Enhancing the Student Project Team Experience with Blended Learning Techniques

Thomas Reichlmayr
Rochester Institute of Technology, 134 Lomb Memorial Drive
Rochester, NY 14613, tjrese@rit.edu

Abstract - Integrating student team projects into the engineering and computing curriculum presents a variety of challenges. At the Rochester Institute of Technology, the undergraduate introductory software engineering course has been redesigned from a traditional lecture-lab format to a project-centric studio format. The student team focus has allowed us to better incorporate active learning by promoting collaboration among students and instructors in the development of a term long software project. In order to make the studio format more effective we have adopted a blended learning approach. Blended learning aims to join the best of face-to-face classroom learning with the best of online teaching and learning to promote active independent learning and reduce class seat time. Student satisfaction in courses requiring a collaborative effort among their peers is heavily influenced by their project team experience. The use of blended learning techniques helps to make the experience a more satisfying one by increasing the effectiveness of collaboration in team activities and interaction with the instructor through the application of distance learning and social computing technologies. The challenges faced and the techniques and strategies utilized in the planning and delivery of the course will be discussed, including the utilization of online learning support infrastructure. This paper presents instructor experiences, analysis of student feedback, lessons learned and recommendations for other educators considering the application of blended learning techniques for their courses.

Index Terms - Blended Learning, Student Teams, Active Learning, Collaborative Learning, Software Engineering.

INTRODUCTION

Programs incorporating student team projects into their curriculum introduce a number of challenges for both students and instructors. At the Rochester Institute of Technology, the Software Engineering Department has made student team projects a central focus of its curriculum. Accompanying this commitment to team projects is a responsibility on the design of the course to create an infrastructure and climate that promotes healthy team activity and appropriate student interaction. For student team learning to be an effective strategy in the classroom, one issue to be addressed is that of empowerment. Teams must be given the means and opportunity to achieve success. [1] Teams or groups that are casually formed with little preparation or instruction have a negative effect on the student’s view towards teamwork and could adversely affect their interaction with peers in subsequent courses or professional career activities. This paper will report on the experiences of an introductory software engineering class with a strong student team project component that has incorporated active and blended learning techniques into the course design. An overview of blended learning in general will be presented, followed by a summary of the strategies and tactics used in the redesign of our introductory software engineering course to create an environment that supported an effective student team project experience for the student.

BLENDED LEARNING OVERVIEW

Higher education is increasingly turning to blended learning as an option for teaching and taking courses. A report by the Sloan Consortium for 2002-03 [2] showed that:

- Eighty-one percent of all institutions of higher education offer at least one fully online or blended course.
- Among public institutions, 97 percent are offering at least one online or blended course and 49 percent offering an online degree program.
- When asked about the role of online education for the future of their institution, 67 percent answered that it is a critical long-term strategy for their institution.

A concise definition for blended learning is elusive. Simply stated blended learning aims to combine the best of classroom learning and teaching with the best of online teaching and learning. However, complexities in the definition arise from the virtually limitless course design possibilities and the difficulty in distinguishing blended learning from other forms of learning that simply incorporate online opportunities. The challenge of blended learning is the effective integration of face-to-face instruction and the complementary application of online technologies [3].

The list of blended learning techniques will look familiar to course designers that have developed online courses and have supplemented traditional course content with online resources:
Use of course management software systems
Placing reference materials online
Use of e-mail
Synchronous instant messaging and chat rooms
Asynchronous threaded discussion groups
Collaborative student team or group activities
Online testing
Web-based video conferencing
Online surveys and assessments

What distinguishes a blended course from a traditional course that merely makes use of the above listed techniques, is that the blended learning approach requires the course designer to rethink fundamental teaching and learning dynamics relative to the discipline, development level, and resources characteristics of the course being developed. The mix of available techniques and their application to desired course outcomes create a diverse set of potential solutions that insures no two blended learning designs will be identical. [3]

Designing or redesigning a course with blended learning techniques can follow a variety of models. One proposed approach has identified five blended learning models (supplemental, replacement,emporium, fully online and buffet) and categorized them on a continuum from fully face-to-face to fully online interaction with students.[7] Another approach focuses on the learning strategies and identifies dimensions – combinations of blended resources, collaboration, structure and pace to get the right content in the right format to the right people at the right time while reinforcing the concept that learning is not just a one-time event, but a continuous process. [8]

The move towards a blended learning approach has shown the potential for a significant list of improved learning benefits: [2]

- Increased access to a range of appropriate, individualized learning and teaching resources.
- Accommodation for learners and teachers of diverse ages, styles and cultures, who can engage in learning activities from remote settings such as workplaces, homes, countries and more.
- Flexibility and cost effectiveness in terms of scalability, breadth, time, value and infrastructure.
- Greater student and faculty satisfaction.

