necessarily identifying assessment criteria, whereas TC faculty regularly used assignment sheets to detail the assignment requirements, explain expectations about the deliverable, provide basic rhetorical information (e.g., purpose, audiences, and expectations about format, organization, and design), and specify assessment criteria. Near the beginning of the program, faculty decided to use assignment sheets for major projects, but their use has developed as the program has matured to include all course assignments.

Currently, all the activities and assignments support the goal of producing a software solution to a client's problem. In working toward this goal, students complete a number of assignments that are unquestionably interdisciplinary, characteristic of both CS and TC: *prototype descriptions*, *design reports*, *recommendation reports*, *final reports*, and various kinds of *presentations*. However, the courses also have other required assignments: a team charter (managing project responsibilities), a *vision statement* (guiding students in conversations with stakeholders), an *MOU* (working as a team-client agreement), *user stories* (characterizing expectations about users), a *demo video* (displaying the operational solution to the problem), and a *retrospective* (a reflective memo capturing the process). All assignments are submitted by the team, with the team receiving the assessment and evaluation.

Assessment and Evaluation. Because of disciplinary assumptions and practices, CS and TC faculty had different notions about assessment of assignments, characterized by what Sally Henschel and Lisa Melonçon (2014) have differentiated as conceptual skills and practical skills. Henschel and Melonçon described research-based conceptual skills: rhetorical proficiency, abstraction, social proficiency, experimentation, and critical system thinking. Each conceptual skill is supported by a cluster of practical skills. For example, the concept of rhetorical proficiency is supported by practical skills such as user analysis, information design, writing, and editing. This attention to both conceptual and practical skills defines what CS-TC faculty eventually decided mattered in assessment and grading, but that was an evolving agreement.

For example, TC instructors initially expected formative assessment to be built into the process of assignments, while CS instructors simply made themselves available to respond to questions if students raised them. With summative assessment, CS faculty initially expected they would assess CS/coding content and TC faculty would assess the mechanical/grammatical conventions of the writing; in fact, CS faculty were surprised by the concern of TC faculty for conceptual skills rather than simply conventions and correctness. TC faculty were led by their disciplinary assumptions to expect that CS and TC would both respond to all aspects of each team's artifacts. Further, the nature of feedback comments differed; CS faculty were especially concerned with conventions and correctness (e.g., if code worked or not), whereas TC faculty were especially concerned with a broader range of response. For example, although TC faculty were also concerned about issues of conventions and correctness in students' writing, they were also concerned about the rhetorical appropriateness paying attention to audience, purpose, context, tone, etc. More particularly, differences included approaches to assessment (e.g., the amount of feedback expected), reliance on an instructor for grading or dependence on a teaching assistant/grader, and awareness of differences resulting from the number of students in a class. Currently, onboarding and the ongoing professional development workshops include attention to both conceptual skills and practical skills as well as formative and summative assessment.

Public-facing Expo

The Computer Science Junior Design Capstone Expo for the CS-TC program is the site for student teams' final presentations at the end of the term. Each student team produces deliverables, including table staging, handouts, an appropriate elevator pitch, screen displays, and a product demonstration. These deliverables reflect materials common to software industry events rather than to academic events (such as posters; Fitzpatrick, 2019). The Expo balances a display of coding competence and communication competence, showcasing the work completed during both terms. Specifically, the Expo is designed to accomplish these objectives:

- Help students develop TC skills specific to software industry events such as tech shows, tech conferences, expos, and recruitment fairs.
- Familiarize students with the processes and deliverables associated with such events.
- Provide a public venue where clients, instructors, administrators, and students can experience and celebrate the work done by CS-TC students.
- Increase the visibility of the course sequence and recruit future project clients (Fitzpatrick, 2019).

The Expo for the CS-TC program is an interdisciplinary, community-centered event that attracts positive attention. The Expo for the CS-TC program is scheduled on a separate day than the capstone design expo for engineering programs at the university, so visiting companies, others from across campus, and alums can attend and view students' work from both CS and engineering. Attendance at the Expo for the CS-TC program fosters intra-college communication, client referral and recruitment, outreach for future student employers, and a venue for other CS-TC students to ask questions and familiarize themselves with their peers' work. Moreover, presenting students gain the skills necessary to explain their project to multiple audiences, discuss their qualifications and skills as experts, answer spontaneous questions, and engage in conversations important to their future career goals.

