Achieving and Assessing Service in Computing Service Learning: Lessons from Computing for Good

Ellen W. Zegura
Professor, School of Computer Science
Georgia Tech
Atlanta, GA 30332
ewz@cc.gatech.edu

Abstract - In the last 10 years, computing departments have introduced service learning in the curriculum at multiple levels, from introductory computing to senior design/capstone. Service learning combines discipline-specific learning content with real-world service in a form of experiential learning. Yet as Connolly observes in his SIGCSE 2012 paper “Is There Service in Computing Service Learning?” the presence of “true service” in these offerings is under examined in comparison to the focus on student learning outcomes. As far back as 2007, George and Shams highlighted in their IJSLE paper the challenge of assessing the closely related concept of “customer satisfaction” in international service learning. We aim to fill this examination gap by drawing on our experiences over the last five years teaching a senior-level Computing for Good class. We identify categories of risk to achieving service and key challenges within each category. We describe our course structure and the mechanisms we use to overcome challenges in achieving true service. We then turn to assessment of service and assemble evidence that many of our course projects have external value against several metrics. We provide guidance to others interested in creating or adapting engineering and computing service learning courses so that learning and service occur in roughly equal measure.

Index Terms - senior capstone, computing education, community impact, assessment

INTRODUCTION

Service learning combines discipline-specific content with a real-world service setting. The concept of service learning is more than 40 years old, with the term first used in the mid-to-late 1960s. Initially embraced largely in the social sciences, the last ten years have seen engineering and computing departments introduce service learning at multiple levels, from introductory to upper division software engineering and capstone courses. The literature contains many good examples of experience papers that describe projects, highlight challenges and discuss approaches to overcome them. Two working groups under the auspices of the Innovation and Technology in Computer Science Education (ITiCSE) have focused on “social good” in computing curriculum generally and in introductory courses in particular. However, as noted by Connolly, under examined is the benefit of service learning projects to the community served. George and Shams similarly call for including measures of partner benefit in assessment.

In the best case, learning and service to (or with) a community occur in roughly equal measure. In practice, care is required to achieve this and to avoid either leaning too heavily
towards service with little real learning, or leaning too heavily towards learning while creating only superficial value to the served community. The first of these is more accurately termed community service, and has both a place and value in co-curricular student experiences. The latter of these, however, carries significant under-studied risk. As Connolly observes for work in the computing service learning literature, “Almost without exception, all the computing papers examined for this study assume (directly or indirectly) that by providing free computer consulting work for non-profit organizations, they are, by definition providing a valuable community service.” And yet “some service-learning projects may, despite the best of intentions, actually be harmful for the recipients and/or their surrounding community”. 19

As far back as 2007, the need to include measures of customer or partner experience as part of assessment was highlighted by George and Shams in the context of international service-learning projects. They offer three assessment questions that speak to customer satisfaction and the challenges therein: “Have the customer’s needs been met?” “Is the project sustainable and maintainable by the customer?” and “Does the project respect the environment and make effective use of local renewable resources?” 20 Based on their own experiences and best practices culled from other sources, they offer six elements for success in meeting customer needs, tailored to the international context. We return to these elements in the next section.

We aim to contribute to the relatively understudied assessment element of service in service learning. We do so by drawing on our experiences over five years teaching a senior-level Computing for Good (C4G) course. Following a pilot in Spring 2008, we have offered Computing for Good every fall since 2009 as a section of our senior design capstone course and as a cross-listed MS level elective. Students can enroll from undergraduate majors Computer Science and Computational Media; graduate students from multiple degree programs can enroll, while in practice most come from the MS Human Computer Interaction (HCI) program or the MS Computer Science. To date, 111 undergraduates and 96 graduate students have enrolled in the class. (Unless otherwise noted, course data is inclusive of Fall 2009 to Fall 2013. The pilot offering from 2008 is not included.)