In the 2003-04 academic year, the Online Learning Department at the Rochester Institute of Technology sponsored the Blended Learning Pilot Program providing faculty with an opportunity to modify an existing course to use blended learning techniques. This program created an instructional model that defined a blended course as any course in which 25% to 50% of classroom lectures and other seat time are replaced by instructor-guided online activities, such as online quizzes, virtual team projects, synchronous chat sessions, and asynchronous discussions [7]. The following section discusses the experiences of participating in the pilot program and redesigning an introductory software engineering course to support course and program outcomes, specifically in the area of student collaboration and team project activities.

Session T4F

COURSE BACKGROUND

Student team projects are a hallmark of the undergraduate Software Engineering program at the Rochester Institute of Technology. The ability for a team to effectively work together towards a common objective is paramount to the success of a professional software product. Other professional fields have similar needs and expectations of the personnel they hire. A survey conducted by Hope College among 58 employers from a variety of business domains revealed that they placed team-building skills at the top of a list of 23 characteristics desired of entry-level employees.[8]

Each course offered by the SE program incorporates at least one project of significant size assigned to a team of 4-6 students. A stated outcome of RIT’s software engineering program is that “software engineering graduates are able to work in small teams to develop a software system and demonstrate the ability to assume team roles in addition to design and implementation.” Other program outcomes also rely on the development of skills (communication, presentation, and writing) that contribute to them being an effective team member on a software development project.

Students are introduced early on to working in teams. The first opportunity to experience the team environment is in the introductory software engineering course, Software Engineering I (SE361). This is the initial course offering in our department’s program. The typical student will take SE361 in the second quarter of their sophomore year. Prior to that they will have completed a sequence of introductory computer science (CS) courses. The CS courses focus on developing programming skills (using Java and C++) and address traditional CS topics – data structures, algorithms, object-oriented concepts and some exposure to concurrent and network programming techniques. Students individually complete lab and project assignments with little opportunity, or need, for collaboration with fellow classmates. The CS course sequence introduces some software engineering concepts, but at a cursory level. For example, students are able to identify the major activities associated with the waterfall life cycle of development. Students arrive to SE361 with accomplished individual programming skills and fine-grained problem solving techniques.

The primary objectives of our introductory SE course are to give students the opportunity to experience teamwork while working on a term-long project that spans a full development lifecycle, applying effective software engineering practices, and following a process discipline and also to expose students to fundamental concepts of software design and process such as lifecycle models, design patterns, testing and estimation.
The following summary of the recent history and evolution of the course highlights the strategies taken to move student involvement from individual and passive to collaborative and engaged using an active learning approach supported by blended learning techniques.

The initial format of SE361 was that of the traditional technology or science course with a laboratory component. Students would meet in lecture three hours a week and also attend a two hour lab session during which time student teams would work on their project activities. Since its inception, the course has always maintained a significant online presence. As is the format for the typical technical course, instructors would post materials and project assignments to a course website. Likewise, students would have a mechanism for posting project assignments and interacting with the instructor and other team members. The interactions were usually limited to the submission and return of project assignments and standard email communications.

An issue that arose from the traditional lecture/lab format was that students at times had difficulty in relating subject matter in lecture to what was required of them for project deliverables. Lectures consumed a significant amount of time during the week covering core material important to learning outcomes, but students perceived a disconnect between what they were passively being presented and what they were applying in their project teams. Students tended to learn the material for the sake of exams, rather than relating it to their project work and as being fundamental to understanding software engineering.

The project centric nature of the course requires the individualistic thinking student to quickly adapt to the challenges and rewards of working in a cooperative team environment. Project teams are formed during the first week of class with students identifying skills sets and team role preferences in an effort to achieve some element of balance among the teams. Since the scope of work required by the project cannot be completed by the team during scheduled class time, one of the first project activities is for teams to identify common meeting times outside of class. A typical experience is for a team to identify only one or two hours during the entire week when they are able to meet. Additionally the times identified were normally odd hours in the very late evening or early morning. Early morning times were almost unanimously vetoed by all of our students, leaving limited hours late at night for teams to meet and work. Further complicating the out of class meeting times were work for many students and family and personal commitments for older students.

To address these challenges the course underwent a major redesign with the following goals:

- Create a stronger relationship between lecture topics and project activities within the course schedule.

