Collaborating With Industry – Strategies for an Undergraduate Software Engineering Program

Thomas J. Reichlmayr
Rochester Institute of Technology
Department of Software Engineering
134 Lomb Memorial Drive
Rochester, New York 14623 USA
585-475-2852
tjrese@rit.edu

ABSTRACT
Software engineering is prominent in the collection of undergraduate disciplines that benefit most from experiential learning. Despite an exhaustive classroom and laboratory curriculum, the attainment of program outcomes cannot be achieved without the opportunity for the student to be exposed to commercial grade software development. Even the most complex academic software engineering exercise falls short in capturing the project dynamics of building a real world software application. The Software Engineering Department at the Rochester Institute of Technology initiated the first undergraduate degree program of its kind in the United States in 1996, graduating its first class in 2001 and was among the first to receive ABET accreditation in 2002. To address the outcomes of the program and expectations of industry, the Software Engineering Department has developed a set of related strategies that provides students and faculty the opportunity to actively collaborate with the commercial software development sector. This paper will discuss how the department has positioned itself to meet the challenges of collaborating with industry by incorporating cooperative education, commercially sponsored senior capstone projects and the development of research partnerships.

Categories and Subject Descriptors

General Terms
Management

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Keywords
Software Engineering, Education, Best Practice, Industry, Experiential Learning, Capstone Projects

1. COLLABORATING THROUGH COOPERATIVE EDUCATION
As is the requirement for over 80% of the academic disciplines at the Rochester Institute of Technology (RIT), our software engineering students participate in the cooperative education program, or co-op. For more than 90 years, the hallmark of an RIT education has been the practical, paid work experience provided through cooperative education. RIT was among the first universities to begin cooperative education back in 1912. Today, our co-op program is the fourth-oldest and one of the largest in the world. Since the inception of the Software Engineering program in 1996, co-op has played a pivotal role in the professional development of our students and has underscored the importance of establishing long term relationships with the companies that provide these employment opportunities. Students are presented with the opportunity to experience a glimpse of their future career and also sample the lifestyle of a working software professional in locations across the country and at times around the world. Employers have access to a talented pool of resources and have the ability to assess the capabilities of what may be future employees upon graduation. Indeed, over 50% of RIT graduates accept full time employment with a company that they worked for as an undergraduate student.

A software engineering student in our program typically begins their co-op experience at the conclusion of the second year. Students are required to complete four blocks of co-op where a block represents a ten week academic quarter, as RIT is on the quarter system. As a result of the co-op requirement, software engineering is a five year program, with students graduating with the equivalent of almost a full year of work experience. Students entering their first co-op block have completed a full-year of introductory programming courses, an introductory software engineering class and a software engineering component design course. As students progress through the alternating pattern of work and study for the next three years, an interesting metamorphosis occurs. Students coming back from co-op employment are more motivated and better able to place course
subject content into context by drawing on their workplace experiences. For example, a course on software process and project management fundamentals becomes less abstract and more meaningful due to the student’s experience on actual software development teams – both functional and dysfunctional. From the perspective of the instructor, these students are a welcome addition to the classroom. They enrich the learning experience of their student colleagues and faculty by sharing relevant experiences from their work assignments.

2. COMMERCIAL SPONSORED SENIOR CAPSTONE PROJECTS

The delivery of a successful software product requires technical skill and adherence to appropriate process. Providing students the opportunity to apply their skills in a long-term, team project will provide a realistic industry experience, especially when the sponsor is from the commercial sector. The notion of a capstone project has been a part of RIT’s software engineering curriculum since its inception and is now commonplace in technical undergraduate programs [1, 2]. The capstone project is expected to be taken by students in the winter and spring quarters of their fifth year. At this point in the curriculum, students have completed nearly all required coursework and all of their cooperative education requirements. Students participating in the capstone projects are expected to look beyond delivering the project’s code and instead engineer a successful project that solves a problem for a real customer. From the department’s perspective, the information gathered from evaluating senior projects feeds into our assessment process and provides an invaluable qualitative viewpoint on our program that cannot be easily gleaned directly from graded course materials.

