Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 07-019

Cooperative, Project-Based Learning Reinforced by Reflection and Presentation in a Graduate-Level Software Engineering Course

Tamara Moore, Anne Kern, and John Collins

September 14, 2007

Cooperative, Project-Based Learning Reinforced by Reflection and Presentation in a Graduate-Level Software Engineering Course

Tamara J. Moore, Anne L. Kern, and John E. Collins[1]
University of Minnesota

Abstract
This project studied the effectiveness of a comprehensive learning project that incorporates cooperative learning, project-based learning, and reflective practices in a graduate software engineering course. This paper will report on the instructor's objectives for the project, the students' achievement, and the team reflections of their progress through the project. The results of this study (1) show the effectiveness of project-based learning in a software engineering course, (2) serve as a model for software engineering instructors to use for the creation and implementation of similar course projects, and (3) show the impact of reflective practices in team settings on student learning.

Introduction

In their seminal work, *Pedagogies of Engagement* [1], Smith, Sheppard, Johnson, and Johnson provide details of using both cooperative learning and project-based learning in engineering education. They discuss these best practices in detail with the purpose of encouraging other engineering educators to use these teaching methods. At the University of Minnesota, a master's level software design course has implemented both cooperative learning and project-based learning. For this paper, researchers have studied the effectiveness of a comprehensive learning project that incorporates elements of cooperative learning, project-based learning, and reflective practices. There are several key elements in the design of the cumulative course project, including:

- Student teams chose the projects and negotiate with the instructor to refine project scope,
- Teams established and provided support for quality criteria for the project,
- Students participated in permanent project teams, which also completed in-class assignments,
- Teams presented their work to the class at each phase of the cumulative project, detailing project progress and reflecting on design process and team interaction,

- Students participated in team and individual self-evaluation based on both quality of project deliverables and peer reflection and input at each phase of the project,
- Instructor provided detailed feedback on both the process and content of project design.

This paper will report on the instructor's purpose and objectives for the project, the students' achievement, and the team reflections of their process and progress through the project. Data collected for this study include an instructor interview, course syllabus and objectives, student artifacts from the project (recorded student presentations, collected student project work, and written reflections), and student project grades. The results of this study (1) show the effectiveness of project-based learning in a software engineering course, (2) serve as a model for other software engineering instructors, as well as other engineering educators, to use for the creation and implementation of similar course projects, and (3) show the impact of reflective practices in team settings on student learning.

Rationale

Cooperative learning (CL) had been demonstrated to be an effective instructional strategy in a variety of disciplines, grade levels, and social domains [2-3]. Numerous studies suggest that when correctly implemented, cooperative learning improves information acquisition and retention, higher-level thinking skills, interpersonal and communication skills, and self-confidence [4]. In addition to promoting learning, cooperative learning provides a means to simulate the working environment in which students will be expected to engage. Engineers working in industry often must rely on the expertise of team members to complete tasks assigned to them. Being able to effectively work in teams is not a skill that most people automatically possess. Therefore, it is necessary to put students in situations where it is essential to work in teams to allow them to develop teaming skills [5]. For future engineers, it is critical that they have an opportunity to engage in collaborative projects that mimic future work environments.

The five essential elements of cooperative learning are positive interdependence, individual accountability, group processing, social skills and face-to-face interaction [6-7]. Without these elements established in the learning process, cooperative learning is not the same as group work. The instructor's role in cooperative learning is to guide and support the interactions between the students using long-term formal teams and informal learning groups. This is a deeply relationship-based practice that encourages all students to feel a part of the classroom community and promotes learning success for all students.

Reflection is the complex process of carefully thinking about one's experiences, and can be a valuable resource and learning tool for engineering students. John Dewey [8] argued that learning requires active reflection upon one's previous experiences and that such reflection is an essential ingredient in all thinking. Researchers suggest that the process of reflection allows for repetition of a particular experience in such a way that students internalize the experience cognitively, and that reflection helps people learn from experience and transfer insights to future applications [9-10].

