A Framework in which to teach (technical) communication to computer science majors

Henry Etlinger

Follow this and additional works at: http://scholarworks.rit.edu/other

Recommended Citation

A Framework In Which To Teach (Technical) Communication to Computer Science Majors

Henry A. Etlinger
Rochester Institute of Technology
Department of Computer Science
Rochester, NY 14623 USA
(585) 475-2097
hae@cs.rit.edu

ABSTRACT
We describe a framework in which to consider designing an entire course for Computer Science majors that covers both technical communications as well as more general communication issues. By utilizing abstraction, we’re able to fashion a framework that allows instructors to design specific course elements that meet their needs or priorities. We take advantage of many other efforts reported upon in the past that discuss processes that students can follow in developing effective communication artifacts, blend writing or speaking assignments into individual courses, or talk about communication-intensive courses.

Categories and Subject Descriptors
K.3 [Computers and Education]: Computer and Information Science Education – Computer Science Education, Curriculum

General Terms
Documentation

Keywords
Communication skills, writing, presentations, audience analysis, framework

1. INTRODUCTION
Computer Science is a technically-oriented discipline as well as a relatively young discipline. Our evolution reflects multiple generations of thought regarding which programming language to teach, which programming paradigm to embrace, which technology to employ, and which topics and themes form the foundation for the field. To our credit, we have also recognized the value of support areas, such as communication skills and ethics, and have sought ways to strengthen such areas in the fabric of our curriculums. We can trace, for example, a growing recognition of the importance of communication skills in a sequence of ACM curriculum guidelines [1,2,4,5]. We also know that employers expect our graduates to not only be familiar with technology that supports written and verbal communication, but to understand how to use such technology effectively and to know what form of communication best serves the needs of a given audience. Our graduates should be familiar with professional level standards for common artifacts, such as requirements documents, test plans, and user guides. In addition, numerous professionals outside the field try to make students aware of how important communication skills are in general. For example, the Career Services and Placement office at Michigan State University publishes 12 Essentials for Success [10] in an effort to help students understand what awaits them in the workplace. Communicating effectively means more than just learning to read and write. It also means learning to speak with authority and delivering the right message to the right audience. And, it means learning to listen carefully so that what you hear and how you interpret what you hear is correct.

Although we seem to recognize the importance of communication skills and we know that employers value these traits, many employers still cite poor communication skills of college graduates. As one example, a recent employment summary posted for new college graduates at a website sponsored by the National Association of Colleges and Employers [7]. While the job outlook for students with technical skills appears to be improving, employer feedback regarding the skills and qualities they’d like to see in new hires is troubling. Communication skills top the list of most desired skills (and has for a number of years), yet employers continue to lament lack of writing as well as more general grammar skills in many college graduates. Paraphrasing one of the respondents, “can graduates communicate good ideas without saying ‘like’ and ‘you know’ seemingly over and over?”

2. A SAMPLE OF APPROACHES
Over the years, authors have reported a variety of approaches to teaching communication skills in technical disciplines such as Computer Science or Engineering. Some instructors report on how to integrate one type of activity (such as oral presentations) into the classroom [9]. Some authors discuss how to add writing assignments to existing courses [6]. We have seen and continue to see proposals that focus on process issues [8] as well as proposals that discuss innovative course designs [3]. These efforts provide useful data, offer advice on how to motivate students, how to structure activities, how to avoid pitfalls, and how instructors can manage the workload of teaching courses with heavy emphasis on writing, speaking, and evaluation. It’s clear that no single approach works best, nor does a single approach always accommodate the needs of particular groups of students or fit the constraints imposed by specific computer
science curriculum. And, even with all of this experimental data available, there still seems to be something missing. Perhaps it’s about how to balance the workload, both for students and for instructors, or perhaps it’s finding ways to truly highlight for students what it is important in any given communications activity. Perhaps, too, we should be reaching for more than just an introduction to the technical artifacts that software developers are likely to see. As employers continue to point out, our students need to raise their level of awareness with regards to all forms of communication.