Similar to other computing service learning offerings, the course centers around team-based projects undertaken with partners who are working locally and globally on pressing social problems. Partners have spanned the full gamut geographically, from hyper-local (e.g., on our campus) to international (e.g., CARE). Partners are active in many sectors including health, homelessness, education, economic development and democracy. The processes for creating and sustaining partnerships are key to the success of the course and are discussed in some detail later.

The course has focused from the outset on the importance of delivering real value to the partners judged on their terms. The origins of the course are in part responsible – rather than coming to service learning from pedagogical considerations, the initial impetus for the course came from discussions among a small group of faculty who held the somewhat naïve conviction that computing could and should make a difference in the world. The initial planning group of faculty was largely ignorant of the service learning literature when they began. They included, however, a faculty member whose long-standing research expertise is in computing for international development, an interdisciplinary field that grapples at its core with the potential and limitations of technology to make a positive difference in highly challenging settings. Not coincidentally, Connolly draws a cautionary analogy between recipients of service learning and recipients of foreign aid in the developing world.
In this paper we describe our experience teaching C4G over five offerings. We outline a set of considerations for achieving service in service learning. We then describe the ways that our course is structured and connect that to methods to achieve these considerations. We provide evidence of positive service impact via quotes from partners and other quantitative data. We aim to provide guidance to those creating or adapting service learning opportunities in computing or engineering so that service and learning occur in roughly equal measure, and we place our recommendations in the context of those offered by others who focus on service value.

**SERVICE CHALLENGES AND RELATED WORK**

Significant challenges are present in both assessing service in service learning and achieving service in service learning. Both matter, of course. We identify the following challenges for achieving service in computing service learning; we return to the question of assessment of service later in the paper. We divide challenges to achieving service into three categories based on where the risk lies. Other authors have identified some of these risks as having an impact on projects and student-centered outcomes of service learning, we are interested in and thus highlight the potential to interfere with service impact.

**Student-oriented risks** include:

- Insufficient time allocated to the project by students, resulting in output that is incomplete and unusable by the partner.
- Insufficient student expertise in the technical domains required by the project, resulting in output that does not meet the required technical specifications.
- Insufficient student appreciation of the broader context and constraints in which a technical solution resides, resulting in output that is unusable in a cultural or contextual setting.

**Partner-oriented risks** include:

- Insufficient time allocated by the partner to student interaction, resulting in partner needs that are not adequately communicated and a project that does not meet these under-specified needs.
- Inability on the part of the partner to deploy the solution, due to lack of planning or understanding of what deployment will require, resulting in a deployable but not deployed project.
- Tenuous or immature partner commitment to real social good, resulting in a project that the partner may be satisfied with and the partner may deploy, but that has little to no real impact on pressing social needs.

A **shared student-partner oriented risk** is insufficient attention to sustainability, resulting in a project that may be successful initially but cannot be maintained by the partner over time. This risk is shared because both students and partners must dedicate time and attention to sustainability.
Project-oriented risks include:

- Project scope that is too large for the available time, resulting in inability for even the most dedicated students to complete the project.
- Project scope that is entangled in considerations out of the control of the students or partner, such as dependence on decision making by other organizations.

In the computing service learning literature, Rosmaita describes a course structured to meet a set of problems he identifies with integrating service learning into a course. His problems include what he calls “asymmetry”, which is closely related to service as it expresses an inequality in what students get from the experience versus what the partner gets. Rosmaita solves the problems he identifies by the careful construction of a course with a specific, tightly scoped project, namely auditing a website for accessibility and general usability. Students in his course begin with a discussion-based exposure to accessibility consideration. Following this background, teams of three students conduct an audit of a specific non-profit website and then complete the redesign of one web page using accessibility principles. The audit is delivered to the non-profit web master and a demonstration of the student web page is provided. With this careful scoping, issues of iteration on design, sustainability and organizational risk are sidestepped, though at the cost of narrow project scope.