Project-based learning (PBL), similar but not exactly the same as problem-based learning, has been described as using projects as a vehicle for "reflective practice and inquiry," to "reveal deeper aspiration," and to "construct shared understanding" [11]. In the engineering education environment, Prince [3] defines PBL as the "instructional method where relevant problems [projects] are introduced at the beginning of the instruction cycle and used to provide the context and motivation for the learning that follows" (p.1). Both Prince [3] and Ayes and Zeniuk [11] suggest that PBL enhances self-directed learning and reflective practices on the part of the learner.

Methodology

A Naturalistic case study method was used to determine the intent of the project and the level of success and collaboration the three student project teams experienced [12-14]. The purpose of the case study research was to explore in depth the team members' ability, success, and group processing on a software engineering design project, both from the perspective of the team members and the course instructor. Multiple sources of data include: (a) in-depth semi-structured interview with the course instructor; (b) complete documentation of teams project phases, five phase reports; (c) phase reflections from each team documents; (d) peer evaluations from team members taken multiple times during the project, and (e) class observations including team project presentations and team work activities. In an effort to establish rigor and credibility for the study, parameters such as triangulation of data source and multiple researcher analysis was employed as described by Darke, et al [13].

Setting

The course discussed in this study is the Spring 2007 section of SEng 5802 - Software Engineering II. This course is taken in the second of the four semester Master of Science in Software Engineering (MSSE) program at the University of Minnesota. The overall goal of MSSE is to raise the level of professional practice in software development. This includes all phases of software product and system life, from initial concept, through design and test, into deployment. The development of complex software systems requires a subtle blend of art and science. The science is the subject of a Computer Science curriculum. Much of the MSSE program focuses on the art and craft of software engineering - the people and organizational issues.

Virtually all students in the MSSE program are working full-time in the software industry. Most have at least three years of professional experience, and they typically view the program as a way to advance their careers. They come to campus one day every week for classes, but their time outside of class is severely constrained, and their homes and workplaces are scattered across a relatively large geographic area. The result is that their ability to meet together outside of class is quite limited.

SEng 5802 is a software design course, which covers just one aspect of the program as a whole. Software design can be thought of as a mathematical art form -- there are typically many correct ways to solve a design problem (and a much greater number of incorrect ways, of course). They tend to differ along a number of dimensions that we characterize as "quality criteria" such as maintainability, extensibility, efficiency, robustness, etc. A good designer starts by understanding both the problem to be solved and the important quality criteria as dictated by users and the business situation in which the work will be used. A good design correctly solves the problem and effectively meets the identified quality criteria.

The Project

The project that will be focused on in this study was a semester long, five phase software design. The students were assigned to teams of four students (except when the class was not evenly divisible by four, then there could be a few teams of three). The three teams that we studied were all teams of four. The overall project task as stated in the syllabus for the course was:

> TASK:
> Your job will be to develop and document a design for a non-trivial project of your own choosing. Here are some possibilities:
>
> - Start with the specification work you did in SEng 5801 - Software Engineering I.
> - Choose a project from your work experience that needs design or redesign (assuming there are no confidentiality issues, since you will need to share your design with the class).
> - Pick an open-source project, understand it, extend it.
> - Work on an idea you could start a business with.
> - Work on a system that could support your own business or your spouse's or friend's business.
>
> We will try to take an agile approach to this project, within the constraints of the class situation. That means that your first task will be to prioritize your requirements and scope a first iteration. Clearly, many interesting projects would involve design efforts

beyond the scope of this class. But that just means you need to scope a first iteration that IS of a reasonable size.