3. A COMMUNICATIONS COURSE

3.1 How Our Course Started

At Rochester Institute of Technology (RIT), our efforts to improve students’ communications skills parallel work done elsewhere. Starting from a situation in which we simply had students occasionally write a research paper in selected upper-level courses, we first introduced a 2-credit Technical Writing course (RIT is on a ten-week quarter system and most courses typically carry 4 credits) as part of graduation requirements. While faculty had some say in content, most sections of the course were taught for us by other members of the academic community. When we significantly overhauled our curriculum in the early 1990’s, we replaced Technical Writing with a 4-credit course titled Professional Communications that we would teach ourselves.

We targeted second-year students for this course and built the course based on a number of guiding principles.

- Students need practice in both writing and in giving presentations and need to have multiple opportunities to do both
- Students benefit from practicing a process that includes not only developing drafts of their own work, but in reviewing and critiquing the works of others
- Students benefit from seeing professional level work products other than code
- Students can see tangible, immediate benefits from drafting business letters and résumés targeted to specific job opportunities (this last point is relevant since our program requires mandatory co-op blocks, starting once a student reaches third year status)

In the earliest versions of the course, we focused on developing notes that could be used for some lectures (such as how to organize material logically, tips for preparing and giving presentations, common grammar problems and possible solutions), activities that could be done in class, either individually or in small groups, and, of course, assignments. We wanted assignments to be representative of the kinds of work products our students could expect to see on co-op or in the work environment. In addition to asking students to prepare sample application letters and résumés, we also asked them to write a user guide explaining how to carry out some task that a beginning student might encounter (such as setting up a different windowing environment in our labs or showing students how to remotely connect their personal computer to one of our lab machines). We offered instructors some choices, so that, for example, some instructors might also ask students to develop a sample test plan or a requirements document. For presentations, either done individually or as part of a small group, we typically asked students to prepare and give a talk on some software product that might be used by students in one of our labs.

We tried to encourage students to follow an iterative and incremental process (not unlike the one we advocated when they developed code in some of our programming-oriented courses). We wanted to encourage them to help one another and so we required at least one draft for each written document. Drafts were due on a specific date, brought to class and given to someone else in the class for review, with the hopes that reviewers would provide helpful feedback. We also grappled with numerous problems that others have often reported. What did we do in class when a student didn’t bring a draft? How could one instructor manage to keep up with grading submissions, all of which were essentially unique? How much time do we devote to basic writing skills and how do we address grading issues with respect to grammar and usage? How could we motivate this group of students to take part in activities that led to work products that would be subjectively judged as opposed to developing computer programs that were often objectively evaluated? Do we require a textbook, do we give students exams, and do we deal with late assignments?

3.2 Taking Things Further

Over time, various instructors have modified the initial course design by adding different assignments or conducting different kinds of in-class activities. The general tone of the course and focus of the assignments, however, has remained the same. Some instructors require a textbook and others make one optional. Most instructors do not give exams, finding it difficult to determine just what to test. We’ve built checklists so that students know what we look for when we grade.

Some students come to realize, if not during the course, then by the time they graduate, the value of improving their technical writing and speaking skills. Some students, unfortunately, are single-minded in equating ‘interesting’ work in computing only with programming. If programming is not a significant feature of a course, then they tend to tune out the course and the opportunities for learning that it presents. This leads to frustration, not only for students, but for instructors as well.

Even though we’ve had some success in improving our students’ abilities to write technical documents or deliver and design technical presentations, and that is evidenced by their work in subsequent courses and on co-op assignments, we’ve had less success in convincing our students to strive to make each and every communication opportunity serve its purpose. Simple tasks such as producing a clear and concise email, framing a request, or making a proposal, are either tasks that our students do not think are important or ones that they simply don’t wish to spend much time on. Since we have dedicated a single course to technical communications, it seems appropriate to stretch the course and find ways to treat the broader communications landscape at the same time that we look at more traditional topics for such a course. One possible answer lies in creating a framework, and then attaching each assignment, each topic, and each skill, to this framework. Our students become accustomed to thinking in terms of abstractions and mapping one level of abstraction to another. The proposed
framework represents a high level abstraction and once we can make students understand it, then it becomes somewhat easier to help them see specifically what is needed for any assigned task, from the most basic to the most complex.

