

Teaching Communication, Leadership, and the Social Context of Computing Via a Consulting Course

Joseph Mertz
H. John Heinz III College
Carnegie Mellon University
Pittsburgh, PA 15213
+1.412.268.2540
JoeMertz@cmu.edu

Scott McElfresh
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
+1.412.268.4859
scottm@cs.cmu.edu

ABSTRACT

This paper describes a course at Carnegie Mellon University that engages students as consultants working with non-profit organizations. Different from most "service learning" courses, students in the Technology Consulting in the Community course focus on building capacity within organizations to sustain IT solutions without ongoing assistance. They do not merely provide IT support, nor do they focus on system development. Rather they focus on solving organizational problems using IT solutions. In doing so, they may develop a system, or adapt open source or commercial tools as appropriate to the situation. Computing systems do not exist in isolation, but in the context of people, organizations, and their policies. We want to train leaders in our community who can not only develop new technologies, but can solve organizational and societal problems. The course has as its learning goals to build inquiry, communication and leadership skills, in addition to engaging students in project development. At the same time it provides a valuable service in the community.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education — *Computer Science Education*

General Terms

Management, Documentation, Design, Human Factors.

Keywords

Leadership, professional skills, consulting, service-learning, project course, capstone course.

1. INTRODUCTION

Computer science and related disciplines have always had a need to include communication skills as part of training students. [1] Furthermore, students need to be exposed to the applications of computer science and its impact on people and society in order to attract certain segments of the population to the field. [1] Computer technology does not live in a vacuum, but in the context of people, organizations, and their policies that surround

us. We want to be training leaders in our community who can solve problems by developing new technologies.

"Service Learning" in computer science curricula is not new, and many schools have students engaged in the community using their IT skill, typically working with segments of the community that are on the other side of the digital divide. Some have included a service learning component to existing courses, such as CS0 or CS1 [2, 3], while other schools have developed extracurricular activities for students to engage in helping the community [4, 5]. Carnegie Mellon has project courses in which teams are engaged in the community, as well as campus groups that engage students in providing IT support or training to the community. We have created a course that engages students in these types of situations in a unified manner within an academically rigorous course.

The course is titled Technology Consulting in the Community (TCinC). Each student is placed individually in the role of a Technology Consultant for a community organization. Students are led through a consulting process developed to enhance their organization's ability to use technology effectively in order to better meet its mission. Going beyond IT support, the course aims to provide a sustained ability for the organization to use technology effectively, without returning to us for support at the moment of any glitch. In this process, students visit the organization and meet with staff members to learn about all aspects of the organization and their mission. The non-technical aspects of the organization are just as important as the technical environment. Students report their findings and negotiate a scope of work with the organization. The leader of the organization is most often from a non-technical background, but is an equal partner in the work, and thus the project requires the student to work closely with that person. Frequent oral and written proposals and status reports keep the student engaged in honing their communication skills, culminating in a final project report and presentation.

This paper will describe in detail some of the learning goals of the course, how we address them, and assess the outcomes.

2. BACKGROUND

The course is designed for 3rd or 4th year students in computer science, information systems, or electrical and computer engineering. However, any students with some technical background are welcome and can benefit from the course. The student will not develop new technical skills in the course, but rather use the technical skills they have already developed as a context to develop professional leadership, communication, and more broadly, knowledge about the social context of computing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE '10, March 10–13, 2010, Milwaukee, Wisconsin, USA.
Copyright 2010 ACM 978-1-60558-885-8/10/03...\$10.00.

For example, a student with knowledge of Facebook, YouTube, and Twitter, could use those skills (developed outside the classroom) to consult with a mentoring organization interested in using social networking to market to a younger set of volunteers. Or for students in the technical majors, a student who has taken a database class can help an organization design their database to maintain their client information, or to evaluate proposals that have already been made for the development of such a project.

2.1 Course's Role in the Curriculum

For computer science or information systems majors, the course serves as an "in-major" elective. For other students, the course satisfies a general elective.

Prior to taking this course, students have had a course in writing, and some have taken a course in technical communication, but that is currently not a pre-requisite for the course.

2.2 Consulting Partnerships

Each student in the course is individually matched with a leader in a local community organization. This is typically a nonprofit organization, school, or a department of municipal government. The student and the partner are expected to spend at least 3 hours a week together, typically onsite at the organization, and each will be doing work outside of that time as well.

