EXPERIENCES WITH A REAL PROJECTS FOR REAL CLIENTS COURSE ON SOFTWARE ENGINEERING AT A LIBERAL ARTS INSTITUTION [*]

Vincent A. Cicirello
Richard Stockton College
101 Vera King Farris Drive
Galloway, NJ 08205
cicirelv@stockton.edu

ABSTRACT

A Real Projects for Real Clients Course (RPRCC) is a course that provides students with the opportunity to develop a solution to a real problem. Students interact directly with a real client, and work on solving a problem for that client. RPRCCs are examples of service learning. Within the computing sciences, there has been a recent surge in the integration of RPRCCs into the curriculum. They are argued to offer an increased industrial awareness, and to help retain computing majors, especially among women. In this paper, we present our experiences with a senior-level computer science course on software engineering (SE) at a liberal arts institution. We have found that recruiting so-called "real clients" from within the liberal arts college setting is surprisingly easy; and that once you have established an initial set of clients is mostly self-sustaining. We demonstrate that an RPRCC enables students to develop their skills in teamwork and applied problem solving. The RPRCC offers the SE course a framework where students gain experience applying a variety of software architectures, design patterns, and other important SE concepts; and simultaneously develop teamwork skills.

---

1. INTRODUCTION

Service Learning is defined as "a teaching and learning strategy that integrates meaningful community service with instruction and reflection to enrich the learning experience, teach civic responsibility, and strengthen communities" [14]. Service-learning courses directly integrate real-world application with the academic experience. Students gain a greater sense of the real-world relevance of their classroom coursework and activities while simultaneously providing service to the community at large.

There are a variety of ways that service learning can be integrated into the curriculum. Within the computing sciences, service-learning has been increasing through the growing movement in Real Projects for Real Clients Courses (RPRCC) [9]. In a RPRCC, students (often organized into teams) develop software systems to meet a real need of a real client. Types of clients can vary, and include non-profit organizations as well as departments within the educational institution. The student teams interact directly with their client to specify, design, and implement a solution to their client's problem.

We present our experience with a senior level RPRCC on Software Engineering at a public liberal arts college. We discuss how our institution infuses and supports service-learning initiatives throughout the curriculum. We present the structure of our course, focusing not only on the disciplinary learning objectives, but also on service-learning objectives. We provide our experience in recruiting real clients and the types of projects encountered. Additionally, we provide our experience with assessing non-technical skills such as teamwork, and in assessing learning objectives related to applied problem solving.

2. BACKGROUND

More and more instructors are integrating service-learning into computing courses as a means of developing students' non-technical skills in teamwork, communications, time management, etc [1] [3]. For example, Tan and Phillips discuss how the interaction with real clients through service-learning develops interpersonal skills, such as effective teamwork, communications, and ethical values; while also providing the opportunity to apply coursework to something real [12]. Although service-learning is most often found in upper level CS courses (e.g., [3] [5] [2]), others show RPRCCs are effective at recruiting new CS majors [7], and in retaining students, especially female students [8], when incorporated into introductory courses, including in pre-college courses [8].

Others offer words of caution on the application of service learning within computing. Connolly argues that most computing service learning projects focus too strongly on student outcomes, and too little on what the recipient of the service actually gains [6]. With the majority of software costs due to ongoing maintenance and support, a project undertaken for a non-profit has the risk of later straining that non-profit's finances. Connolly argues that we cannot ignore this, and other related issues, and must take care that our service-learning projects provide true benefit to the recipient of that service [6].

Werner and MacLean discuss difficulties encountered in service-learning computing courses related to the tight semester timeframe [15]. They observed that many non-

profits share a variety of system requirements. This, they argue, motivates design reuse via application frameworks providing functionality commonly required across non-profit clients. Their framework consists of a layered architecture with a small set of implementation languages that have offered past project success [15].

Others have also developed approaches for the challenges of delivering RPRCCs. For example, Chase, Oakes, and Ramsey consider challenges of software maintenance, issues with limiting project scope to manageable semester projects, issues with soliciting projects, providing data access to students, etc [4]. To deal with some of the challenges, they developed a support center at their institution to support past completed projects within their institution, and to act as a clearinghouse for potential new projects.

Sabin explores issues related to assessing student performance in courses where student teams develop real world projects for non-profit organizations [10]. Sabin found that multiple sources of assessment data, especially for assessing learning outcomes such as teamwork and communication, are necessary; and they specifically integrate student self-assessment, peer-assessment, and external evaluator assessment (e.g., other faculty and the project clients) with instructor assessment of project deliverables.

3. A SOFTWARE ENGINEERING RPRCC AT A LIBERAL ARTS COLLEGE

3.1. Institutional Support for Service Learning

Our institution is a medium-sized public liberal arts college. We offer degrees in computer science and information systems and are located in a school of business. Our college strongly encourages and supports service-learning across the curriculum. An Office of Service Learning facilitates the development of service-learning courses, helps find and match external non-profits to relevant courses, and provides other forms of support for students and faculty engaged in service-learning initiatives.