Sustainability: The Ongoing Conversation

For us, sustainability is "the ability of a system to maintain its health and diversity" (Fleckenstein et al., 2008, p. 411). We believe "our classrooms should offer compelling environments" (Sirc, 2002, p. 1) "for new collisions of ideas, interest, [and] creativity [and, thus, maintain] the energy, interest, and growth of students" (Newcomb, 2012, p. 596). A capstone program such as the one we analyze "brings together theories and practices of the academic field and the workplace" (Melonçon & Schreiber, 2018, p. 322). Designing a situated, sustainable program not only requires attention to functionality but also to an imagined future. The process should always be rhetorical, acknowledging "constraints, competing possibilities, audience factors, and purposes… an innovative response to a perceived situation" (Newcomb, 2012, p. 594). For us, these components include stakeholder commitment, collaboration, and interdisciplinarity as well as five factors we discuss below: flexibility, situatedness, funding constraints, legal issues, and scholarship.

Flexibility

Flexibility undergirds everything in the CS-TC program. The program's culture of flexibility is facilitated by regular faculty meetings. Over the ten years, we have had to explain our disciplinary positions and find workarounds, sometimes resolving that "good enough" was indeed a win-win situation. One of the most visible areas of flexibility in response to change exists in technology, with the program adopting new technology as attitudes and affordances evolve (Clark & Andersen, 2005; Duin & Tham, 2020).

Three examples suffice to demonstrate the importance of flexibility to sustainability. First, in the Early Years, some students focused on CS coding, neglecting their TC effort while other students focused on their TC documents and presentations, assuming they already knew how to code. Administrators and instructors agreed that students needed to pass both courses each term in order to earn credit and to meet their graduation requirement. Second, in the Middle Years, the faculty introduced a usability module that was adopted for all sections, reinforcing a critical competence in both CS and TC. Third, in the Established Years, faculty have addressed issues of IP as the client-based projects have become more challenging.

Situatedness

As James Paul Gee (2004) noted, "we have general expectations about how our language is normally used," but "in actual situations of use, words, and structures take on much more specific meanings"— what Gee called "situated meanings" (p. 21). We argue that situated courses, with a discipline-specific focus, help students better prepare for long-term work in professional environments. They learn common language patterns of that profession as well as "social practices [that] have implications for inherently political things like status, solidarity, distribution of social goods, and power" (p. 21), all of which affect the ways they interpret and create artifacts. Students become members of a community of practice, "people who share a concern or a passion for something they do and learn how to do it better as they interact regularly" (Wenger 2011; Wenger & Snyder, 2000). This interaction is part of situated learning, valuable because, as Jean Lave (2009) explained, "Situated activity always involves changes in knowledge and action, [which] are central to what we mean by 'learning'" (p. 201).

Understanding the situation is important if a program is to be sustained, and one useful way to think about the situation is to borrow the taxonomy developed by William Condon and Carol Rutz (2012) to describe WAC programs. Their generative taxonomy (foundational, established, integrated, and change agent) characterizes various kinds of interdisciplinary programs, drawing attention to programmatic goals, funding, structure, application, and assessment. Each institution needs to determine its own commitments.

Funding Constraints

In our experience, although establishing the courses, managing institutional processes, and hiring/supervising faculty were conducted by program administrators, actually developing, revising, maintaining and sustaining these courses required significant additional time and labor. Most of our programmatic development was accomplished by term-limited postdoctoral fellows (on the TC side) and non-tenure-track lecturer faculty (on the CS side). In all cases, developing this course sequence occurred in addition to the already heavy work of teaching these and other courses.

All the accomplishments discussed here occurred without funding beyond the existing postdoc and lecturer lines. At the same time, though such a program can be constructed "on a shoestring," that doesn't mean that such a program should be. The ingenuity described in our case study didn't occur because of the lack of funding; it happened in spite of it. Course releases, administrative support, or a coordinator hired in 2014 rather than 2019 (all of which were argued for at the time, unsuccessfully) would have made the development process less time-intensive and less stressful.

Legal Issues

Legal issues frequently need to be addressed in sustaining client-based courses that cross disciplinary as well as academic–industry–community boundaries. One of the most common issues involves IP (intellectual property). Although the client's IP includes the idea for the project and any data provided, students' IP includes anything that they code. However, students typically do not understand how IP works, so CS-TC faculty explain IP in class and provide explicit activities and assignments for students to learn about IP as well as other legal concepts, including non-disclosure agreements (NDAs), MOUs, work for hire, and paid research. Even though summaries are provided in class, only a lawyer can advise students about their individual rights. Basic information about IP and MOUs has been built into the CS-TC program. Georgia Tech has an academic unit (Create-X) whose mission supports students' entrepreneurial initiatives, so at least once per semester this unit provides access to legal resources to students who are developing startups. Advice from these external attorneys is available to students in the CS-TC program.