Organizations are solicited via word of mouth and postings to local nonprofit email lists. The leader of an interested organization is engaged in conversation to determine the potential needs of the organization and the organizations' ability to work with a student consultant. The leader is an equal partner in the project, is expected to work hand-in-hand with the student, and gives a large portion of the final presentation.

2.3 History

The course has been taught in fall and spring semesters since spring 1998, with 10-25 students in most semesters. In that time, the course has had over 350 partnerships with 250 different nonprofit organizations, schools, and municipal agencies. Many organizations wish to partner with the course multiple times, which we are happy to do as long as we are not re-solving the same problem.

3. LEARNING GOALS

The course learning goals cover the full process of engaging with a client in order to solve a problem using technology.

Many of these goals overlap the "Social Context of Computing" learning goals contained in [1] and revised in [6]. In the following sections, we will describe the major learning goals of the course, and discuss where they overlap with the Social Context goals.

3.1 Lead a Diverse Working Team

Students seldom get a chance to lead a working team, let alone one that involves anyone other than students. Similarly, "work in interdisciplinary teams" is an oft-stated general education goal, but in practice this most often is reduced to teams of students with different majors. Upon graduation, these students will likely be working in teams that cross many differences, including age, race, gender, language, and technical expertise to name a few.

CS2001 articulates this learning objective as part of the Social Context of computing as:

Indicate the role of cultural issues in considering team-work.

Similarly it addresses the flip side of diverse representation not being included in :

Articulate the impact of the input deficit from diverse populations in the computing profession.

TCinC addresses both of these learning goals by having students lead a diverse team of non-technical (and sometimes technical) professionals of diverse backgrounds and experience. With the student, the team always includes someone in a leadership position in the client organization, and often includes additional management and staff.

To prepare students to assume this role, we begin by discussing the importance of relationship building. We discuss in class the concepts of common ground, shared experience, trust, and social capital, as well as strategies for developing each.

We also have students consider the alternative consulting roles they could play. Most students would default into being an expert-for-hire. This is most closely related to the typical jobs and internships they have held; they have some labor or skill to provide, and so they should get on with providing it. Instead, we teach a capacity-building model of consulting. In this case, the student's goal is help an organization expand its ability to use technology to better meet its mission, in a way that is sustainable after the student consultant leaves. System development is not the goal. Rather the goal is mission focused, for example:

- To improve mental health services by using a database to better track patient assessments of health care providers.
- To feed more hungry families by helping a food bank more effectively use the web to inform donors of the need.

We assess how student consultants have achieved this goal in a number of ways:

- Clients provide survey feedback not only on student performance, but also on how the student increased their ability to manage the technical advances on their own.
- The client and student consultant do a joint presentation, in which the client speaks to the outcomes to them and their organization resulting from the student's leadership.

This feedback is almost always very affirming to the student. Before the semester begins, there are often some students who express concerns that they don't know enough to be useful as a "consultant" to an actual organization. Near the end of the semester they document their perception of the outcomes in a final consulting report, and in person sometimes fret that their results were not technically sophisticated enough. But in many cases, it is not until they hear their client speaking about how their help has impacted the organization that they get a full appreciation of the value of the leadership role they played. The clients typically leave happy because when the student is focused on building capacity, then invariably they provide some residual value, even if their technical solution is sometimes lacking (which is inevitable in a student population).

3.2 Assess a Complex Technical Environment

A second challenge students seldom get the chance to experience is how to implement a technical system in an authentic organization. Technical systems live (or die) within social systems. Users needs change with time; companies release new software updates regularly; new hardware is purchased every few years. It is impossible to create a robust system that takes all potential social changes into account. Rather, computing professionals need to understand and account for the social systems in which the technical systems are embedded.

Furthermore, the value of technology is not intrinsic, but is derived from its ability to solve individual, organizational, or societal problems.

CS2001 includes two related learning objectives:

Interpret the social context of a particular implementation.

Identify assumptions and values embedded in a particular design including those of a cultural nature.