4. A FRAMEWORK

The framework is presented as a series of concentric circles. The goal is to present the entire framework early in the term, discussing each point briefly and offering some illustrations or examples to help clarify each point. Then, throughout the rest of the term, as each assignment is given or each task emerges, we go back to the framework and isolate those portions of the framework that seem especially important or relevant. We have not abandoned lectures or classroom discussions and these activities also provide opportunities to make connections to the framework. In essence, we have adopted a spiral approach, in which we present ideas and then revisit them in order to reinforce them while at the same time, appealing to our students’ talent for dealing with abstractions.

The innermost circle of the framework contains the most critical ideas that pertain to any communication task – namely, purpose, strategy, and audience. We define the purpose of any form of communication as either to inform, instruct, or to persuade. Students have opportunities to do all three during the term. Once you know the purpose behind some piece of communication, you need to develop a strategy to get your point across. Thinking about strategy means thinking about the tone and form you need to use, as well as the logical organization for the information that works best. One must also analyze the potential audience (will they be hostile or receptive, supportive or neutral, internal or external, interested in the entire artifact or only a part, and so on). It turns out, especially for our students, that this last point is the most critical and the most difficult one to master. It comes down to thinking about communication from the point of view of the receiver rather than the sender.

A second circle fits around the innermost one and contains the skills that we hope students will practice during the term. Here we list reading and writing, listening and speaking, reviewing and evaluating, and finally, thinking. While these skills may be ‘obviously related’ to the thrust of the course, it’s useful to highlight them. Students, for example, will have numerous opportunities to improve their listening skills and we can talk about ways to help them do this. Students are asked to review the work of others and to provide helpful feedback. That is not a skill that they often get to work on and once again, they need guidance. It’s also useful to bring all of these skills into the early part of the course so that students will get a better appreciation early on for the expectations we place on them. For example, students often severely underestimate how long it will take them to produce a ten-page user document. They typically do not factor in time to write multiple drafts, to craft their document to a potential audience that consists of people unlike themselves, and to use technology to produce a document that includes mandated elements such as a table of contents or labeled and captioned figures (some of our assignments insist on compliance with IEEE standards).

There is a third and outermost circle to the framework and here we place process issues and more general traits or qualities. Again, by talking directly about many of these points, we hope to raise awareness and to help students learn how to improve their skills. The following lists some of the issues and qualities we want to highlight.

- Learn to work asynchronously; when presented with multiple tasks, not only prioritize, but take advantage of opportunities to make incremental progress on more than one task by defining, scheduling, and completing subtasks associated with each major task
- Listening is often undervalued and under practiced, so take advantage of opportunities to improve in this area
- Giving constructive criticism that is not mean-spirited and directed at a person, but rather at a work product, takes effort
- Make each instance of communication stand on its own; there’s often a balance point between being concise and providing context and sufficient background that must be found
- Learn to evaluate sources of information (in fact, learn to find information in the first place, since it’s not always the case that all of the information you need either appears in one place or is given to you at one time); are the ideas valid and relevant to your task and does the author of the ideas seem credible?
- Learn to make judgments, especially when you have conflicting recommendations to deal with
- Learn to recognize that ‘small things matter’; meeting deadlines is important; producing exactly what someone else called for is important; providing accurate information the first time is important; somewhat related to this idea is the fact that larger, more complex artifacts are made up of a series of smaller components, each of which contribute to the overall success of the entire item (the parallels here to software development should be ‘obvious’ to most students)

We present below a selection of opportunities and suggestions for how use of such a framework enables instructors to ‘pack more into their course’, covering not only traditional types of assignments and activities, but also activities broader in scope.