To submit a project, clients must sign an agreement to comply with these CS-TC program policies:

- The projects are designed and implemented by a 5-person team of CS students. Students select and bid on projects. Proposing a project is not a guarantee that a student team will select it.
- Students do not provide a warranty or maintenance for the software applications developed. After customer delivery, no guarantee exists of further support for the software. Requests for further development and enhancements can be conducted between the client and the student team.
- Students—not faculty—are responsible for developing the requirements and for scoping the project; therefore, the client must communicate with the team if a project is selected.
- To propose a project, a client must provide a one- or two-paragraph description of the project including any specialized skills needed on day one of the project. The clients must provide the name and email for the person serving as the primary contact for the students.
- The IP rights to the software are handled between clients and students. Georgia Tech claims no ownership of student work.
- The client must agree to provide guidance to any student team(s) throughout the two-term project. Though students are responsible for establishing and maintaining contact, the client must respond to reasonable student correspondence and feedback requests in a reasonable time. The client must also agree to provide feedback about the performance of the student team(s) to course instructors each term.

Clients who comply with these policies tend to become long-term stakeholders in the CS-TC program.

Scholarship

Even though the primary mission of the CS-TC program is teaching, faculty research and scholarship are also critical. As Ernest L. Boyer (1990) argued more than 30 years ago, teaching is an appropriate subject for research, a position that has given strong support for the scholarship of teaching and learning (SOTL).

SOTL systematically investigates questions related to student learning for teachers to improve their own teaching and also to advance the teaching of others (Kern, et al., 2015), a practice supported by our program. A number of CS-TC administrators and faculty have presented and published scholarship with “the hope of making a difference” (Fleckenstein et al., 2008, p. 406). To date, faculty in the program have generated dozens of local, regional, national, and international SOTL presentations and numerous SOTL publications on academic blogs as well as in refereed proceedings, edited collections, and peer-reviewed journals that refer to the program as an example or provide detailed discussion of some aspect of the program. (See a list of the nearly 60 program-related presentations and publications; CS-TC Scholarship, 2022). Our case study provides an in-depth description of a single program that is part of a complex network in one institution. Because programs are "quite distinct from one place to the next" (Steinberg, 2015, p. 154), some believe programmatic case studies have a "perceived inability to generate theoretical insights beyond the case in question" (p. 152). We pose an alternative perspective: generalization does not necessarily need to suggest broad applicability, predictability, or transferability; instead, generalization can "focus our attention on the practical challenge of moving from the facts at hand to broader claims" (p. 153). We believe re-focusing attention is one of the enormous values of case studies. Though local networks are distinct, the categories of actors exist from one network to the next, so we can strengthen our understanding of one network by learning about ways in which another network functions.

Although our case study has been organized around four key success factors (stakeholders, collaboration, interdisciplinarity, and sustainability), we recognize that building and maintaining common ground and encouraging program responsiveness emerge as central to all four factors. Thus, we conclude with a series of questions that programs or program administrators might respond to as they begin conversations with another program or seek to expand or enrich existing TC programs. The usefulness or appropriateness of a question depends not only on the demographics of the student population and the resources of the institution, but especially on the phase of development in a program.

Building and Maintaining Common Ground

Creating an integrated, interdisciplinary program requires finding common ground—that is, areas of mutual concern or interest. The purposeful and intentional process involves intense intellectual and emotional labor, with attention to disciplinary criteria, pedagogical philosophy and praxis, workplace expectations, and teaching pairs (Hutter & Lawrence, 2016). The following questions might help your program work towards building common ground with stakeholders in your network:

- Who are your stakeholders, what motivates them, and what is needed to identify common ground with them?
- Once stakeholders join your project, what is the common ground between you and them? How do you and your stakeholders develop and maintain common ground? How might the departments/

units involved understand interdisciplinary collaboration?
- What do you know/assume (and not know/assume) about the other discipline's pedagogical, administrative, epistemological, and cultural differences, and are you willing to have those assumptions challenged towards building common ground?
- How can professional development support your interdisciplinary faculty and programmatic collaboration?