We introduce students to the problem of neglecting the social context by using a documentary video that tells the story of two approaches to providing clean drinking water to a village in Togo [7]. In the first approach, a well is drilled and a pump installed without any allowance for maintenance and repair. Of course, the pump eventually breaks and villagers must return to collecting disease-laden surface water. In the second story, government extension workers take into account the social context and work with villagers to build a social system to support the technological water system. We drive the point home by using a short video taken from the final presentations of an earlier TCinC class in which the client discusses how a student team from another project course built them a system that did not take their organizational culture into account, and the system failed. In contrast, TCinC student consultants helped his organization build a solution in Microsoft Access that they could manage and evolve. In the client's words:

"What impressed me the most was their ability to listen and to help assess our needs and then to help us to think about them critically. And they didn't really come up with this solution for me, I felt like I was able to come to this myself. Again, just like good counseling or therapy, you come to that and you can use it in life. If someone gives you a magical solution like the other group did it doesn't work."

But just being aware that there is a social context is not enough; students have to have tools and strategies to investigate and understand it.

Students begin by reviewing local and national benchmarks on how technology is used by nonprofit organizations. This gives some context with which to assess what they observe with their client. We also provide a broad framework for investigating how technology impacts an organization. This framework includes suggested areas to explore along on 10 dimensions, including:

1. Organization – e.g. mission, history, governance structure
2. Programs – e.g. day care, job training
3. Staff – e.g. what applications do they use regularly
4. Facilities – e.g. adequate power and lighting
5. Technology infrastructure – e.g. hardware & software

6. Technology management – e.g. troubleshooting & support
7. Technology planning – e.g. is there a planning committee
8. Information management – e.g. databases or paper forms
9. Communications – e.g. staff email addresses, web site
10. Business systems – e.g. accounting or inventory systems

We teach students to use both interviewing and observation[8].

In terms of evaluating students learning, students articulate their assessment of the organization in a working document that goes through a repeated write-review-revise cycle. This cycle is discussed in more detail in Section 3.4. and is the basis for providing not one but multiple evaluations of the student's ability to meet this goal. In general, students do not have much trouble providing a descriptive account of their client's organizational context. Sample assessments of other organizations are helpful for students to model their work on. Of course, some follow the example too closely. One semester it was curious why several students had included an accounting of the number of bathrooms in an organization until it was found that the first example report provided to them had mentioned a bathroom while describing the facilities.

3.3 Structure a Complex Problem

A third useful challenge for students to learn from is to have to wrestle ambiguity and complexity while structuring an otherwise unstructured problem. All instructors know that problems outside the classroom tend to not be as tidy as those we use for homework assignments.

Students in TCinC do not receive a clear problem statement, because the client often does not have one. Rather, the student consultant has the interesting task of working with their client to investigate problems in the organization, decide which of those problems are important to solve, and what alternatives exist to solve them.

We help students to structure these unstructured problems in a number of ways:

- We practice multiple diagramming techniques, including Cross-Boundary Causality Maps [9], causal flow diagrams, and work process diagrams [10] in order to analyze problems within the context of the organization.
- We differentiate between problems (e.g. losing patient information) and solutions (e.g. a database).
- We emphasize the primacy of mission in any organization, and that the only reason to implement a technical solution is because it will positively impact their mission (e.g. by solving a problem that is hindering them from achieving their mission more fully).
- We assess alternative solutions in terms of maximizing positive impact while still being feasible.

In this way we lead the student through a process by which they define a scope of work. Ideally the scope of work clearly prescribes a solution to a problem that is important to the organization because it impacts their mission; and they can argue why that solution is better in terms of impact, feasibility, and risk than the alternatives.

The student documents this scope of work in a report that is circulated to the client and the instructor, and is the basis for

evaluating the students attainment of this learning goal. Therefore it is straightforward to assess how each student has or has not been able to clearly structure and communicate the reasoning that backs the solution path they are taking. Not all students are highly successful at this. All carry on and do something with their client, but the reasoning and planning behind what they are doing is not well defined. Therefore the degree of mastery of this skill is a significant differentiator of students' overall performance in the class. And there is certainly room for finding better ways to help students systematically structure complex problems.