George and Shams identify six elements for success, three that focus on preparation by the faculty member in (1) beginning a relationship with an on-site partner, (2) choosing a suitable project and (3) building a relationship over time with the on-site (international) partner. In the risk structure we propose, these largely fall into partner-oriented and project-oriented risks. The fourth element -- executing the project --- covers many issues that we identify as student-oriented risks and shared partner-student risks associated with sustainability. The fifth element of long-term involvement is a structural challenge related, among other issues, to the mismatch between academic semesters and project timelines. Finally, defining success is an element that goes directly to the issue of project goals and assessment of service. George and Shams connect their personal experience to these six elements for success by describing a project in Haiti to harvest breadfruit.

In contrast and complement, we develop and describe practices for mitigating risks that include significant elements of curricular design and class structure that can be adopted regardless of computing or engineering discipline. These practices go along with effort on the part of individual faculty to ensure success, and may even serve to institutionalize successful practices in ways that reduce some of the burden on individual faculty.

**Course Structure**

Our course structure has a strong bearing on how we overcome the challenges outlined above. Class time periods are used for four types of activities. *Project presentations* are frontloaded. These are relatively short overviews of each project context and opportunity, with some limited time for student Q&A. Ideally a representative from the partner organization gives the presentation so that the students hear first hand about the problem and the motivations of those involved in addressing it.
Project management sessions are spread throughout the term with greater frequency early on as student teams are getting established, organizing roles, understanding their stakeholders and developing initial priorities for the project. The course teaches the Agile project management and the Scrum methodology, which are also covered in standard senior design. Agile and Scrum is a framework for managing complex projects whose requirements are changing and/or emergent over time. While frequently used for software projects, the methodology is directly applicable to any project. Over the last two offerings, a software project management consultant with personal ties to our faculty taught the project management sessions. It is not necessary to bring in special experts – there are good resources available for regular instructors to pick up and teach project management. We also find that more of our students recently have been exposed to Agile development in summer jobs and internships, though we do not assume this means they have mastered it.

Readings and class discussion sessions occur with semi-regularity, to fit in with other activities. Readings and discussions are a key component for overcoming some of the service challenges.

Lectures cover key topics that students encounter as they work on their projects and may dovetail with readings. Lectures include topics such as sustainability (what is it, how do projects fail to achieve it, what can be done to improve sustainability), system design and tools (what components show up in many project designs, how should existing tools and open source software be identified and evaluated, when should new code be written), user interface design (what are the basics of good UI design, what considerations are present for different user populations).

Students are assigned to project teams at the completion of the project presentations. We allow students to request other students as team members. Undergraduates are much more likely to request team members; graduate students are often in their first semester and don’t know other students well enough to pre-form teams or partial teams. Each individual or group provides an ordered list of three project preferences. We create teams of 4-6 students, with occasional smaller teams. Perhaps surprisingly, in practice we find it possible to assign nearly all students to their first or second choice of project.

As a first activity, teams must complete a team contract that establishes the logistics of meetings, roles and accountability. A team may choose to all receive the same project grade or to evaluate one another as input to project grade decisions. Not surprisingly pre-formed teams have a strong preference for the “same grade” option while teams we assign generally chose the “peer evaluation” option.

The course culminates with a poster session, project demonstrations, and reception open to the campus and external community. Most project partners send multiple representatives. Potential partners attend to learn more about how the class works and what types of projects are suitable. Members of the upper university administration attend to see what student effort and ingenuity can accomplish in a real-world setting. Some years have featured a panel as part of the activities. In Fall 2013 the event included students from multiple courses across the university that have a community engagement and social impact component.
MEETING SERVICE CHALLENGES

We consider the challenges identified earlier and connect them to aspects of the course and strategies used by the instructors in preparing and managing the course. When the same method addresses multiple challenges, we group the challenges.