Encouraging Responsiveness

Creating an integrated, interdisciplinary program also requires programmatic responsiveness—that is, attention to attitudes and actions that are needed to make things work. The following questions might be useful as you consider your program's readiness and responsiveness with a view towards sustainability:

- What interdisciplinary framework(s) will the program use for creating, building, and sustaining curricular practices?
- How well are stakeholders and programmatic units prepared to respond to challenges (e.g., identifying common ground, planning to meet programmatic needs, leveraging resources, managing conflict, navigating complexity, and maintaining responsiveness)?
- How can you identify and access the resources needed to create, build, and sustain a program?

We encourage readers to use the case study not as a roadmap so much as a felt sense that integration and interdisciplinarity are possible. The case encourages beginning with thinking, planning, and piloting rather than jumping into action. We hope readers consider ways to use the concepts in their own situations, redefining common practices (e.g., as we redefined productive class size) and developing workarounds (e.g., inviting postdoc coordinators until funding was available for a permanent coordinator). Even when the challenges are abundant, the case provides evidence that successes exist, for example, moving to new ways of thinking about collaboration with colleagues, to shared workspaces, to different approaches to concepts.[4]

[4] Organizational change signals both political and pedagogical evolution. In August 2022, Georgia Tech's College of Computing announced the launch of its School of Computing Instruction (SCI)—formerly the Division of Computing Instruction (DCI)—responsible for teaching all 1000- and 2000-level courses in CS, as well as some upper-division CS courses, such as those discussed in this article (CS 3311 and CS 3312). In announcing this change, Charles Isbell, Dean, and John P. Imlay, Jr., Chair of the College of Computing, explained that, in addition to teaching, SCI faculty "produce new scholarship and techniques that expand everyone's ability to both teach and learn computing" (Claycombe, 2022). This organizational change will influence faculty responsibilities and the role of the CS-TC coordinator as well as the relationship with the Writing and Communication Program. Colleges and universities initiating interdisciplinary programs should anticipate similar organizational changes that necessarily influence the development and direction of their programs.

Thus, we hope the contextual narrative of our case study becomes a stimulus for conversation, focusing attention on concerns relevant to any institution considering collaboration and interdisciplinarity as foundational for an integrated program.

References

Aka, Kadia Georges; & Labelle, François. (2021). The collaborative process of sustainable innovations under the lens of actor–network theory. *Sustainability*, 13(19), 10756. https://doi.org/10.3390/su131910756

Alpi, Kristine M.; & Evans, John Jamal. (2019). Distinguishing case study as a research method from case reports as a publication type. *Journal of the Medical Library Association: JMLA*, 107(1), 1–5. https://doi.org/10.5195/jmla.2019.615

Baker, Michael J. (2015). *Collaboration in collaborative learning. Interaction Studies*, 16(3), 451–473. https://doi.org/10.1075/is.16.3.05bak

Bourelle, Tiffany. (2015). Writing in the professions: An internship for interdisciplinary students. *Business and Professional Communication Quarterly*, 78(4), 407–427. https://doi.org/10.1177/2329490615589172

Boyer, Ernest L. (1990). *Scholarship reconsidered: Priorities of the professoriate*. Princeton UP, The Carnegie Foundation for the Advancement of Teaching.

Bridgeford, Tracy; & St.Amant, Kirk. (Eds.). (2017). Academy-industry relationships and partnerships: *Perspectives for technical communicators*. Routledge.

Burnett, Rebecca E. (1993). Decision-making during the collaborative planning of coauthors. In Ann Penrose & Barbara Sitko (Eds.), *Hearing ourselves think: Cognitive research in the college writing classroom* (pp. 125–146). Oxford UP.

Burnett, Rebecca E. (1994). Productive and unproductive conflict in collaboration. In Linda Flower, David L. Wallace, Linda Norris, & Rebecca E. Burnett (Eds.), *Making thinking visible: Writing, collaborative planning, and classroom inquiry* (pp. 239–244). NCTE.

Burnett, Rebecca E.; Cooper, L. Andrew; & Welhausen, Candice. (2013). What do technical communicators need to know about collaboration. In Johndan Johnson-Eilola & Stuart A. Selber (Eds.), *Solving problems in technical communication* (pp. 454–478). University of Chicago Press.

Burnett, Rebecca E.; Girard, Amanda; Omojokun, Olufisayo; & Stallworth, Cedric. (2019, March 13). *Coordination, collaboration, and sustainable client practices* [Conference session]. Association of Teachers of Technical Communication Annual Conference, Pittsburgh, PA, United States.

