Combining Real-World Internships With Software Development Courses

Cynthia J. Martincic
cmartincic@stvincent.edu
Saint Vincent College
Latrobe, PA 15650 USA

Abstract

Providing students in software development courses with opportunities to apply the course concepts within the time constraints of the traditional semester presents many, diverse well-documented challenges. Selecting a project for the students to work on is one of these challenges. Solutions to this challenge have included "toy" projects for which the instructor is the client, or "service learning" projects for which an outside organization is the client. A rather unique opportunity was presented to the students in the Computing and Information Science Department at Saint Vincent College, a small liberal arts college in southwestern Pennsylvania. An Information Technology solutions company offered to provide a small computer lab in which students could work as interns for the company on-campus and coordinate their work with existing courses. Although this collaboration presented many challenges of its own, the internships have been incorporated into the department's curriculum as course projects for the systems analysis and design, software engineering and senior capstone courses. The collaboration has been immensely beneficial in providing students with industry experience in all aspects of software development during their college years.

Keywords: software engineering course projects, systems analysis and design course projects, capstone projects, industry collaboration, student internships.

1. THE CHALLENGES IN DESIGNING SOFTWARE DEVELOPMENT COURSE PROJECTS

Courses in software engineering and systems analysis and design are requirements in many undergraduate computer science, information science and information technology programs. These courses may be combined with a capstone project, or culminate in a capstone course. These courses are typically taken after the students have had some coursework in programming and possibly in other areas such as database management. Topics and concepts that are recommended for inclusion in these courses by the ACM/IEEE Computing Curricula for Software Engineering include requirements elicitation, analysis, validation and documentation, design concepts and strategies, human computer interface design, software verification and validation, software quality, software processes and project management. In addition to these topics, the same IEEE curricula document mentioned above recommends experience with what are sometimes called "soft skills." These "soft skills" include the following (p15):

- "Work as an individual and as part of a team to develop and deliver quality software artifacts."
- "Reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, existing systems, and organizations."
- "Design appropriate solutions in one or more application domains using software engineering approaches that integrate

ethical, social, legal, and economic concerns."
- "Demonstrate an understanding and appreciation for the importance of negotiation, effective work habits, leadership, and good communication with stakeholders in a typical software development environment."

Employers often look for experience in these "soft skills" in addition to technical experience when looking for new employees (Russell et al., 2005).

Given that the number of concepts and skills covered in software development courses is quite large, designing a course project that covers all aspects is challenging (Tadayon, 2004; Scott, 2006). Often these projects are designed to simulate, as much as possible, an actual software development environment; providing experience not only with the topics and concepts in the course, but also with new software development environments and a chance to develop a number of those "soft skills."
Solutions to this challenge have included:

1. Not requiring an actual software product so that students focus on the documentation issues (e.g., Martin, 2003).
2. "Toy" projects designed by the textbook author or by the course instructor for which the instructor acts as the client.
3. Real-world projects so that students have an actual real-world client with which to communicate and for which there is an actual software need.

It has been this author's experience that students did not take the "toy" projects seriously enough to put forth substantial effort. On the other hand, incorporating a real-world project with a real client is somewhat risky, because after all, these are students and not professionals. However, this author has had success in incorporating real-world projects into courses, as have many others (e.g., Helwig, 2006; Reichlmayr, 2006; Pimentel et al., 2006; Pletch & Agajanian, 2007).

The incorporation of a real-world project into software development coursework is the primary focus of this paper. In particular, this paper describes a collaborative endeavor between the author's department and an external IT solutions provider, Prologic, Inc. (www.prologic-inc.com) which is headquartered in Manassas, VA. For the past two years, this endeavor has provided students with real-world project experience in both technical and soft skills over multiple semesters. While industry collaboration itself is not unique (e.g., Reichlmayr, 2006), this sort of opportunity is not often encountered in CS departments situated in small liberal arts colleges. The collaboration with the company has provided students experience with most if not all of the topics, concepts and skills mentioned above.

2. BEYOND PROJECT SELECTION - ADDITIONAL ISSUES IN COURSE PROJECTS

Beyond the concepts and skills noted above, there are additional challenges encountered when deciding upon a project for a software development course. These include pedagogical issues involved with team efforts, team management, and the number and type of required deliverables.