**Mitigating student time risk**

We have attempted to mitigate risks resulting from insufficient student time through frequent interaction with the instructor advising each team, regular class check-ins where peer-pressure can be applied, and a class culture of responsibility towards the partner and the community served. Close project advising is critical and requires considerable instructor time outside of class.

**Mitigating student expertise risk**

Our system of allowing students to state project preferences, combined with the senior and MS level of the students, seems to have largely eliminated the risk of insufficient technical expertise. Even if not every student on the team is exceptionally comfortable with the technology, the teams have proven able to manage diversity of skill sets. Not every project succeeds, of course, and our quality control methods are limited by the fact that these are students taking a course, not employees in a company.

**Overcoming insufficient student understanding of the broader context**

Technologists understandably tend to place technology in high regard. Many computing students and instructors have little exposure to the tempering and context-setting elements of social science including history of science and technology, which focus on the interactions between society and technology.

We address this tendency towards technological determinism in two ways. First, we include readings that expose students to the literature in history of science and technology. For example, we read and discuss the classic Langdon Winner paper “Do Artifacts Have Politics”, a paper that is not easy for students to read, but which contains compelling examples of technology-society interaction. Second, we provide as much exposure as possible to the partner and the community to increase awareness of the context where the problems and solutions reside. For community exposure, where feasible, students volunteer or otherwise spend time at the organization.

**Overcoming large project scope and evolving partner needs**

Projects that last for only one semester are necessarily tightly constrained in scope, especially when one considers that student technical skills are still under development and students have many other calls on their time. And yet iteration with a partner is an essential component of understanding and working to meet needs. Partners may not be able to articulate their real needs until they experience a prototype solution. Within the semester, the Agile methodology insists on delivering prototypes to customers on a tight iteration cycle. These can include paper prototypes and mock-ups that can be produced relatively quickly yet provide valuable feedback.
Projects that can persist beyond one term enable valuable iteration. One model for these projects has been to repeat the project in subsequent offerings, with the instructor providing the continuity from one project group to the next. Because we don’t offer C4G every term, however, this produces gaps in effort that are problematic. A complementary successful model for these long-running relationships has been for students in the class to work beyond the class semester, bridging to the next offering. MS students can work as graduate research assistants if funding from the partner is available, or they can work for credit towards their MS degree using our project or thesis options. Undergraduates can earn credit for independent study. To date about 16 MS students worked on projects for pay or credit and about 8 undergraduates did the same. The undergraduate numbers would likely be higher except students in senior design tend to be graduating thus are not around to continue.

Starting in the summer of 2014 we have developed a summer internship program at our Georgia Tech that provides paid internships for students to work on projects with partners full-time. Modeling after the successful Data Science for Social Good program at the U. of Chicago, our program offers another option for beginning or continuing work with partners. The pilot offering of the program was funded through a gift from the Oracle Academy and internal university funds. More information can be found at dssg-io.atl.

Achieving sustainability

even a long-running relationship with a partner can and should eventually transfer ownership and responsibility from the university and students to the partner. Many available partners have extremely limited IT staff and expertise; indeed this is one of the factors that makes free and semi-skilled student labor so attractive to them. If sustainability does not get early and first-class consideration, students can easily make early design decisions that doom the project to failure in the future.

Each time the course is offered, we have placed greater emphasis on sustainability of student solutions. In the current term, we shared early on an example of a sustainability failure from the prior offering, where the student who had the only copy of the final code disappeared after the term and could not be located. In a subsequent class, we read the blog post “A Ugandan mHealth Moratorium is a Good Thing”, by David McCann and used that as the basis for a discussion on the risks of unfettered, independent solutions that do not plan for future data or system interoperability. We are requiring students to produce an initial sustainability plan prior to any development work, and we require a final plan as part of the deliverables to the partner.