Burnett, Rebecca E.; Menagarishvili, Olga; & Frazee, Andy. (2019). Student attitudes about teamwork in face-to-face and blended technical communication classes. In Amanda G. Madden, Lauren E. Margulieux, Robert S. Kadel, & Ashok K. Goel (Eds.), *Blended learning in practice: A guide for practitioners and researchers* (pp. 155–191). MIT Press.

Claycombe, Ann. (2022, August 16). New school highlights importance of computing instruction at Georgia Tech. https://www.cc.gatech.edu/news/new-school-highlights-importance-computing-instruction-georgia-tech

Clark, Dave; & Andersen, Rebekka. (2005). Renegotiating with technology: Training towards more sustainable technical communication. *Technical Communication*, 52(3), 289–301.

Cohen, Susan G.; & Bailey, Diane E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of Management*, 23(3), 239–290. https://doi.org/10.1177/014920639702300303

Condon, William; & Rutz, Carol. (2012). A taxonomy of writing across the curriculum programs: Evolving to serve broader agendas. *College Composition and Communication*, 64(2), 357–382. https://www.jstor.org/stable/43490756

Cross, Geoffrey; & Wills, Katherine. (2001). Using Bloom to bridge the WAC/WID divide. https://files.eric.ed.gov/fulltext/ED464337.pdf

CS-TC Faculty Education. (2022). Programmatic History: Georgia Tech Computer Science Junior Design Capstone. https://sites.gatech.edu/csjuniordesigncapstone/programmatic-history/

CS-TC Faculty by Year. (2022). Programmatic History: Georgia Tech Computer Science Junior Design Capstone. https://sites.gatech.edu/csjuniordesigncapstone/programmatic-history/

CS-TC Planning Committee and Program Coordinators. (2022). Programmatic History: Georgia Tech Computer Science Junior Design Capstone. https://sites.gatech.edu/csjuniordesigncapstone/programmatic-history/

CS-TC Program Timeline. (2022). Programmatic History: Georgia Tech Computer Science Junior Design Capstone. https://sites.gatech.edu/csjuniordesigncapstone/programmatic-history/

CS-TC Scholarship. (2022). Programmatic History: Georgia Tech Computer Science Junior Design Capstone. https://sites.gatech.edu/csjuniordesigncapstone/programmatic-history/

de Vries, Kay. (2020). Case study methodology. *Critical Qualitative Health Research: Exploring Philosophies, Politics and Practices*, 41–52.

Duin, Ann H.; & Tham, Jason. (2020). The current state of analytics: Implications for learning management system (LMS) use in writing pedagogy. *Computers and Composition*, 55, 102544. https://doi.org/10.1016/j.compcom.2020.102544

Duin, Ann H.; Tham, Jason; & Pedersen, Isabel. (2021). The rhetoric, science, and technology of 21st century collaboration. In Michael J. Klein (Ed.), *Effective teaching of technical communication: Theory, practice and application* (pp. 169–192). University Press of Colorado. https://doi.org/10.37514/TPC-B.2020.1121.2.09

Dusenberry, Lisa; & Robinson, Joy. (2020). Building psychological safety through training interventions: Manage the team, not just the project. *IEEE Transactions on Professional Communication*, 63(3), 207–226. https://doi.org/10.1109/TPC.2020.3014483

Fitzpatrick, KellyAnn. (2018a, January 29). On conferences in academia and the tech industry. *TECHStyle*. https://techstyle.lmc.gatech.edu/on-conferences-in-academia-and-the-tech-industry/

Fitzpatrick, KellyAnn. (2018b, March 8). Public tech comm: Preparing students for tech expos. *TECHStyle*. https://techstyle.lmc.gatech.edu/public-tech-expos/

Fitzpatrick, KellyAnn. (2018c, April 4). From first year comp to tech comm (and beyond): An interview with Dori Coblentz. *TECHStyle*. https://techstyle.lmc.gatech.edu/interview-with-dori-coblentz/

Fitzpatrick, KellyAnn. (2018d, April 23). On Teaching computer science: An Interview with Bob Waters. *TECHStyle*. https://techstyle.lmc.gatech.edu/interview-with-bob-waters/

Fitzpatrick, KellyAnn. (2018e, November 5). Intersections of tech comm in the tech industry and classroom. *TECHStyle*. https://techstyle.lmc.gatech.edu/intersections-of-tech-comm-in-the-tech-industry-and-classroom/