A key decision made during the redesign process was to change the format of the course from the three hour lecture/two hour lab format to a studio format that met twice a week for a two hour session in the lab. The strategy was to change the role of lectures from merely covering material to helping students integrate the material and provide perspective. The students were expected to prepare for class by reading relevant sections from the textbook and online sources. Lectures and in-class activities completed and consolidated this learning process, rather than being the primary source of knowledge.

Time was allocated over a two hour period for instructor mini-lectures, class activities and project work, with the objective of allowing the majority of the time for student team project activities. This approach also addressed the issue of students being scheduled for different lecture and lab sessions, with potentially different instructors. All teams remained together for the entire project with same instructors. The studio format also reduced the amount of seat time in the classroom by one hour per week (20%) with the expectation that additional time would be available for individual student driven learning activities and student team project work.

In order to assess the new course format, the students were surveyed at both the course midpoint and at the end of the quarter. The survey was administered online to students enrolled in the course during the Fall, Winter, and Spring quarters. The survey asked the students to supply demographic information, but concentrated on gauging the effectiveness of the course structure, project, course elements and student preparation. The students also had the opportunity to comment on aspects of the course that they valued and aspects of the course that should be revised.

During the course many students commented informally that the project was very important to them and that they wanted more course time devoted to the project. The survey results supported the students’ comments. In terms of lecture, 47% of students were content with the amount of lecture in the course while at the same time 47% of students wanted less lecture. In terms of the effectiveness of lecture, 31% viewed lecture as being either Helpful or Extremely Helpful while 34% found lecture to be Moderately Helpful.

Instructor observations correlated with the student responses. Students appeared to be more motivated and engaged while participating in project activities. The issues regarding the classroom, or lecture side of the course had been improved by incorporating activities both in groups during class and online individually, but course evaluations revealed that students were still expressing a concern for the amount of time allotted to the team project.
ACTIVE & BLENDED LEARNING: PHASE 2

The move to the studio lab format had already decreased the amount of in-class face time by one hour per week. This move was supported by moving a significant amount of traditional lecture time into online resources in the form of self-study and online discussion activities. The blended addition for this phase of the redesign focused on the team project aspect of the project by applying active learning strategies with goals similar to those for Phase 1:

- Make more effective use of online technologies when teams are not in class. This included instructor interaction with the teams and communication between team members.
- Address the issue of students not having enough time for projects during class time by reducing face to face lecture time using blended learning techniques.

A primary example of taking advantage of existing technology for more effective communication was the use of a chat application that is embedded in our course management system. Students are very comfortable with the use of chat applications, such as America Online Instant Messenger, and use it informally to communicate during team projects. Our plan was to build on the familiarity and place some formal conventions around the use of the tool.

During the project instructors take the role of surrogates for the customer. As our project statement of needs is fairly open-ended, teams must interact with the customer to elicit requirements for the product. In our pre-blended SE361 sessions this would require the instructor, using class project time, to meet with each team. The blended approach in this session was to establish scheduled online chats with the “customer”. This encouraged teams to be more focused and prepared when meeting with the customer, as well as helping the instructor manage the interactions with four and sometimes eight different project teams. The time normally consumed by customer elicitation activity was now available for other project development activities and opportunity for additional instructor interaction on other facets of the project.

In addition to freeing up more class time, this approach addressed key software engineering learning outcomes related to the activity of requirements engineering. Although students are quite adept at managing multiple chat sessions in casual conversations, professional communication with a business associate or customer requires an added level of formality and focus. The chat session required teams to prepare ahead of time for the interaction with the customer. Several lines of questioning needed to be prepared to account for potential customer responses. This approach contrasted to in-class requirements questioning in that the teams did not have the luxury of the instructor/customer being readily available at anytime in person. Teams were made aware that the customer would only be available for fixed time periods for the chats (20-30 minutes) and that customer would never be available during class, meaning the instructor was not required to maintain dual personalities in person.

Sample Chat A is a portion of an actual chat held early on by a student development team and the customer. The product being created is an educational software game for grammar school students in the 1st through 4th grades. The customer role being played by the instructor is a representative from the grammar school system where the product will be deployed.

**Sample Chat A:**

_**Team:** Good Morning  
**Customer:** Hello  
**Team:** Thank you for your time this morning. We just have a few questions to ask you regarding the spelling game we’re designing for you  
**Customer:** Are we waiting for additional team members?  
**Team:** All of the team members that will be here are. We are all sitting in a meeting room with a projector. I am in a sense, the scribe.

An interesting note here is that one of the objectives of using online chats as a blended learning technique was to address the issue of students not being able to physically meet outside of class. In this case the entire student team convened in a team project room to take part in the discussion! This observation was discussed during class and students were encouraged to try using a truly distributed chat session from whatever location was most convenient for them.