Mitigating partner-oriented risks

High quality partnerships are one of the most important elements for project success. Strong partnerships ensure that project work takes place in a robust organizational structure with seasoned and dedicated staff. Tucker et al. synthesize excellent guidelines for creating and sustaining strong partnerships in learning through service efforts. Echoing Tucker, we find that the most successful partners are those who can make staff available to interact with student teams throughout the semester, as project goals and priorities are set, as sustainability plans are developed, as testing and evaluation proceeds. Staff time increases partner stake in the success of the project.
Early on we solicited ideas for projects and partners from College of Computing faculty and staff. Our work with the Carter Center came from such a solicitation; a staff member connected us to his wife who was just beginning a project on Mental Health in Liberia. Many partnerships have developed from personal contacts between course instructors and individuals at organizations. Our work with the Centers for Disease Control and Prevention (CDC) falls into this category. One instructor knew staff at the CDC through a playground in his neighborhood. Word of mouth from satisfied partners also produces potential new partners.

Especially valuable have been partners whose organizations are established and stable and who seek a long-term relationship with Georgia Tech. Projects from these organizations can and do span multiple offerings of the course and students have sometimes continued to work on projects after the class ends via independent study options. Follow-on work has been most successful at the MS level since undergraduates taking the senior capstone are usually very close to graduating. We have had seven partners with projects in more than one term --- CDC, United Way, Georgia Tech, the Carter Center, the Marcus Autism Center, Emory University and CARE. In some cases these projects have resulted in grants or contracts with the university and/or other arrangements regarding the working relationship, access to data, etc.

**Mitigating project-oriented risks**

Partners must be carefully selected, but projects also need to be subjected to acceptance criteria. In terms of project work, we see several themes that are largely specific to computing. Regardless of domain, it is useful to think about projects in finite categories based on how they impact service considerations. One theme we encounter is updating the website of an organization to make it more user friendly, more manageable by staff, more compatible with mobile devices, etc. For example, a Fall 2011 team created a new website for the State Campaign for Adolescent Pregnancy Prevention. While computing students are well suited to these types of projects, the projects often bear a very indirect relationship to the work of addressing a pressing social problem. We sometimes turn away web development projects or refer them our non-service oriented senior design sections.

Another theme is to convert a paper process into a robust electronic process. Organizations that have limited IT staff and limited resources for IT often employ paper or basic Excel spreadsheets for data collection and analysis. Converting to a more robust and flexible system can offer substantial benefits. As an example, a long-running project with the CDC has built a customizable system for laboratory test management, to enable record keeping and track results in developing world clinics. The Basic Laboratory Information System is currently being used in four countries.  

Depending on the setting, data collection on mobile devices may be called for. A Fall 2013 project will partner with the Emory University School of Nursing and the Farm Worker Family Health program to convert paper charts to electronic form. Health services include visits with migrant workers in the fields, where handheld devices or tablets would be more practical than laptops. Robust electronic data collection projects often involve a customized UI and a database backend possibly with analytics and/or data visualization. Prior projects have used tools such as Frontline SMS for text-message-based data collection and Open Data Kit for handheld data collection. Some projects have required the technical support for work off-line with upload to a server when connectivity is available.
Projects that develop robust electronic data collection are appealing because they tend to be intimately linked with the primary work and workflow of the organization. Designing suitable systems requires an understanding of the work done and the constraints encountered. Electronic data collection also has so many advantages over paper processes from the standpoint of record accuracy and data preservation that organizations readily derive value once these systems are in place.

A third theme concerns education projects that have allowed students to use their recently attained skills in computing to create learning experiences for other students. For example, a Fall 2011 team created a wearable computing activity for our computing outreach organization to use with middle school students. The project involved a glove that had sensors on the fingers wired to an Arduino board with light and sounds associated with each finger. The exercise also involved programming a repeat-after-me pattern game. Another Fall 2011 team created materials to teach Python programming to adults at a technology hub in Liberia. These types of projects have the appeal of tapping into the unique learning experience of students.