Fitzpatrick, KellyAnn. (2018f, November 13). Georgia Tech's CS tech comm & junior design sequence. *TECHStyle*. https://techstyle.lmc.gatech.edu/cs-tech-comm-junior-design-sequence/

Fitzpatrick, KellyAnn. (2019, May 6). The spring 2019 CS Junior Design Capstone Expo: A recap and interview with Amanda Girard and Alyshia Jackson. *TECHStyle*. https://techstyle.lmc.gatech.edu/spring-2019-cs-junior-design-capstone-expo/

Fitzpatrick, KellyAnn. (2021). A tech comm classroom update: Georgia Tech's CS Junior Design courses (and Expo) go virtual. *Sometimes Dragons*. https://redmonk.com/kfitzpatrick/2021/06/22/classroom-update-cs-courses-and-Expo-go-virtual/

Fleckenstein, Kristie S.; Spinuzzi, Clay; Rickly, Rebecca J.; & Papper, Carole Clark. (2008). The importance of harmony: An ecological metaphor for writing research. *College Composition and Communication* 60(2), 388–419. https://www.jstor.org/stable/20457064

Flyvbjerg, Bent. (2006). Five misunderstandings about case-study research. *Qualitative Inquiry*, 12(2), 219–245. https://doi.org/10.1177/1077800405284363

Flyvbjerg, Bent. (2011). Case study. *The Sage handbook of qualitative research*, 4, 301-316.

Ford, Julie Dyke; & Riley, Linda Ann. (2003). Integrating communication and engineering education: A look at curricula, courses, and support systems. *Journal of Engineering Education*, 92(4), 325–328. https://doi.org/10.1002/j.2168-9830.2003.tb00776.x

Freudenreich, Birte; Lüdeke-Freund, Florian; & Schaltegger, Stefan. (2020). A stakeholder theory perspective on business models: Value creation for sustainability. *Journal of Business Ethics*, 166(1), 3–18.

Gee, James Paul. (2004). Discourse analysis: What makes it critical? In Rebecca Rogers (Ed.), *An introduction to critical discourse analysis in education* (pp. 19–50). Lawrence Erlbaum.

Hea, Amy C. Kimme; & Shah, Rachel Wendler. (2016). Silent partners: Developing a critical understanding of community partners in technical communication service-learning pedagogies. *Technical Communication Quarterly*, 25(1), 48–66. https://doi.org/10.1080/10572252.2016.1113727

Henschel, Sally; & Melonçon, Lisa. (2014). Of horsemen and layered literacies: Assessment instruments for aligning technical and professional communication undergraduate curricula with professional expectations. *Programmatic Perspectives*, 6(1), 3–26.

Hora, Matthew T.; Smolarek, Bailey B.; Martin, Kelly Norris; & Scrivener, Luke. (2019). Exploring the situated and cultural aspects of communication in the professions: Implications for teaching, student employability, and equity in higher education. *American Educational Research Journal*, 56(6), 2221–2261. https://doi.org/10.3102/0002831219840333.

Hutter, Liz; & Lawrence, Halcyon M. (2016, October). From finding to building common ground: Negotiating issues in the co-design of a cross-disciplinary computer science and technical communication course. In 2016 *Proceedings for the Conference of the Council for Programs in Technical and Scientific Communication* Savannah, Georgia (pp. 47–49).

Hutter, Liz; Lawrence, Halcyon M.; McDaniel, Melinda; & Murrell, Marguerite. (2018). Fostering meaningful collaboration in an interdisciplinary capstone course. In *SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 263–264). https://doi.org/10.1145/3159450.3159625.

Johnson-Eilola, Johndan. (2004). Relocating the value of work: Technical communication in a post-industrial age. In Johndan Johnson-Eilola & Stuart A. Selber (Eds.), *Central works in technical communication* (pp. 175–192). Oxford UP.

Jones, Natasha N.; Moore, Kristen R.; & Walton, Rebecca. (2016). Disrupting the past to disrupt the future: An antenarrative of technical communication. *Technical Communication Quarterly*, 25(4), 211–229. https://doi.org/10.1080/10572252.2016.1224655

Kain, Donna; & Wardle, Elizabeth. (2005). Building context: Using activity theory to teach about genre in multi-major professional communication courses. *Technical Communication Quarterly*, 14(2), 113–139. https://doi.org/10.1207/s15427625tcq1402_1

Kans, Mirka; & Gustafsson, Åsa. (2020). Internal stakeholders' views on interdisciplinarity: An empirical study within an interdisciplinary master's program. *Cogent Education*, 7(1), 1731221. https://doi.org/10.1080/2331186X.2020.1731221