As indicated in Section 1 above, working in teams better prepares the students for the work environment in which they are likely to find themselves after graduation. Team or group projects are done in many courses and in many different disciplines, but can present some complications for the instructor of the course (Tan & Phillips, 2005). An example is the problem of individual assessment and grade assignment when all deliverables were team efforts (Hayes, Lethbridge and Port, 2003). Other challenges here include balancing the teams in terms of the skill and experience levels of the students and in terms of individual personalities.

The number and type of deliverables that accompany real-world software development projects are many and vary depending upon the actual project. They may include a requirements document, risk identification and management plan, a design document, a software prototype, system documentation, user documentation, and test plans. These must be kept to a reasonable level in order to be completed within the time constraints of an academic semester. A separate challenge in the area of documentation is getting students to take the production of project documentation seriously. Many students in this discipline would rather spend time on the actual implementation rather than on documentation. Tadayon (2004) mentions

the required project documentation was viewed as "overhead" by the students in a course while Martin (2003) reports on a course project that deliberately focused on the documentation rather than implementation of a final product.

The representatives of the company were willing to work with the faculty in determining the scope of the project, the required deliverables, the timing of the deliverables, the interviewing and hiring of potential interns for each semester and in assisting with project management issues. The only issue mentioned above that was not addressed by the company representatives was the issue of individual grade assignments for a team project.

3. INITIAL CONTACT WITH THE COMPANY

Initial contact with the company began in the summer of 2004 when two CIS majors at the college obtained internships with the company through the Career Services Office of the college. Both the students and the company would have liked to continue the internships during the following academic year. However, the closest company office was over one hour away from the college, making it difficult for students to work as interns while taking a full course load. In the spring of 2005, the company contacted the CIS Department chairperson and the dean of the school in which the department is situated with an interesting proposition. The company offered to provide a dedicated computer lab with a server and five computers for use by paid on-campus interns. The company was interested in coordinating the internship project work with coursework, such as the software engineering course, the systems analysis and design course, and the senior capstone course.

The offer of on-campus internships was seen as an intriguing opportunity for the students, the participating faculty member and the department as a whole. The college is situated about 40 miles from Pittsburgh, PA and students who were living on campus and working at internships with a number of companies were often driving for more than one hour each way to work as interns. This lessened the appeal of internships for some students and for those students without reliable transportation, made outside internships impossible. An additional enticement of the proposed internships was that the students could set their own hours, with the exception of meetings and project review sessions.

4. ISSUES AND CONCERNS WITH THE PROPOSED INTERNSHIP PROGRAM

While this opportunity did seem interesting, a number of issues needed to be resolved. These included how to incorporate the internships with coursework, and how to handle students who would continue to work as interns during semesters in which they might not be enrolled in a coinciding course. A separate concern was whether or not the students would have sufficient experience and knowledge to be successful. The project was quite complex and involved many techniques and concepts that our curriculum covers at introductory levels. This is not, by any means, a criticism of our curriculum or our department, as every effort is made to be current with ACM and IEEE curriculum recommendations. But, as in many CS departments that are situated in liberal arts colleges, it is a challenge to fit the curriculum recommendations into the credit limitations imposed by the broad liberal arts requirements. The department faculty did not want to put students into a situation in which they were unlikely to be successful, and during the initial stages of negotiations with the company, it was difficult to get a clear picture of the overall project scope.

5. THE PROJECT

The proposed project was a portion of a larger project for which the company was under contract. The company is one that provides IT solutions to a number of different government and private agencies (www.prologic-inc.com). The government sector of the company's business has included the Department of Defense and the Department of Energy. The project that the interns were going to work on was one for the Department of Defense using the JREAP (Joint Range Extension Application Protocol) and TADILJ (J-series Tactical Digital Information Link) specifications. At the time of the initial project description, these were just two of a long list of unfamiliar acronyms and terms. Two volumes of DoD requirement specification documents were provided

along with some interesting, but rather vague diagrams (Figs. 1, and 2). The first document, INTEROPERABILITY STANDARD FOR THE JOINT RANGE EXTENSION APPLICATION PROTOCOL (JREAP), or MIL-STD3011, was relatively small at 149 pages. The second document, Tactical Data Link (TDL) 16 Message Standard or MIL-STD-6016C, was quite intimidating in length. The table of contents alone was 51 pages long. After more discussion and review of the documentation, it became clear that the students would be responsible for implementing a portion of a system that would be receiving, storing and forwarding XML and bit-oriented messages over TCP/IP and UDP connections. They would not be responsible for all message types and other types of