The project was broken into five phases: 1) Project Proposal, 2) Scoping and Planning, 3) Preliminary Domain Model, 4) Complete Design, and 5) Refactoring. Each of the phases had specific objectives and was worth a percentage of the overall grade of the project. Each team gave an in-class presentation of their current project after phase 2, phase 3, and phase 4. The phase 2 presentations were five minute introductions to the project, and due to the lack of detail of the projects in phase 2 presentations, this paper will focus only on the phase 3 and phase 4 presentations. The following paragraphs will give detail about each phase of the project and its percentage of the overall grade. In the discussion of the project, the authors and the instructor use the words "project" and "design" interchangeably.

Phase 1 - Project Proposal

The goal of the project proposal was to build community among the teams by having them choose and negotiate a project. The teams were required to submit a written proposal that described the problem, summarized the requirements of the project, and gave justification for the teams' belief on why it was an interesting problem. While the instructor of the course originally created the teams, they had a chance to redefine their team-membership (with the help of the course instructor), if they were finding they needed different expertise, or had personality conflicts. The project proposal was worth 10% of the total project grade.

Phase 2 - Scoping and Planning

Phase 2 of the project asked teams to articulate the requirements, risks, and quality criteria for the project, as well as submit an overall plan for design and construction of the entire project. The teams were to provide a high-level analysis of the project requirements in a manner that identified and justified the teams' assessment of project risks, three top quality criteria for design and implementation, description of the approximate scope of the first three development iterations of the project including assumptions made by the team, description of the areas of anticipated change in the requirements of the project, and a process narrative. The process narrative described the steps the team went through to arrive at the current stage of the project. The teams were asked to focus on the parts of the phase that were easier or more difficult than what they expected and to reflect on why they thought that was the case. The process narrative was important to this study in that this data was used by the researchers to study team change and process throughout the project. This phase was worth 15% of the overall project grade.

Phase 3 - Preliminary Domain Model

In phase 3 of the project, the teams developed clear understandings of their problem in the form of a domain model as described by Evans (2004). Here the student teams were to focus on the scope of their first iteration, providing detail including a glossary of terms to describe the domain and design, a Unified Modeling Language (UML) model with both static and dynamic elements with explanation, a written narrative description of the design to aid in the understanding of the team diagrams, a self-assessment of the design at this point listing strengths and weaknesses of the design and well-known patterns used in the design. Again, the teams were required to include a process narrative that recorded the team struggles and successes of the model development for this phase. This narrative also included a step-by-step process for arriving at the current model, had the team members break down their individual contributions to this phase by time, and discuss how the individual effort differed from their original expectations. Phase 3 was worth 25% of the overall project grade.

Phase 4 - Complete Design

The goal of phase 4 was for the teams to develop, document, and evaluate a design for the first iteration of the system as identified in phase 2. The design needed to be ready for implementation and testing of the first iteration. This phase was primarily a refinement and extension of their domain model with the addition of significant elements in the infrastructure and application layers. The specific requirements of this phase include updates to the glossary of terms, design artifacts in the form of UML diagrams to express the teams' design ideas, narrative to guide the audience through the team design, a brief quality assessment to describe the strengths and weaknesses of the team design with respect to the quality criteria set up by the team and the criteria discussed in class, and a process narrative. The basic tenets of the process narrative are the same in this phase as in the previous two phases, except here the teams were to also include a discussion of the significant changes in this iteration versus the phase 3 model and why the team decided to make those changes. The complete design phase was 30% of the final project grade.

Phase 5 - Refactoring (20%)

The purpose of phase 5 was to improve their design as represented in the previous phase by refactoring and applying patterns including a description of how the refactored design is an improvement over the original design. For this phase, the deliverables were the revisions of phase 4 with at least two distinct refactorings demonstrated. For each of the refactorings, the teams were to identify the problem solved by the change, identify the pattern(s) applied and how the pattern roles map to the team design, include a sequence or collaboration diagram showing how the refactored design worked, and a brief description of how the team design quality was improved by the refactoring. Phase 5 was worth 20% of the final project grade.

Instructor Purpose
An interview with the professor of SEng 5802 was conducted at the conclusion of the course. The purpose of this interview was to gain insight into the instructor thinking and reasoning behind creating and implementing a software design project of this magnitude. There were four main topic areas of the interview: goals and objectives of the project, creation of the project, project evaluation, and perceived value of cooperative learning/teaming.