**Evidence of Service Impact**

The previous section outlines the techniques we have developed to mitigate risks to service impact. We now turn to the question of evaluating service impact, a notably difficult task. Defining what it means to achieve successful impact is not easy. Relying on assessment from instructors of the course or even partners can pose issues of subjectivity. Impact is often not immediate, but rather becomes clear only over time, suggesting the need for longitudinal study. We take steps to address these challenges by defining three variations of “successful impact” and by using both our own judgment for which projects achieved success and an indirect, but less subjective measure of success, based on repeat projects from the same partner. For a few projects where the data is available, we describe measures of long-term impact based on deployment.

We begin by analyzing the disposition of projects undertaken over the history of the course. Based on data from Fall 2009 to Fall 2012, about 40% of the projects were “one shots” in the sense that they were completed in one term, delivered to the partner, and put into use by the organization in one of several ways --- deployed immediately, used as the software framework or foundation for additional development, or used as a prototype to inform later development. For these one shots, that concludes the relationship. We view each of these types of partner usage (immediate deployment, foundation for future work, and informative prototypes) all as variations on success, though the last of these begs for longitudinal study to determine just how informative the prototype really was.

By our own analysis, a small number of projects (3 out of 41) have been failures in the sense that the team was unable to complete work that was useful to the partner in the ways outlined above. All of these failed projects had in common that the team did not devote the necessary time and dedication to the project work, a well-known risk for student projects, and one of the most difficult risks to mitigate with 100% success. Indeed, that is likely an impossible goal.
“Repeat” Partners

A strong, though indirect, indicator of success is the presence of projects over the five-year history where the partner is a “repeat customer”, indicating that the prior experience had such value that the organization was interested and willing to devote time to another round. About 60% of the projects have taken place as part of a long-running partner relationship. Students complete and deliver their one-semester project, but their work is part of a larger project undertaken between Georgia Tech and the partner or the partner engages with Georgia Tech on more than one project.

Partner Feedback

A second form of evidence of service impact comes from asking partner organizations to describe what impact the relationship with C4G has had on their organization. Here is a sample, with three quotes from repeat partners (United Way, Carter Center, CDC Vein-to-Vein Program) and one from a first time partner CDC Autism Program). Because of the heavy representation of repeat customers, the quotes may contain some bias.

The United Way highlights the impact on organizational effectiveness and the cost savings resulting from working with a faculty member and student group whose time is free to the organization:

*C4G has allowed our organization to provide greater accountability to our donors and partners and allowed the service community to track the results of our effort to end homelessness. It has made tracking easier and has saved us thousands of dollars by allowing us to track our efforts and serve the homeless better. And it was all developed pro bono by C4G.* – Protip Biswath, United Way

The CDC highlights a common theme – the development of a small project into a larger version that will reach a greater audience and increase impact:

*Working with C4G has enabled the CDC’s Learn the Signs. Act Early. Program to take a huge step forward in reaching parents of young children with information about their child’s development through mobile technologies. We would not have been able to develop an app with this information as quickly or as efficiently without the great work of the C4G team. The project was such a success that one of the students is continuing the work for his Master’s thesis to test the use of the app as a way to get developmental information to moms of young children enrolled in Our State WIC. – Cathy Rice, CDC*

The Carter Center echoes the evolution that the CDC noted, namely that a relationship that started small has become critical to program success:

*When the relationship between The Carter Center’s Mental Health Liberia Program and Georgia Tech’s Computing for Good program began in October 2010, it was originally envisioned as a method to incorporate patient data collection software into the training regimen of mental health clinicians in Liberia. The connection was intended to be a relatively small, yet instrumental, fix for The Carter Center’s gap in data collection methods for mental health patients in rural West Africa. Instead, over the course of several semesters and after discussions on the types of problems that could be addressed by working together, the collaboration between the Mental Health Liberia program and the Computing for Good program grew into a pivotal*
partnership that has expanded substantially the scope, impact, and sustainability of the mental health service activities in Liberia. –Shelly Terrazas, Carter Center