3.4 Communicate Professionally

A fourth useful challenge for students to learn from is to communicate professionally. There are a few aspects to this communication that TCinC helps students develop:

3.4.1 Use Writing Skills to Maintain Working Documents that Describe, Plan, Persuade, and Coordinate Work With Others

TCinC is a very writing intensive course, but it is writing for which we provide clear guidelines and examples. For students who prefer coding over writing, we joke that TCinC is not a writing course, it is a programming course in which the programming language is English. We model in the class that the same writing can be used for multiple purposes:

- To help the writer clearly understand their own reasoning.
- To allow the client to validate or correct the consultant's understanding.
- To share thinking and coordinate the work of the project team.
- To inform stakeholders of the progress of a project.
- To provide an archival record for others to refer to in similar situations.

In TCinC, the student maintains working documents that serve these purposes. By working documents, we mean that the documents are repeatedly updated and shared as the project progresses in a write-review-revise cycle.

For example:

- The student consultant produces a description of the organizational context they are working in and share it with peers, the teaching assistant, and a volunteer mentor.
- The peers, teaching assistant, and volunteer mentor provide written feedback, as well as oral feedback in a small-group meeting.
- The student consultant adds the scope of work to the document and again circulates it, this time also to the client.
- In a small group meeting, a peer orally presents the student consultant's scope of work to the group and it is discussed. In this way the student consultant can observe and listen more closely to the feedback.
- The student consultant revises the description of the context and the scope of work and submits it to the instructor. The instructor provides extensive written feedback and a grade.

Nearer the end of the semester, this write-review-revise process is repeated with the student's documentation of outcomes and recommendations for future work.

Finally, the student consultant takes all the pieces and synthesizes them into a final consulting report. This is distributed to all stakeholders and some are archived on the course web site for future reference.

Students are given clear guidelines for the expected content of the reports, and these guidelines are used to evaluate the students success in meeting the communication goal.

The quality of the students' reports varies, as do the grades they receive; nevertheless, the documents do provide a basis for the student, their client, and the instructor to have a common understanding of the project. Rarely does the client report that the document is significantly flawed, and most of the time find it mostly accurate, complete, and ultimately publishable. In that respect, all students experience the role working documents play in coordinating project work.

3.4.2 Communicate Technical Ideas to a Non-Technical Audience

Another aspect of communicating professionally is to be able to communicate technical ideas to a non-technical audience. Students are immersed in a sea of technical terms, and may have been their whole lives, so that they might fail to realize that those who are not so immersed may have no idea what they are talking about. Most students have not stopped to think that "semaphore" and "daisy chain" are only metaphors taken from the physical world, and that others may not understand the association.

In order to help students be aware of and plan for communicating clearly, the class:

- Discusses the use of metaphors in computing and ways of assessing their value in communicating.
- Recommends that the student models technical skills while subtly giving a verbal think-aloud protocol so that others can learn while watching.

The write-review-revise cycle also helps to flag where communication is unclear to non-technical readers. Throughout the process, the client is asked to review and comment on the clarity of the document.

Finally, the student is expected to make an oral presentation, with the client, in a public forum at the conclusion of the semester. This event is another opportunity for the student to practice communicating to a larger audience in a formal presentation, and provides another assessment opportunity for the instructor.

This is an area of the class that students uniformly do well at. Because it is treated as a priority, and one that is not difficult to carry out, students are good at communicating to their non-technical clients. Clients routinely comment on how much they value the patience with which students help them to understand bewildering technical concepts.

3.4.3 Document Outcomes Objectively

The final communication challenge for students is to learn to document outcomes clearly and objectively.

It is very easy to document outcomes poorly. For example, by:

- Leaving out detail: "the process is faster"

- Using summary statements: “the reporting activity is now more efficient”
- Describing what was done instead of what resulted from that action: “I trained the staff to use of the database.”

It is easy to leave the reader with impressions of outcomes without providing any clarity on what exactly the final state was. It leaves this interpretation up to the reader. This is useful when trying to gloss over imperfect results, but it is not useful in providing a clear assessment that in turn can guide future work.

CS2001 articulates a similar learning objective:

Evaluate a particular implementation through the use of empirical data.

We use several practices to help students meet this learning goal:

- When crafting their scope of work, students are taught to define measurable expected outcomes.
- We explicitly review and discuss how language can be used to present clear and concrete evidence about an outcome.
- The write-review-revise cycle is used in order to give feedback to students on where their outcomes are not clear.
- We emphasize the value of negative results alongside positive outcomes, for negative results are useful new information about approaches that have not been successful.
- We try to reassure the student that a well documented negative outcome is far superior (and is graded such) to an ambiguously worded outcome that suggests success.