Goals and Objectives of the Course Project
The instructor of the course began the interview by talking about his reasoning behind the goals and objectives of the project assignment. He described four primary goals he had that were related to the project. His goals were to i) develop fluency in practice of software design, ii) teach the communication of design ideas, iii) provide students with the opportunity to negotiate with team members to achieve a design product that represented the best thinking of the team as a whole, and iv) teach students to evaluate designs according to a set of quality criteria.

The instructor developed the course with the intention to develop skill and fluency in software design, and an ability to practice that skill effectively as a member of a team. The project required students to (1) collectively understand a problem that was new to most of them, to (2) develop and critique design approaches to solving that problem, to (3) agree among themselves on a specific approach as a team, to (4) elaborate their ideas to a level of detail where a path to implementation was clear, to (5) communicate their design ideas in written and graphical form, to (6) evaluate their own design work in light of a set of identified quality criteria, to (7) explain and defend their ideas in oral presentations to peers, and to (8) reflect on the process of developing their designs in a series of process narratives. The instructor stated, "The project is designed to develop and exercise critical professional skills in a situation that allows for evaluation, feedback, and improvement without putting jobs and critical projects at risk."

Creation of the Project
The instructor discussed how he conceived of the course project by drawing on his experience in industry as a software engineer. In order to tell the narrative of his conception of the project, he began by talking about the course design. He designed the course to cover design process, evaluation and analysis of design, and a number of specific design techniques. He felt that in courses that cover this sort of material, it is very difficult for students to apply what they learn, since many of the important issues are very hard to demonstrate on problems that can be worked in class or by individual students outside of class. He wanted to make the course an

effective learning experience; therefore, he created the semester-long project that was scoped to about 150 hours of effort. This amounted to almost 1/3 of the total time investment for a 3-credit semester course with project teams of four students.

He indicated that the project in its current form evolved over a period of five years. Originally, all teams worked on the same problem, which varied from year to year. He found three major problems with this approach. First, it was very hard to evaluate team submissions, because the instructor was faced with 8-10 designs for the same problem, using pretty much the same terminology. After going through several of them, the instructor felt that he lost the ability to differentiate between project submissions. Second, he indicated that class discussions and oral presentations became "quite tedious" after several teams had presented their work, and they failed to generate discussion. Third, he found it very difficult to find projects that effectively motivated students who came from a wide variety of professional backgrounds. Therefore, in recent years, he has asked each team to propose their own projects and identify the specific quality criteria that were most critical for the chosen projects. In order to control the scope of the project, the instructor stated that some negotiation was required between individual teams and himself, since early in the semester students tended to be overly enthusiastic and badly underestimated the expected effort. He provided an example of this, "One team wanted to build a system for collecting data on clinical trials. The goal was to allow clinicians to design the questionnaires and data formats without programmer intervention, producing databases that can be readily analyzed. This is a type of problem that has been well known and unsolved for years. After a few rounds of negotiation, they agreed to focus on just a stripped-down tool for designing questionnaires, which they were able to design to an adequate level of detail in about 200 hours." Because students chose their own projects, the instructor felt that motivation tended to be higher, presentations became interesting to the other students, and it was much easier to evaluate the work and give effective feedback.

Evaluation of the Project

When asked about the evaluation of the project, the instructor indicated that the project involved a series of five deliverables, starting with the proposal. Since the deliverables were not independent - each one built on the previous work - this meant that each phase required both correction of problems found in the previous phase as well as potential problems with the new work. Because the students were going to be professionals in software engineering, the instructor wanted the general criterion for evaluation to be professional-quality work. He indicated that submitted material was to clearly address both assignment and project requirements. He required that the team presentations be clear and well organized. He stated that the design details correctly addressed identified requirements, and teams needed to choose the design details appropriately according the specific quality criteria identified by the team for the project. He chose to use quality assessment to identify both strengths and weaknesses.