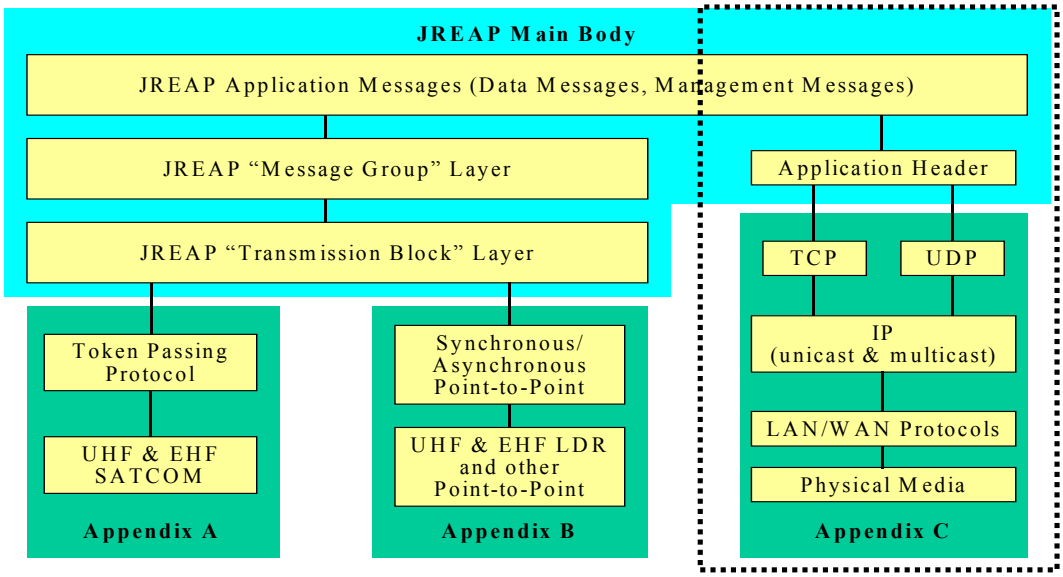
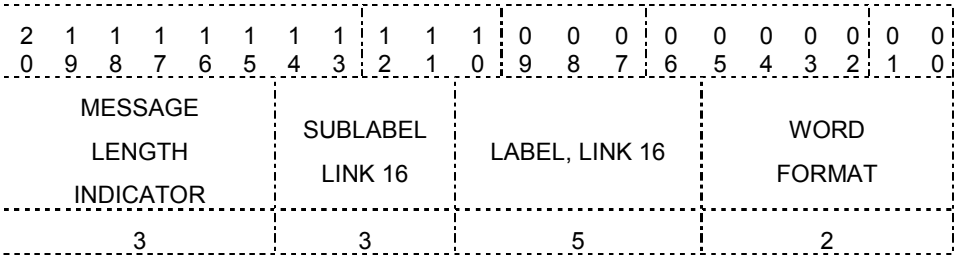


Fig. 1 Project Scope (in dotted lines)



DFI/DUI	DATA FIELD DESCRIPTOR	POSITION	BITS	RESOLUTION, CODING, ETC
550 001	WORD FORMAT	0- 1	2	00
270 004	LABEL, J-SERIES	2- 6	5	00000
271 005	SUBLABEL, J-SERIES	7- 9	3	000
800 001	MESSAGE LENGTH INDICATOR	10- 12	3	0 NO ADDITIONAL WORDS. 1-7 NUMBER OF ADDITIONAL WORDS.

Fig. 2 Example Bit Layout and Field Definitions

connections that were included in the documentation.

The messages to be implemented were of two basic types. The first type included link management messages. The second type contained information about "tracks." A track could be a stationary location or a moving object such as an airplane, a ship, a submarine, a convoy of trucks, etc.

The project was broken down into parts such that each semester's implementation would build upon the previous semester's work. However as with any large project, it was necessary to get a broad sense of the overall project scope as well as the details of the first semester's requirements.

Once the project was better understood and a formal agreement was reached between the college and the company, the actual lab had to be established. As in many institutions, space is always at a premium. A portion of a storeroom was allocated from another department. The server and computers were delivered and set up during the first week of classes of the Fall 2005 semester.

6. THE FIRST SEMESTER – FALL 2005

The scope and constraints of the project for the first semester was further refined to be the following:

System Requirements:

1. Produce JREAP C protocol communication system using TCP/IP for transport protocol. Messages to be implemented this semester were JREAP C management messages for link establishment and support. (MIL-STD-3011) These messages included:
   a. Echo messages to test connections periodically.
   b. A Common Time Reference sequence of messages to negotiate a time reference between two Processors.
   c. A Round-Trip Time Delay message to determine the round-trip delay over a link.
   d. The Latency Threshold message to specify the maximum acceptable latency parameters for transmissions that are received from the originating JRE Processor.
2. Produce a user interface for message initiation for testing purposes.
3. Produce data reduction tool to display the messages received and transmitted in a readable format.