4.1 Multitasking

At the start of the term, students are given a calendar which includes when they can expect all major assignments to be handed out and when these assignments are due. Assignments that call for written documents also call for drafts to be produced and suggested dates for completing drafts are noted as well. The course is front-loaded in that most of the major assignments are handed out within the first few weeks. This is done deliberately so that students can see what each assignment calls for. There are subtasks associated with some assignments and these subtasks can be completed early and without investing large amounts of time.

4.2 Coalitions

We tell our students to expect that they will most likely work in teams once they graduate. We offer team-based experiences in some of our courses in order to prepare students. In this course, students form coalitions for the duration of the course. They are
told that coalitions fall somewhere in between groups and teams, but that they offer students an opportunity to practice some of the skills that ultimately make them successful team members. Students review drafts of documents for their coalition partners. Students within one coalition work together in preparing and delivering a group presentation. Part of the benefit from having coalitions established at the beginning of the course is to minimize some of the logistics associated with having students find someone else to work with or to exchange drafts with. Coalitions are free to set internal deadlines that differ from the suggested deadlines on the course calendar, provided final deadlines are met. Students are given an opportunity at the beginning of the term to find out about other members of the class so that they (hopefully) can choose to work with students who they are compatible with. When we talk about personality and temperament, we try to make connections to how this may influence communication. We also point out that different coalition members are likely to have different strengths and weaknesses. If one coalition member is well organized and stays on top of deadlines, that person may be able to help others in the coalition who frequently struggle to remember what’s due next and when. In return, someone else may already be well versed with word processing or presentation software and so they may be able to jumpstart their colleagues.

4.3 Writing a Letter and a Résumé
Certainly, there are standard forms for application letters and résumés and it’s easy to make students aware of what these are. The ‘fun’ comes in when sifting through the myriad of opinions that people offer about what makes for a ‘good’ letter and résumé. Often these opinions are in conflict and so it’s up to students to carefully weigh the advice and then decide what will work best for them. More than any other assignment, this one perhaps best illustrates how important it is to continually refine drafts and strive for perfection. Since a letter and a résumé are often both only one page long, even one error becomes that much more noticeable. As with many other forms of communication, students also get to see firsthand that there may not be a simple way to judge whether a letter or résumé is finished or whether it met its intended goal. For example, the class can discuss a situation where a student sent out their résumé to ten different employers and did not hear anything back from any of them. Was the problem with their letter, their résumé, or something else?

4.4 Meeting Specifications
We know that people supervise or manage us often ask us to complete a task and provide them with specific information, organized in a specific way, and delivered to them by a specific time. Life is often ‘simpler’ if we do just what’s called for. There may be times when we don’t understand all of the reasons behind a request, but it still makes sense for us to comply. This is an example of meeting the needs of the audience or thinking about a task from the point of view of the receiver, rather than just thinking about a task from our point of view. Many assignments carry specifications and students earn a modest amount toward their grade for the assignment by simply meeting specifications. Not following specifications or turning in items late can reduce a grade.

4.5 Oral Presentations
Given a large class and a finite amount of time, presentations tend to be relatively short. Still, they can embody many of the elements that go into successful presentations. We can ask students to prepare remarks, prepare support material to hand out, and be prepared to show visuals of some kind to the rest of the class. Students can propose topics to speak on (within broad parameters), but they must propose their topic using email and they must include certain other information as a part of the email. Many students can’t seem to include the information called for in their email and there are consequences for that in the assignment. If students have to be asked to send a second (or third) email with the information called for, they may not actually get the topic they want or the date and time they want. Email requests are handled on a first-come, first-served basis and students are reminded that they may not get an immediate response to their email because they don’t control the schedule that the message receiver operates under. Students who can’t get something simple done right the first time may lose a small amount of credit on the assignment (not a great deal of credit, just a ‘little bit’). Students also learn in preparing a presentation that the ‘whole’ really is ‘the sum of its parts’ and that each element contributes to the overall success of a presentation.