The instructor discussed his process for evaluation. He said that he typically evaluated each deliverable in two steps. First, a teaching assistant (TA) read through each submission and attempted to identify missing elements and/or inconsistencies. Yet he indicated that most TAs lacked the depth of experience to quickly understand and evaluate design materials; therefore, he did the detailed evaluation. The instructor indicated that detailed feedback was given for every identified weakness in the submitted materials. In order for the instructor to help his students see that there were many models that would work for a design, he indicated that he often provided feedback to recommend consideration of alternate approaches. Scoring was primarily based on identified weaknesses and missing elements, and secondarily on clarity and organization of the presentation.

Reasons Behind the Team/Cooperative Learning Approach

There were two major reasons the instructor provided for the team approach in his course. First, he felt that "the skills we are developing are a combination of technical and social skills, and are almost always practiced in a team environment." The second is that, given the time limitations of a single semester, a team project could have larger scope than an individual project, and was more likely to afford interesting design situations. When discussing the reality of the course project, he stated, "Professional software developers are called upon to invent some of the most complex artifacts that humans are capable of understanding. Almost all non-trivial development projects are undertaken by teams of engineers, because the combination of project complexity and time pressure puts them beyond the capability of even the most talented individuals."

The instructor discussed two types of assessments used to evaluate team effectiveness. First, he provided students with a detailed evaluation of the teamwork. He stated that he had a strong experience base with teamwork based on his industrial experience and his pedagogical knowledge of teams in the classroom. He felt that this gave students the feedback they needed to improve their understanding and skills of teamwork. The second was a simple peer evaluation, in which each team member allocated 100 "value" points to each team member (including themselves) for each project deliverable. This data was given directly to the instructor, and individual peer evaluations were not shared with the student being evaluated. Instead, students were given the mean of their peer evaluation scores. The instructor believes that because peer evaluation is common in professional settings and served multiple purposes in the context of the course, 1) "free-riding" was strongly discouraged, 2) students received honest feedback from their peers on the value of their work, and 3) teams were motivated to discuss the relative values of planned contributions as they organized their work. Due to holistic nature of his two types of team assessment, he felt that teaming in his course gave students a sense of the reality of teamwork in industry, coupled with the educational benefits that he knew to be supported by educational research.

The Student Teams and Projects
For this study, the researchers focused on three student teams throughout the project phases. For publication purposes, the team names NoBrainer, AppleTime, and PDQ are the pseudonyms that will be used throughout the paper. The following paragraphs will detail the project contexts, phase changes, and phase grades for each team. For the sake of space, team NoBrainer will be discussed in significant detail, but abbreviated versions of the project will be provided for teams AppleTime and PDQ.

Team NoBrainer
Team NoBrainer chose to work on a project, which they titled "Text Message Ordering System." This project focused on allowing restaurant customers to submit food orders from their mobile phones using text messages, without limit on which mobile provider was used. In phase 1 of the project, team NoBrainer gave this list of requirements for the system they planned to develop:

- Accept text message from any cell phone provider
- Track merchant business hours for product availability
- Provide list of merchants that use this ordering system
- Allow merchants to maintain current list of products and prices
- Provide user the ability to create "favorites" for quick ordering
- Handle financial transactions
- Web interface for users, allow registration, setting up favorites, and ordering from website
- Web interface for vendors, allow them to track orders and financial transactions through the system
- GPS location of cell phone user and locations of businesses
- Transaction confirmation and notification of time to pickup product

The instructor feedback for phase 1 for NoBrainer suggested that he did not understand how this was going to result in a clear first iteration of the project. He gave this team a 95/100 on this phase. After the return of the graded document for phase 1 to the team, the team and the instructor negotiated the appropriate first iteration of the NoBrainer idea.