**Sample Chat A (continued):**

_**Team:** Should a student be associated with a class, or by grade?  
**Customer:** If by class, you mean associating teachers with a set of students  
**Team:** Will any one teacher be administering the game to different classes of the same grade level?  
**Team:** We’re concerned that one teacher may have multiple classes of students, and we had figured that it would make sense for the teacher to be able to administer them easily.  
**Customer:** No, each teacher has only one class. No teachers teach across grade levels.

**Team:** OK, for our fill in the blanks game, we were going to have it display the correct answer in jumbled form. We wanted to make sure a jumbled word game is not too complicated for the students.

**Customer:** Like in the newspaper?  
**Team:** Yes  
**Customer:** That’s a good idea, as long as the words are not too hard for a grade level.

From this sampling of interaction between the team and instructor one can observe that the student entered questions and responses are at the appropriate level of formality for a professional setting and do not slip into common chat slang. The most difficult part of most software development efforts is getting a clear understanding and agreement on what the product requirements are. Getting the opportunity to think first...
and enter questions in written form, as opposed to informally asking verbal questions in an unstructured format helps that process to be more effective. The second hardest part of the development effort is in documenting the conversations with the customer and creating a written software requirements specification. Each chat automatically created a recorded log of the questions and answers from the elicitation session.

The next chat sample was taken during the second round of requirements elicitation. Note that there are now multiple team members participating in the chat session as opposed to one team voice. This introduces a new dynamic for the team in that at times different team members can alter the theme of the questioning. Some chat applications have a “whisper” feature, it allows chat participants to talk directly to another member in the chat room without others hearing. As time went by, several teams discovered this feature and began to use it effectively.

Sample Chat B

Team Member B: OK, let's start this off with some questions: Are the computers in the school networked to send data to each other?
Team Member A: and will there be a machine that's always on that we can use to run a dedicated server process?
Customer: We are networked, but this application will only run on single machines
Customer: What's a dedicated server process do?
Team Member A: it would be always running on one machine, and collect data from the other machines running our software so that a teacher could follow their students' progress

From the instructor’s perspective, here is where the chat interface allows them to more easily assume the role of the customer. In this case the term “dedicated server process” would be foreign to most casual computer users

Sample Chat B (continued):

Team Member D: We know teachers can add words; do the teachers want to add one word at a time or a entire 'word list' at a time?
Customer: One word is fine, but a word list sounds interesting. Would they type in a list or be able to reference it from some other document?
Team Member D: Most likely some other document.
Customer: Is this what "importing" means?
Team Member C: Yes, the teacher could import a list of words that the spelling games would use.
Customer: What kind of documents could we use? Do they have to be in some special format?
Team Member A: The document would have to be a .txt file
Team Member C: in other words, a text document with a word on every line.
Customer: Oh, That's a good idea, but let's assign it a lower priority than one word at a time for now.

Note in this next section the fact that Team Member C recognizes that Team Member A has made a jargon reference ("*.txt file") and jumped in to correct the question prior to the customer responding.

Each team participated in 2-3 customer chat sessions throughout the quarter. As noted, for the first chat some teams physically met in the lab and huddled around one terminal while the chat was taking place. During future chats teams found it more convenient to use the power of the virtual meeting to participate from remote locations. The chats were also quite valuable to the instructor in being able use the logs to document conversations with student teams and maintaining consistency in the responses. Prior sessions were often a collection of undocumented, informal responses as questions were posed by the project teams in an ad-hoc manner. It also allowed the opportunity to vary the requirements between project teams.

Conclusions

In addition to our regular course survey, student data was also collected from the Online Learning Department as part of the Blended Learning Pilot Program. In the quarter that the blended SE361 course was being given, 26 courses across RIT were also participating in the program. Feedback from the survey indicated that 51% of the students felt the blended format increased their interaction with other students (19% decreased, 30% no difference) and 72% of the students liked having part of the course online and part of it in the classroom (17% disagree, 11% neutral).

From our own observations we found the blended techniques we used to be effective from the perspective of reducing seat time in favor of student centered activities which in our case was more time allotted to student team projects. The usage of chat rooms for requirements elicitation proved effective in decreasing the time required for the instructor to circulate among team groups during class time and as an effective teaching software engineering teaching tool.

We are continuing to explore additional techniques and tools that will enhance the student team and group learning experience. Future plans include the incorporation of software development and team communication tools that support team skills and effective collaboration between students and instructors.

Acknowledgment

The author would like to acknowledge the support of the Online Learning Department at the Rochester Institute of Technology (sponsors of the Blended Learning Pilot Program) for their assistance in the design and evaluation of our blended introductory software engineering course.
REFERENCES