Students who complete a service-learning course have that service-learning experience documented on their transcript via a 0-credit hour "service learning course" for each semester where they participated in service-learning. The Office of Service Learning maintains a list of courses with either required or optional service-learning experiences to assist students who are actively seeking this type of opportunity.

3.2. Course Content and Structure

From the perspective of the academic content of the course, our software engineering course is typical. We use Ian Sommerville's textbook [11]. The course includes in depth coverage of software process models, agile software development, requirements engineering, system modeling, architectural design, implementation, software testing, and evolution; as well as topics related to software dependability.

The course is 4-credit hours and scheduled 2 days a week, with 1 day in a lab to enable coverage of software tools, such as version control systems, design tools, etc; and to provide a dedicated common time for reviewing weekly project progress with the instructor. The lab is a specialized lab for the computer science program, designed to support collaborative software development. All workstations have dual-monitors and

are organized into 4 collaborative groups of 5 workstations each. Each team's lab area is equipped with its own whiteboard and a dedicated ceiling mounted projector.

Students' grades are comprised of exams, the team project, and a service reflection essay (see Section 3.3). The exams test knowledge of the software engineering concepts. Instead of traditional assignments, teams undertake real projects that integrate all aspects of the software development process. All projects share a set of deliverables. Although we focus on agile development, we additionally want students to become familiar with traditional requirements and design documents; so one deliverable is a (simplified, scaled-down) software requirements & design document. A series of revisions is due at checkpoints throughout the semester to coincide with the topical coverage of the course (e.g., when covering architectural design, the teams add the design of their system's architecture to the document). To immediately engage them with their real clients, we begin with requirements engineering, and their first formal assignment is a requirements specification. This document is particularly useful for larger projects that may lead to projects for teams in subsequent course offerings as a means of documenting design rationale. At the midpoint of the course, teams submit prototypes, and then at the end of the semester present a demonstration to the class. Additional deliverables include a user manual to be provided to the client, and an "activity log" maintained electronically where they record minutes of team meetings, any decisions, etc. The project deliverables are assigned a single team grade. However, 10% of an individual's project grade is an assessment of that member's teamwork discussed in Section 3.5.

3.3. Integrating Service Learning Objectives

In addition to the disciplinary learning objectives, we directly integrate service-learning objectives, which include: (a) developing a clearer understanding of, and commitment to, community engagement; (b) demonstrating professionalism in interacting with members of the community; and (c) learning to apply academic learning to solving problems that benefit community needs. These service-learning objectives are assessed through an essay that each student in the class must write independent of their team members. In this essay, students reflect on how their academic learning experience prepared them to solve their particular problem of community need as well as how that service activity further enhanced their academic learning. They are asked to refer to specific examples from their experience within this reflection essay.

3.4. Recruiting Real Clients

Spring 2011 was the first offering of the course as a service-learning RPRCC. The real clients were recruited via an e-mail sent to all faculty and staff in May of 2010 to provide time for the instructor to assess feasibility for single-semester team projects. This e-mail explained the concept of an RPRCC and explained how the instructor was recruiting real clients with real software development needs. Within two hours of that e-mail, there were a dozen responses and another 10 responses trickled in over the next day or two. Of the responses, about half were of suitable complexity and scope for a one-semester team software development project. I explained at the outset to the respondents that the course would have enough students to support 4 projects to minimize

disappointment if I could not match a team to their project. Most of the clients recruited in this manner were on-campus projects in support of various administrative or academic offices. Two projects recruited in this manner were for external clients. An additional real client was an external research collaborator (an FAA research lab) of the instructor.

Teams are provided with brief summaries of the available projects, and submit a ranking of their preferences, which is used to match teams to projects. An attempt is made to give each team as close to their first choice as possible. The final set of projects included: (a) a system to geolocate and visualize historical data ("real client": a history professor); (b) a web-based system to maintain and support an inventory of natural resources (e.g., plants, animal species) ("real client": a municipality's environmental commission); (c) a system for documenting physical therapy cases ("real client": a local physical therapist); and (d) a system for extracting commands from natural language voice transcriptions of simulated air traffic control scenarios as a component to a simulated pilot ("real client": the author's research collaborators at the FAA).

In a subsequent offering in Fall 2012, no additional recruitment was needed. Sufficient projects remained from the previous recruitment effort, and included: (e) a web-based system for maintaining educational case studies developed and used by physical therapy students and faculty ("real client": physical therapy faculty); (f) a system for visualizing campus electric usage and solar generation ("real client": a college staff member); and (g) a tool for managing bibliography data ("real client": a faculty member).