Kern, Beth; Mettetal, Gwendolyn; Dixson, Marcia D.; & Morgan, Robin K. (2015). The role of SoTL in the academy: Upon the 25th anniversary of Boyer's scholarship reconsidered. *Journal of the Scholarship for Teaching and Learning*, 15(3)1–14. https://doi.org/10.14434/josotl.v15i3.13623

Kmiec, Dave; Menagarishvili, Olga; & Longo, Bernadette. (2017a, July 24). *Humanistic approaches for teaching technical writing service courses better meet the needs of our students (and our discipline)* [Conference session]. IEEE ProComm2017, Madison, WI, United States.

Kmiec, Dave; Menagarishvili, Olga; & Longo, Bernadette. (2017b). Humanistic approaches for teaching technical writing service courses better meet the needs of our students (and our discipline). *IEEE ProComm2017 Proceedings*, 1–12. https://doi.org/10.1109/IPCC.2017.8013951.

Lancaster, Amber L.; & Yeats, David. (2016). Establishing academic-industry partnerships: A transdisciplinary research model for distributed usability testing. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, 8(3), 29–45. https://doi.org/10.4018/IJSKD.2016070103

Latour, Bruno. (1996). On actor-network theory: A few clarifications. *Soziale Welt* 47(4), 369–81. https://www.jstor.org/stable/40878163

Latour, Bruno. (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford UP. http://hdl.handle.net/2027/heb.32135

Lave, Jean. (2009). The practice of learning. In Knud Illeris (Ed.), *Contemporary theories of learning: Learning theorists in their own words* (pp. 200–208). Routledge. (Reprinted from *Understanding practice: Perspectives on activity and context*, pp. 3–34, by Seth Chaiklin & Jean Lave (Eds.), 1996, Cambridge UP.

Lindgreen, Adam; Di Benedetto, C. Anthony; & Beverland, Michael B. (2021). How to write up case-study methodology sections. *Industrial Marketing Management*, 96, A7–A10. https://doi.org/10.1016/j.indmarman.2020.04.012.

Melonçon, Lisa; & Schreiber, Joanna. (2018). Advocating for sustainability: A report on and critique of the undergraduate capstone course. *Technical Communication Quarterly*, 27(4), 322–335. https://doi.org/10.1080/10572252.2018.1515407

Menagarishvili, Olga. (2018, January 5). *Technical communication process in a face-to-face and a blended learning class: Managing time, drafting, collaborating, and providing feedback* [Conference session]. MLA 2018, New York, NY, United States.

Menagarishvili, Olga; Frazee, Andy; & Burnett, Rebecca E. (2022a, July 18). *Peer feedback in linked courses: Perceptions of benefits and problems* [Conference session]. IEEE ProComm2022, Limerick, Ireland.

Menagarishvili, Olga; Frazee, Andy; & Burnett, Rebecca E. (2022b). Peer feedback in linked courses: Perceptions of benefits and problems. *IEEE ProComm2022 Proceedings*, 318-324. https://doi.org/10.1109/ProComm53155.2022.00065

Nardone, Carroll Ferguson; Strubberg, Brandon; & Blackburne, Brian. (2020). Complementing the classroom: Building productive spaces for technical and professional communication. *Programmatic Perspectives*, 11(2), 113–137.

Newcomb, Matthew. (2012). Sustainability as a design principle for composition: Situational creativity as a habit of mind. *College Composition and Communication* 63(4). 593–615. https://www.jstor.org/stable/23264230

Paretti, Marie C. (2008). Teaching communication in capstone design: The role of the instructor in situated learning. *Journal of Engineering Education*, 97(4), 491–503. https://doi.org/10.1002/j.2168-9830.2008.tb00995.x

Pirozzi, Massimo. (2019). Stakeholders, who are they. *PM World Journal*, 8(9), 1–10. https://pmworldlibrary.net/wp-content/uploads/2019/10/pmwj86-Oct2019-Pirozzi-stakeholders-who-are-they.pdf

Porter, Alan L.; Roessner, J. David; Cohen, Alex S.; & Perreault, Marty. (2006). Interdisciplinary research: Meaning, metrics and nurture. *Research Evaluation*, 15(3), 187–195. https://doi.org/10.3152/147154406781775841