Students graduating from the program will have demonstrated abilities that are consistent with the ABET outcomes (a)-(k) of Criterion 3 [3] as well as the program specific criteria applicable to software engineering. These outcomes also reflect the vision of the faculty and our Industrial Advisory Board (IAB) with respect to undergraduate software engineering at the start of the 21st century.

The capstone project experience provides the opportunity for students to work with an external sponsor for 20 weeks. This format requires students to expand their skill set and to learn new domains through ongoing interaction with a project sponsor. Faculty advisors provide guidance, while student teams take the initiative to apply the skills acquired in their courses and through their co-op experiences. As a result, students drive their own learning experience based on scaffolding created by the capstone project sequence’s design.

The process of soliciting sponsorship, vetting proposals and executing capstone projects continues year-round. While the students do not begin their capstone projects until the start of Winter quarter, the acquisition of the projects begins the previous spring. Beginning in late spring through early fall, the department actively seeks capstone project proposals from a variety of entities including:

- Nonprofit organizations
- Local industry
- Industry from outside the area

While some capstone sponsors have typically been members of our Industrial Advisory Board, our growing student body has led to a rapid increase in the number of proposals over the last four years. As our capstone program has become more visible, we have attracted an increasing number of proposals. Companies are now routinely submitting multiple proposals. From a humble beginning of two capstone projects completed by our first graduating class in 2001, we now have 9-11 projects running on a yearly basis. During a typical capstone project cycle we will have at least one not-for-profit project, with the reminder coming from the commercial sector. For the graduating class of 2006, we received 25 project proposals of which we were only able to select nine.

The department provides sponsor information, including project proposal forms, on the departmental website [5]. Faculty have also “advertised” the capstone project opportunity with industry and academia when attending industry workshops, conferences, lectures, and similar venues. Some of the venues that have been successful for forming relationships with potential sponsors have been multidisciplinary lectures and workshops, where speakers with a limited computing background have presented problems that they are working on with limited development support to fully explore the issue or problem at hand. The growing interdisciplinary nature of computing lends itself to such relationships, where the work students deliver can have an impact.

After a faculty member makes contact with a prospective sponsor, the prospective sponsor provides a brief description of their project(s). The faculty member provides feedback, to assist with the framing of the project at an appropriate level of scope and format. The high-level requirements, including any development constraints, are provided. Potential problems or issues are also identified so that the students can see how the project fits into the sponsor’s organization. Due to the educational nature of capstone projects, prospective sponsors are informed that proposed projects should not be critical to their business or any product. In addition, potential project sponsors are informed that the submission of their project proposal does not guarantee that students will select the project.

By early fall, all potential projects are gathered by the capstone project faculty lead who shares the projects with all departmental faculty. The faculty meets to vet all of the projects, and when needed discard any projects that are not appropriate for a capstone project or have not been proposed in an acceptable format. In such a circumstance, the capstone project faculty coordinator attempts to work with the prospective sponsor to revise the proposal for the current year or for the following year. After this initial vetting process, the remaining proposals are presented to the student teams, who then rank order their project preferences. In the event of multiple teams selecting the same projects, an informal bidding process is used during which the teams submit a justification for their team being best qualified to execute the project.

During the first month of the project the student teams meet with their sponsors to develop a project and process plan [4]. Unless the sponsor already has a well defined process in place for the team to follow (which is rare), the students are responsible for
tailoring a process that best fits the type of application being developed and is sustainable for the resources and time allocated for the project. As the time and resources are fairly well fixed, most teams implement an incremental plan of staged deliveries with increasing functionality in each release. As the problem domain is typically not well known, the initial releases serve as working prototypes that aid in the understanding of the problem and elicitation of further requirements. Students draw on their course and co-op experiences in defining their process approach. Our industrial sponsors are often pleasantly surprised with the students’ grasp of process and project management fundamentals and the ability to coordinate the execution of the project.