Finally, the CDC uses similar language to the United Way regarding the value of pro bono work and the increased ability of the organization to act on their mission due to project work:

C4G’s contributions to the V2V project were fundamental to the project’s success – and to CDC’s ability to help address an important gap in African health information systems. Without the pro bono technical input from Professor Vempala and several generations of C4G students, CDC would have faced a potentially insurmountable hurdle to fund a low-cost system from the ground up. – John Pitman, CDC

These examples highlight multiple areas of benefit from the partner perspective --- from enabling greater accountability (which can increase donor satisfaction and donations), to cost-savings in acquiring a much needed service for free (allowing scarce resources to be used in direct service), to better serving a target population, to adding to organization capacity, to proving a valued and even pivotal partner in support of program goals.

**Quantitative Data**

To complement these qualitative accounts, a third form of evidence of impact is quantitative data documenting direct impact. This is somewhat harder to come by and generally takes time to emerge, but we offer two specific examples. The Basic Laboratory Information System (BLIS) project, conducted with the Centers for Disease Control and regional partners, has current deployments in Cameroon, Uganda, Tanzania and Ghana, with three clinics in each country. In each case the system has been in use for at least 21 months and up to three years. In the Carter Center Mental Health Liberia project, more than 100 mental health clinicians have been trained in the use of the patient encounter form, and over 1000 patient encounter forms were submitted from June 2012 to July 2013. Data from these encounter forms has provided a picture of mental health in Liberia that rivals any other country, developed or developing, in completeness and comprehensiveness.

**Student Impact**

We close this section by noting that not only is real service impact important to partners, students cite this potential as a key motivator for taking the class. In response to a first week survey in Fall 2013 asking students to give their reason for taking the course, we heard the following:

*I'm really interested in learning how technology can be used to help people and being a part of helping people through technology.*

Another theme is that among the options for satisfying the senior capstone requirement, the service learning option holds particular appeal. As one student puts it, after citing that it meets the capstone requirement:

*The C4G projects are more meaningful to me than those offered in the regular CS 4911 sections.*

We also see students express interest in what they feel may be a limited opportunity to make this type of difference:

*My co-op right now is a pretty corporate job, it's web development for the Department of Defense. My first job out of college will also likely be a pretty corporate job. I wanted to have
the chance to see the other side of computer science and get to do something different instead of having a project where I'm just doing another corporate job.

Rader et al. have observed in a controlled research study that toy problems motivated from humanitarian contexts were less preferred by students than “fun” projects such as those derived from games. Interestingly, and in some contrast, we see students include the word “cool” in the reasons they cite for wanting to take the class, suggesting that understanding what projects are engaging is not as simple as a game/humanitarian distinction. This is typical:

*Originally I was only taking the class because I had to, but when I found out it was computing for good I thought that it would be cool to do something that can actually benefit someone.*

The class has also had impact on students beyond the course experience, specifically because of the experience with impact. A student notes:

*Your class was a refreshing reminder that our skills can be used to enact real social good, to impact the lives of others in a positive way that goes beyond letting them use their iPhone to hail a cab. Since graduation, this experience has helped me more honestly evaluate the worth of the work I do. When I mention the program to coworkers, they're often surprised - they've never considered that their skills might be more positively employed. I hope that as graduates of the class spread this mindset around the tech industry, we'll start to see a shift in the kinds of problems that startups are tackling.* – Computer Science 2011 graduate

**LESSONS LEARNED**

We bring together lessons learned and recommendations for those who want to create or adapt a service learning course offering.

**Lesson 1.** Help students to understand that while technology can play a role in tackling social problems, technology alone is never “the” answer. Indeed technology can do more harm than good, for example by diverting resources into ill-conceived projects that are never used in practice. One way to help make this point is to ask students to consider project sustainability from the outset. To do so they must understand the constraints of the partner organization for continued use and support of any technology they develop. Another method is to read and discuss an article about technological determinism and its pitfalls.