3.5. Assessing Teamwork and Individual Contributions

As acknowledged by others (e.g., [10]) assessing individual contributions within team assignments is challenging. Following the recommendations of Sabin [10], we incorporate a variety of measures for assessing teamwork. 10% of the project grade is an individual grade on teamwork. A combination of student self-assessment of their own individual contribution, of their team as a whole, and peer assessments are used. The "activity logs" of the teams also document individual contributions to the team project.

To safeguard teams from "free-loading" members (non-participative teammates), teams negotiate a team contract at the beginning setting out ground rules and expectations for the team. In that contract, teams are required to specify a procedure that they use at the end of the semester to apportion their project grade among the team members based on their semester long participation. An N-member team is given N*100 points to allocate to the members constrained such that no one is allocated more than 110 and no one less than 70. A member with M points gets M% of the team project grade. The final allocation is submitted with their project, and their contract is used to settle any disputes.

3.6. Effects on Student Outcomes

Our institution uses the IDEA system [13] for course evaluation. In addition to the usual summary data, the IDEA system provides comparison data for other courses in the IDEA database across all institutions that use the system as well as discipline specific comparison data. Instructors select among a list of general learning objectives those most relevant to their course. Here we discuss the results for 3 of these learning objectives.

Objective 1: "Learning to apply course material to improve thinking, problem solving, and decisions." In Spring 2011, students self-rated progress on this objective was an average of 4.5 (on a 5-point scale), which is significantly higher than the computer science discipline wide average of 4.0. **Objective 2:** "Developing specific skills, competencies, and points of view needed by professionals in the field most closely related to this course." Students' self-rated progress on this objective is 4.2, which is higher, but not significantly, than the discipline average of 4.0. **Objective 3:** "Acquiring skills in working with others as a member of a team." Students self-rated progress is 4.6, very significantly higher than the computer science average of 3.7. Students in the course therefore recognize the development of skills in applying their knowledge to solving problems and feel they have gained teamwork skills more so as compared to the average computer science course; yet do not recognize any significant additional industrial relevant experience compared to computer science courses as a whole.

4. CONCLUSIONS

In this paper, we have presented our experiences in the design and delivery of a senior level software engineering RPRCC within a liberal arts institution. The nature of our institution has had positive effects, specifically due to its strong support for service-learning across the curriculum. Institutional support is critical in dealing with some of the challenges likely encountered with RPRCCs. It helps enable the course itself to be an integral part of the mission of the institution. Additionally, we have demonstrated that RPRCCs enable effective development of teamwork skills.

REFERENCES

[1] Abernethy, K., Treu, K., Teaching computing soft skills: an experiential approach, *Journal of Computing Sciences in Colleges*, 25, (2), 178-186, 2009.

[2] Beck, J., Fair division as a means of apportioning software engineering class projects, *SIGCSE Bulletin*, 40, (1), 68-71, 2008.

[3] Carter, L., Ideas for adding soft skills education to service learning and capstone courses for computer science students, *Proceedings of SIGCSE '11*, 517-522, 2011.

[4] Chase, J. D., Oakes, E., Ramsey, S., Using live projects without pain: the development of the small project support center at Radford University, *SIGCSE Bulletin*, 39, (1), 469-473, 2007.

[5] Chen, C., Chong, P.P., Software engineering education: A study on conducting collaborative senior project development, *J. of Sys. & Software*, 84, (3), 479-491, 2011.

[6] Connolly, R.W., Is there service in computing service learning?, *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 337-342, 2012.

[7] Egan, M.A.L., Johnson, M., Service learning in introductory computer science, *Proceedings of ITiCSE '10*, 8-12, 2012.

[8] Klappholz, D., Teaching a female friendly RPRCC (real projects for real clients course) introduction to software development at the middle school, high school or freshman college level, *Journal of Computing Sciences in Colleges*, 25, (3), 2010.

[9] Klappholz, D., Almstrum, V.L., Modesit, K., Owen, C., Johnson, A., A framework for success in real projects for real clients courses, *Software Engineering: Effective Teaching and Learning Approaches and Practices*, 157-190, IGI Global, 2009.

[10] Sabin, M., Assessing collaborative and experiential learning, *Journal of Computing Sciences in Colleges*, 25 (6), 26-33, 2010.

[11] Sommerville, I., *Software Engineering*, 9th Edition, 2011.

[12] Tan, J., Phillips J., Incorporating service learning into computer science courses, *Journal of Computing Sciences in Colleges*, 20, (4), 57-62, 2005.

[13] The IDEA Center, 2012, http://www.theideacenter.org, retrieved Nov. 21, 2012.

[14] The National Service Learning Clearinghouse, What is Service Learning?, 2012, http://www.servicelearning.org/what-service-learning, retrieved November 21, 2012.

[15] Werner, H., MacLean, L.M., Building community service projects effectively, *Journal of Computing Sciences in Colleges*, 21, (6), 76-87, 2006.