Constraints:

1. The software was to be developed using WIN32 C/C++ using Visual Studio .NET 2003 (or later) compiler. The Microsoft .NET Managed Code (MFC) was not to be used.
2. The GUIs needed were to be implemented using WIN32 Windows only.

The first semester's project coincided with an offering of the Software Engineering course. A final concern for the inaugural semester was whether any students in the Software Engineering class would sign up for the intern project. That concern was relieved when nine of the seventeen students in the class applied for the internship project. Seven of the nine were hired as interns with an additional student hired to act as network/lab manager. The remaining students in the class were assigned to service learning projects, both of which involved creating single user databases for two departments of a nearby assisted living residence. All three teams were required to conduct periodic project review sessions and to turn in project schedules, minutes of weekly meetings and individual weekly timesheets.

As with all projects, some problems were encountered during this semester. Some were of a technical nature such as the discovery that the bit order of the message header bytes and the message body bytes were different. That is, the message header bit order was 0-1-2-3-4-5-6-7 while the body bit order was 7-6-5-4-3-2-1-0. This was discovered rather late in the semester when enough of the application had been implemented to run against a test tool.

Problems of an entirely different nature also arose. One such incident occurred when the student-elected project manager for the semester sent a very inappropriate email to the company representative after being reprimanded for not being adequately prepared for the second project review. Another incident involved a brief feud between two subgroups within the team. Both the technical issues and personnel problems were resolved and the end result was successful link establishment between the intern-produced software and the testing tool provided by the

company. A formal presentation and a run-through of the test plan concluded the semester's work. The company reported that integration of the interns' code with the code produced by the full-time company's full-time programmers took only half a day instead of the anticipated three days allotted for integration.

The benefits far outweighed the perceived risks as the students worked more diligently on this project than was the case with students in previous course offerings who had worked on "toy" projects.

7. THE SECOND SEMESTER – SPRING 2006

The second semester ran more smoothly in many regards, since the technology infrastructure and lab were already installed. There were still a daunting number of military terms and acronyms to learn, as well as more in-depth knowledge of the overall project. The second semester's offering coincided with the senior capstone project course. Four students from the prior semester did not apply for re-hire. Two additional students were hired and the student who was acting as network administrator was hired for additional hours to work as a programmer as well. The two seniors who continued working as interns for the second semester used their portion of the project as their capstone project. One of these students focused on the system architecture and lower level communications implementation, while the other acted as project manager and focused on the testing plan and procedures.

The project for the semester was the implementation of receiving and de-coding a selection of the track messages. The messages included reference tracks (geographical locations), land, air and surface tracks. When a track message was received, its bits needed to be decoded into numeric fields. Some of the numeric fields in the message were then used to look up an English specification of a specific type (such as a specific type of fighter jet) in a series of XML files provided by the company. Other fields needed to be converted into different units (e.g., feet to meters). After a message was decoded, it was to be re-transmitted as XML. Once again, at the end of the semester, a formal presentation and testing demonstration were completed successfully. Integration of code produced during this semester with the company's code reportedly took only 15 minutes.

8. SUBSEQUENT SEMESTERS – SUMMER 2006 - SPRING 2007

The original plan for the summer 2006 semester was to have one or two of the interns work full-time at the nearest company office. However, as that office was beginning a shift in priorities, and since we had the computer lab already set up, it was decided that students would continue working on campus with faculty supervision. The two seniors who had graduated were replaced by two new hires.

The project for the summer of 2006 was to establish the means to store and re-transmit messages in bit format based on a set of transmission rules. The transmit rules for each message specified how often the message should be transmitted and the ordering of the sub-messages that make up each message. The data reduction window from the first semester was to be re-formatted and updated to include the display of track messages instead of only link management messages.

The project for the fall of 2006 and the spring of 2007 included adding the ability to accept XML input and convert it to the bit-oriented message format to be stored and re-transmitted, the addition of new message types, and further improvements to the system overall. At the end of each semester, both testing and integration were successful.

9. BENEFITS OF THIS ENDEAVOR

This collaborative endeavor between the company and the college has been a tremendous success in a number of ways. First and foremost is the benefit to the students who have taken part. These students' expertise levels and their overall maturity levels grew, sometimes exponentially, as they took the concepts and skills from the classroom and realized the utility of these concepts and skills in a real-world project. More specifically, the interns have experiences that include:

- Reading and understanding complicated documentation.