**Lesson 2.** Expose students to the partner and to the community served, wherever possible. Direct interaction with the community increases student understanding of the problem domain. Direct interaction also provides an opportunity to increase student empathy and influence longer-term commitment to community-based involvement.

**Lesson 3.** Consider having the class co-taught if the enrollment numbers justify it. Each project needs an instructor who serves as project advisor and monitors closely the progress of the team. The instructor may find it useful or necessary to attend team meetings with the partner to assist in communication and otherwise help maintain the relationship. Co-instruction also allows projects to be drawn from a larger set of personal contacts.

**Lesson 4.** Attend to partner-related ethics. Sometimes potential partners are not convincing in their commitment to the project or to the community served. We have turned down partners if they seem to be in the early or tenuous stages of connecting to a community. A weak or
inexperienced partner does not support a good student experience. In addition, we do not want to support and be connected to organizations whose commitment to communities is short-lived.

A second ethical issue can arise regarding the domain or context in which the organization works. We have accepted a small number of projects with organizations whose work might be regarded as controversial due to religious affiliation (e.g., homelessness work by a Christian organization) or subject matter (e.g., website work in support of a pregnancy prevention organization). We have handled these requests at the discretion of the instructor.

**Lesson 5.** Pursue relationships with stable long-term partners and consider cross-listing the course at the graduate level. Both of these strategies tend to result in projects that are long running and ambitious and produce student capacity to work beyond one semester.

**FUTURE WORK**

Much work remains to be done in examining the ways that service learning courses in computing achieve Connolly’s “true service” and in understanding how courses can be structured to do so. Many computing projects are intended for use over the long haul, thus longitudinal study is needed to fully understand impact. We have some data from our longer term relationships, but it is not systematic and many projects are too new for conclusions regarding sustainability.

We have proposed a set of risk categories that can be used to account for threats to true service. Similarly, we have offered a set of methods and processes for mitigating these risks. Many of our methods and processes are suitable for engineering and computing projects. As the computing and engineering communities becomes more attentive to service value, additional ideas are sure to surface that can benefit many instructors, students, project partners, and ultimate project recipients.

**ACKNOWLEDGMENT**

The author would like to thank course co-creators Michael Best and Santosh Vempala and additional faculty involved in teaching, namely Gregory Abowd, Jim Foley and Beki Grinter.

**REFERENCES**

10 Brooks, “Community connections: lessons learned developing and maintaining a computer science service-learning program.”
12 Rosmaita, “Making service learning accessible to computer scientists.”
13 Stone and Madigan, “Experiences with community-based projects for computing majors.”
14 Werner and MacLean, “Building community service projects effectively.”
17 Connolly, “Is there service in computing service learning?”
19 Connolly, “Is there service in computing service learning?”
20 George and Shams, “The challenge of including customer satisfaction into the assessment criteria of overseas service-learning projects.”
21 Brooks, “Community connections: lessons learned developing and maintaining a computer science service-learning program.”
22 Traynor and McKenna, “Service learning models connecting computer science to the community.”
23 Rosmaita, “Making service learning accessible to computer scientists.”
24 Stone and Madigan, “Experiences with community-based projects for computing majors.”
25 Werner and MacLean, “Building community service projects effectively.”
26 Rosmaita, “Making service learning accessible to computer scientists.”
27 George and Shams, “The challenge of including customer satisfaction into the assessment criteria of overseas service-learning projects.”
31 McCann, D. (2012), “A Ugandan mHealth moratorium is a good thing.” *ICTworks*. (See http://www.ictworks.org/2012/02/22/ugandan-mhealth-moratorium-good-thing/)
34 Frontline SMS. http://www.frontlinesms.com