4.6 Article Review
As an example of a different assignment that has been added to the course, students must find a suitable technical article on which to write a summary and critique. There are several motives behind this assignment, including exposing students to serious technical literature and forcing them to confront issues of authority and credibility. Having students find suitable papers allows us to add another element to the course. Rather than just showing the instructor a copy of a paper and asking if “it’s okay to write about this paper,” students must prepare a memo that will persuade the instructor that the paper not only meets the assignment specifications, but is worthy of review. In many Computer Science courses, we make the assignments and students carry out the work, but there’s little opportunity for students to have any say in the matter. Not all papers submitted for review are accepted, which means some students must find another paper and go through the process all over again. Starting this step early in the term can alleviate the stress of trying to get this subtask successfully completed before the deadline.

4.7 Acting as a Reviewer
We follow a process that in some ways mirrors what a referee does when reviewing a paper for possible inclusion in a journal. Each time a student reviews a draft, they are asked to mark up the draft and also complete a review form. The form calls for an overall assessment (similar to accept, reject, reject subject to revisions) and then more specific comments. Students must offer key recommendations that they wish to see addressed in the next draft. Individual coalitions can go further if they wish. For example, coalition members can decide ahead of time what to focus on during each review, whether or not they should be looking at grammar as well as content, and what kinds of suggestions might be most helpful.

4.8 Examples
We know that examples, especially ones that are negative, are often the most instructive. Students must find and present
examples in class that relate to communication. They can be serious or not and related to technical issues or more general communication issues. While the assignment again gives students practice in writing a memo and preparing to present and lead discussion of examples in class, the real value of the assignment is that it gives students an opportunity to extend the course. There often just does not seem to be enough time during the term to cover all examples of technical communication or to bring up all issues that make communication more or less effective. This is a means of providing students a chance to bring a topic into the classroom that may be of importance to them and one that an instructor might not have thought of.

5. SUMMARY
Communication skills are an important asset for technical practitioners to possess, in addition to their technical expertise. While we recognize this and attempt to encourage our students to improve their writing and speaking skills, the actual creation of either a set of activities or an entire course devoted strictly to communication issues is a challenge for many Computer Science instructors. One can easily bog down in the details, become overwhelmed by grading, and lose sight of more important goals.

We have introduced a high-level framework that seems to place critical skills and concepts in proper perspective. The framework is elastic in that it allows individual instructors to adapt it to their own needs or their own sense of priority. For example, if an instructor thought it was more important for their students to learn how to produce a design document, then that could be substituted for the user document assignment. If more and different kinds of presentations were important, then such activities could be added and others could be removed. The framework allows instructors to ‘see the big picture’ and to then think about assignments that best reflect how to reach selected goals.

Some instructors want to teach the importance of doing quality work and getting work done on time. This framework presents instructors with built in opportunities to carefully integrate these ideas into the course and into the grading system. If it’s critical that students learn to work in teams and to learn how to handle difficulties that often arise, then replace coalitions with teams and devote class time to covering team-related issues. If it’s important that students utilize software technology for particular purposes (for example, to exchange and review documents electronically), then simply require that aspect in assignments.

We work in a field in which change is the ‘name of the game’ and one can certainly also see change taking place with respect to communications. New means of delivering information and collaborating with others take their place on the stage and often will require that we come to embrace new technologies and learn to use them effectively. While it’s important (and often fun) to learn new technologies, of more lasting value to our students is to give them a clear sense of what’s important about any form of communication and to help them learn how a relatively small number of important ideas (such as purpose, strategy, and consideration for audience) can be utilized to frame any form of written or spoken communication.

6. ACKNOWLEDGMENTS
It’s most appropriate to thank Margaret Reek who was one of the founding architects for Professional Communications. Thanks must also go to hundreds of Computer Science and Software Engineering majors who have taken the course. While it may not be apparent to individuals, their comments and suggestions have helped strengthen the course. Finally, thank you to the anonymous reviewers for suggesting changes to the paper and for recommending topics to cover in the presentation. Copies of specific assignments and support materials may be obtained by contacting the author.

7. REFERENCES