Students often have difficulty documenting outcomes objectively. And like the ability to structure a complex problem, this skill is a second significant differentiator of students’ overall performance in the class.

4. IMPLEMENTATION EXAMPLES

What we have not described in this paper is the implementation of the actual project, for it follows more conventional lines. After completing the scope of work, the student generates and executes a work plan. The type of projects students complete are quite varied for they are custom to the needs of each organization. Nevertheless we will mention a few representative ones:

- Developed a system to better manage the production process of Braille books by improving coordination of information among employees involved in the production process .
- Defined requirements for a library management system, researched alternative solutions, then installed and configured the open source Koha system.
- Designed and implemented a web site using the open source Joomla content management system. Extended Joomla with the open source CiviCRM to keep client, partner, and donor information, and integrated LimeSurvey to accept applications to their program.
- Developed a system to enable patients to make requests for Medical Alert Jewelry on-line. The system was implemented using Microsoft SQL Server and Cold Fusion.
- Developed a system to support coaching feedback to swimmers using video. The project involved choosing appropriate equipment, testing camera locations, integrating hardware and software systems for editing and archiving the video, and training the coach on its operation.

- To support the management of multiple real estate re-development projects, researched shared project-management solutions and helped the organization adopt the hosted Ace Project tool. To help the organization better manage versions of documents related to each project, installed, configured, and trained staff in the use of the open source tools Sub Version and Tortoise SVN.

5. CONCLUSION

This course has been taught continuously each spring and fall since 1998. Clients often will repeat with the course because of the value they get from the student consultants. The median “Overall Course Evaluation” by students is over 4 out of 5.

In conclusion, we have developed a course that successfully teaches our students professional communication and leadership skills that are important for leaders in the discipline. By involving them with actual clients in the community and providing a structured consulting class to support their engagement, students are able to develop many of the Social Context of computing skills that are necessary for leaders of the future.

More information about the course can be found at our web site: <http://cmu.edu/tcinc>

6. REFERENCES

- [1] Computing Curricula 2001: Computer Science, Final Report, December 2001. The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery (2002).
- [2] Adams, J. B. and Runkles, E. 2004. "May we have class outside?": implementing service learning in a CS1 curriculum. *J. Comput. Small Coll.* 19, 5 (May. 2004), 25-34.
- [3] Rosmaita, B. J. 2006. Accessibility first!: a new approach to web design. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. ACM, New York, NY, 270-274. DOI=<http://doi.acm.org/10.1145/1121341.1121426>
- [4] Brooks, C. H. 2008. Community connections: lessons learned developing and maintaining a computer science service-learning program. *SIGCSE Bull.* 40, 1 (Feb. 2008), 352-356. DOI=<http://doi.acm.org/10.1145/1352322.1352256>
- [5] Christensen, K., Rundus, D., Perera, G., and Zulli, S. 2006. CSE Volunteers: A Service Learning Program to Provide IT Support to the Hillsborough County School District. *SIGCSE Bull.* 38, 1 (Mar. 2006), 229-233. DOI=<http://doi.acm.org/10.1145/1124706.1121412>
- [6] Computer Science Curriculum 2008: An Interim Revision of CS 2001. Report from the Interim Review Task Force includes update of the CS2001 body of knowledge plus commentary. Association for Computing Machinery, IEEE Computer Society (2008)
- [7] *The Water of Ayole: Togo, West Africa.* [videorecording] Directed by Sandra Nichols, Sandra Nichols Productions, USA. (1988)
- [8] Beyer, H. R. and Holtzblatt, K. 1995. Apprenticing with the customer. *Commun. ACM* 38, 5 (May. 1995), 45-52. DOI=<http://doi.acm.org/10.1145/203356.203365>
- [9] *To Think Bigger Thoughts: Why the Human Genome Project Requires Visual Language Tools*, Robert E. Horn, <http://www.stanford.edu/~rhorn/a/recent/SpchNBICHrnSpks.html>
- [10] *Contextual Design: Defining Customer-Centered Systems* By Hugh Beyer, Karen Holtzblatt, Morgan Kaufmann Publishers, 1997.