Prince, Michael. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223–231. https://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Ritter, Christopher. (2012). A letter to students and clients—Drawing lessons from failure in a service learning classroom. *TECHStyle*. https://techstyle.lmc.gatech.edu/a-letter-to-students-and-clients-drawing-lessons-from-failure-in-a-service-learning-classroom/

Robinson, Joy; Dusenberry, Lisa; & Lawrence, Halcyon M. (2016). Collaborative strategies for distributed teams: Innovation through interlaced collaborative writing. *2016 IEEE International Professional Communication Conference*, pp. 1–9. https://doi.org/10.1109/IPCC.2016.7740489

Schreiber, Joanna; Melissa Carrion; & Lauer, Jessica. (2018). Guest Editors' introduction: Revisiting the service course to map out the future of the field. *Programmatic Perspectives*, 10(1), 1–11.

Sirc, Geoffrey. (2002). *English Composition as a Happening*. Utah State UP.

Spinuzzi, Clay. (Ed.). (2017). Introduction to the special issue on the rhetoric of entrepreneurship: Theories, methodologies, and practices [Special issue]. *Journal of Business and Technical Communication*, 31(3), 275–289. https://journals.sagepub.com/doi/full/10.1177/1050651917695537.

Steinberg, Paul F. (2015). Can we generalize from case studies? *Global Environmental Politics* 15(3), 152–175. https://www.muse.jhu.edu/article/587551

Thomas, Gary. (2011). A typology for the case study in social science following a review of definition, discourse, and structure. *Qualitative Inquiry*, 17(6), 511–521.

Watts, Julie; & Burnett, Rebecca E. (2012). Pairing courses across the disciplines: Effects on writing performance. *Written Communication*, 29(2), 208–235. https://doi.org/10.1177/0741088312438525

Wenger, Etienne. (2011). *Communities of practice: A brief introduction*. https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/11736/A%20brief%20introduction%20to%20CoP.pdf

Wenger, Etienne C.; & William M. Snyder. (2000). Communities of practice: The organizational frontier. *Harvard Business Review*, 78(1), 139–146.

Williamson, W. J.; & Sweany, Philip. (2004). Discipline-specific instruction in technical communication. In T. Bridgeford, K. S. Kitalong, & D. Selfe (Eds.), *Innovative Approaches to Teaching Technical Communication* (pp. 60-80). University Press of Colorado. https://wac.colostate.edu/docs/books/usu/ttc/chapter4.pdf

Author Information

Rebecca E. Burnett (PhD Carnegie Mellon) recently retired as Director of Writing and Communication at Georgia Tech, where she held an Endowed Professorship. Her research includes arts and humanities, collaboration, leadership, multimodality, risk, technical communication, and visual rhetoric. She is particularly interested in the engagement, comprehension, and response of public audiences, whether to the arts or to workplace discourse.

Andy Frazee (PhD University of Georgia) serves as the Director of Georgia Tech's Writing and Communication Program (WCP), supporting teaching, research, service, and professional development for 38 Marion L. Brittain Postdoctoral Fellows and WCP lecturers. In addition to research examining writing program administration, pedagogy, and faculty development, he is also a publishing poet (The Body, The Rooms, Subito Press).

Amanda K. Girard (PhD Michigan Technological University) works for the Division of Computing Instruction in the College of Computing at Georgia Institute of Technology. A former Brittain Postdoctoral Fellow at Georgia Tech, she is currently the Computer Science Junior Design Capstone Course Coordinator. Her current research interests include digital humanities, writing program administration, technical communication, multimodality, and rhetoric.

Liz Hutter (PhD University of Minnesota) is an Assistant Professor at University of Dayton. Her research centers around rhetorical and cultural studies of health and medicine, and her pedagogical interests span technical communication and the medical and health humanities. She is a former Brittain Postdoctoral Fellow at Georgia Tech and served as a co-coordinator of the computer science-technical communication program.

Halcyon Lawrence (PhD Illinois Institute of Technology) is an Associate Professor at Towson University. Her research focuses on speech intelligibility and the design of speech interactions for voice technologies, particularly for under-represented user populations. She is a former Brittain Postdoctoral Fellow at Georgia Tech and served as a co-coordinator of the computer science-technical communication program.

Olga Menagarishvili (PhD University of Minnesota) is an Assistant Professor at Metropolitan State University. Her research includes scientific and technical communication, online and blended learning pedagogy, multimodality, rhetoric, and lexicography. She is a former Brittain Postdoctoral Fellow at Georgia Tech and served as the first coordinator of the computer science-technical communication program.