As students continue to interact with the sponsor, the breadth and depth of documentation is ascertained. Typical artifacts include:

- Vision and Scope
- Software Requirement Specification
- Software Design Specification
- Test Plan
- Source Code, and Unit Tests
- Installation and User Guides
- Process Description
- Schedule
- Metrics Tracking Document

At the conclusion of the two-quarter capstone project teams are responsible for a final presentation to the sponsor, department faculty and student colleagues. A poster presentation and a conference style paper reflecting on their project experiences are also required. The senior capstone experience has become the jewel of our program’s curriculum. Student teams are exposed to realistic and common project characteristics often slighted in traditional software engineering courses – tailoring a project plan, working with vague and changing product requirements and accounting for installation and delivery issues required at the conclusion of the project’s term.

### 3. COLLABORATIVE RESEARCH WITH INDUSTRY

The Center for Advancing the Study of Cyberinfrastructure (CASCi) at RIT is a new, fundamentally “virtual”, research center within the Golisano College of Computing and Information Sciences. The Center is motivated by the opportunities of computing disciplines to advance modern science and engineering research, the so-called cyberinfrastructure initiative. The new center serves as an umbrella organization encompassing several independent specialty laboratories and recruits faculty across the Institute to participate in multidisciplinary efforts [6]. Within CASCi, the Laboratory for Advanced Software Engineering has been established to support the exploration and advancement of best software engineering practices through academic research and external collaboration with industry. Focus areas include software evolution, reengineering of legacy applications, distributed software development and agile software methodologies. Software Engineering faculty members of the laboratory have extensive industrial experience and are skilled in the areas of software development methodologies, requirements engineering, architecture, design, modeling, patterns, user interface design and security.

Of the above mentioned research areas, the field of software evolution and the reengineering of legacy applications is perhaps the subject of most shared interest between academic software engineering research and issues currently confronting the software development industry. Lacking in today’s undergraduate software engineering curriculum is comprehensive coverage of the areas of software maintenance and evolution. Academic exercises typically have the student developing design solutions for “green field” projects, when in reality the vast majority of industrial projects need to take into consideration the existence of legacy applications from which reusable assets must be mined. Mining involves rehabilitating parts of an old system for use in a new system. The Software Engineering Institute (SEI) [7] has developed product line strategies [8] for migrating older systems to modern software architecture. A successful technical approach we were able to apply on a project for an industrial partner was Options Analysis for Reengineering (OAR). OAR identifies potential reusable components and analyzes the changes that would be needed to rehabilitate them for reuse within a software product line or new software architecture [9].

The field of software reengineering and software product line architecture is a fertile research area for software engineering faculty and students. There is a growing challenge for industry to implement practical, cost effective solutions for migrating existing applications in order to leverage the power of emerging technologies. The combination of these forces is forming the “perfect storm” for stimulating a stronger collaborative climate between academia and industry.

### 4. CONCLUDING REMARKS

“Software has a pervasive need for people who have been well prepared to work in industry” – Steve McConnell [10]

Developing an undergraduate software engineering curriculum that provides the opportunity for students to experience the challenges and dynamics of professional software product development is a daunting task. Even the best written text books and most well designed laboratory exercises on their own cannot account for the preparation needed to become effective members of professional software development teams. This paper has outlined a strategy for immersing students in real world software projects at varying levels of their academic careers. Opportunities are made available though direct employment as part of the student’s cooperative education requirement, participation on a student team undertaking a senior capstone project with a commercial sponsor and the option to participate in the department’s research efforts with our industrial partners. At each of the same levels, the companies we are collaborating with obtain mutual benefit. Companies have access to a unique population of undergraduate software engineering students often better positioned to add value to their software projects than may be realized by students from another computing discipline. Capstone projects and research partnerships create similar opportunities for companies to execute meaningful projects and work with prospective employees. Faculty partnerships through
shared research are a vehicle for companies to explore and implement advancements in best software engineering practices.

5. REFERENCES