- Thinking through design options prior to implementation.
- Defending their design choices and learning to accept other options as viable.
- Honing their presentation and communication skills.
- Testing and debugging code.
- Developing formal test procedures.

Each of these students matured academically and professionally far beyond the average graduating student. To be fair, the students who are selected to work as interns each semester are typically some of the best students in the department. However, we have accepted some average students, and it is these students who have made the most progress.

Every student who has been a part of the endeavor at the time of graduation has had a job lined up prior to graduation. In some cases, students have had more than one job offer. During the 2006-2007 academic year, each of the four seniors working on the project prior to the beginning of the fall semester had a job offer from the company by mid-September. Two of the students accepted the offers. The other two accepted positions elsewhere. Two seniors who joined the project in the fall of 2006 each had job offers from other companies early in the spring of 2007. This means that students who have participated in the internships have a 100% employment rate at the time of graduation. The employment rate at the time of graduation for the students who were not involved in this internship program was 58.8%. (At the time that this paper was written, the employment rate for the graduates of the last two years is 94%.) As mentioned above, many students chosen to work on this project were some of the best in the department, some of the best students opted not to take advantage of these internships and some who were hired as interns were not among the top students in the department. In fact, the average of the GPAs of the students who were involved in the internships was 3.26 and was exactly the same as the average of the students who were not involved.

Of course, there are other factors other than participation in this particular internship program that impact employment at the time of graduation. For example, the author knows of at least one student who was not in the internship program who did not start looking for a job until after graduation. In truth, there are many factors that might be at play in the employment numbers and no claim may be made as to proof that this particular internship program was the greatest determining factor in the employment of the participating students. However, it is the author's opinion that these students had learned to present themselves in a professional manner and were able to discuss their role in the design and implementation of a complex software project.

In addition to the benefits to the student interns, other students have profited as well. A company representative has presented lectures to all students on topics such as project and risk management. The documents produced by the interns and the project review presentations done by the interns became quite professional in nature as the project progressed and these served as examples to the other students in the related courses, who then stepped up their performance. The end result is that the overall quality of student projects in the software engineering, systems analysis and design and the capstone project courses has noticeably increased.

10. REFERENCES

ACM & IEEE (2004) Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering: A Volume of the Computing Curricula Series. http://sites.computer.org/ccse/SE2004Volume.pdf

Hayes, Jane Huffman, Timothy C. Lethbridge and Daniel Port (2003) "Evaluating Individual Contribution Toward Group Software Engineering Projects." Proceedings of the 25th International Conference on Software Engineering May 3- 10 2003, pp. 622-627.

Helwig, Janet (2006) "Using a 'Real' Systems Development Project to Enrich a Systems Analysis and Design Course." Information Systems Education Journal, 4(62).

Martin, Dennis S. (2003). "Project in Applied Software Engineering." Journal of Computing Sciences in Colleges pp. 22-27.

Pimentel, Bruno, Wilson P. Paula Filho, Clarindo Pádua and Fabiana T. Machado (2006). "Synergia: A Software Engineering Laboratory to Bridge the Gap between University and Industry." Proceedings of the 2006 international Summit on Software Engineering Education May 20, 2006, SSEE '06 pp. 21-24.

Pletch, Andrew and Aram Agajanian (2007). "A Software Engineering Project that Looks Like the Real World." Journal of Computing Sciences in Colleges, 22(6) pp. 92-99.

Reichlmayr, Thomas J. (2006) "Collaborating with Industry – Strategies for an Undergraduate Software Engineering Program." Proceedings of the 2006 International Summit on Software Engineering Education. May 20, 2006, SSEE '06 pp. 13-16.

Russell, Jack, Barbara Russell and Wiiliam J. Tastel (2005). "Teaching Soft Skills in a Systems Development Capstone Class." Information Systems Education Journal, 3(19).

Scott, Elsje (2006) "System Development Group Project: A Real-World Experience." Information Systems Education Journal, 4(23).

Tadayon, Nasser (2004) "Software Engineering Based on the Team Software Process with a Real World Project." Journal of Computing Sciences in Colleges 19(4) pp. 133-142.

Tan, Joo and John Phillips, (2005). "Real-World Project Management in the Academic Environment." Journal of Computing Sciences in Colleges 20(5) pp. 200-213.