For phase 2 of the project, the team set up a development plan for their project. They identified six risk areas. For example, team NoBrainer stated that one of the high risks for this project implementation was with the actual Ordering System. The team stated that the ordering system "requires confirmation from merchant. Will need to interface with merchant's internal ordering system to provide customer confirmations."

With this risk, they had a contingency plan to deal with this. Because it would impact the ability of the customer to receive accurate information about the order, the team stated that they would select and interface with a "popular particular merchant ordering system." They also identified three quality criteria. As an example, they stated, "The system shall allow at most one erroneous order per 1,000,000 customer text messages. Justification: if we fail to handle too many orders, we won't have any customers. Finally, they created a three-stage plan for development. Stage 1 addressed Core Functionality - which encompassed the actual development of the system and the process an order must go through to be completed. Stage 2 dealt with the Supplemental Functionality - which addressed the integration methods for the merchant ordering systems and possibilities for expansion. Stage 3 identified the Change Cases - which suggested GPS Integration for use to pick closest restaurant and developing a team web service to implement and integrate. For this phase, team NoBrainer received 98/100.

Phase 3 included the preliminary domain model for the project. Here team NoBrainer provided the UML model as a Class, a Messaging Sequence, and an Ordering Sequence. The Class diagram detailed the static view of the domain model for their project with three classes identified: the messaging system, the order processor, and the controller. The Messaging Sequence (Figure 1 shows the diagram as an example) detailed how the customer used the system to place an order, and the Ordering Sequence showed the order processing sequence interacting with the ordering interface. Phase 3 also included an identification of the known patterns that the team was applying. Team NoBrainer identified five patterns: the adapter pattern, the bridge pattern, the observer pattern, the strategy pattern, and the aggregate root. This phase was completed by a team evaluation based on the quality criteria identified in phase 2. For phase 3, the instructor feedback mainly focused on organization and communication issues. He asked for clearer explanations and wanted less confusing language. For this phase, team NoBrainer received 94/100.

[Insert Figure 1 here]

Team NoBrainer had a very detailed submission for phase 4. The team had five major sections of this submission: (1) Overview, (2) Ubiquitous Language (i.e., a glossary of terms), (3) UML Model (the diagrams of the model), (4) Narrative Descriptions (explanations of part 3), and (5) Assessment (evaluation of the quality criteria and assessment against risk factors). This phase is the refinement and extension of their phase 3 model, so the parts of the phase submission were relatively similar. The UML model was broken into five sections: Class, State, State and Event Class, Incoming Communications and Events, and Outgoing Communications. The diagram formats were very similar to those in phase 3. The patterns identified by NoBrainer went through changes. The team identified the patterns for phase 4 as: adapter pattern, visitor double dispatch, and strategy pattern. Again, the team concluded with an evaluation based on the quality criteria and added an assessment of the risks. Team NoBrainer was given 97/100 points for phase 4.

In phase 5, team NoBrainer refactored their design. The team chose to start their refactoring process with the need to handle asynchronous order processing. This resulted in the two major components of the system being refactored. First, the System Controller was replaced with an Event Dispatcher, and their Order Processing and Order Detail were replaced with an entity they defined as Order. According to the team, the second refactored component was changed to make it easier to understand. They and the instructor felt that the name "Order Processing" was confusing. The name implied that it was a service, but in actuality, it was an entity. They identified that the visitor pattern using double dispatch was the heart of their final domain model, and by the refactoring process, transformed their model into an event driven system. See Figure 2 for the state diagram for team NoBrainer Phase 5. The instructor comments for phase 5 were very positive. He suggested that there were still some confusing elements in the state diagram (Figure 2), but that if the team were going to go further, they would be ready to "cut code." Team NoBrainer received 98/100 for phase 5.

[Insert Figure 2 here]

Team AppleTime

Team AppleTime worked on a project that included the software Microsoft Small Business Accounting 2006 (SBA). The SBA software was designed to bring small business owners and managers the benefit and understanding of their accounting and finances with one, comprehensive solution. However, it is a desktop program, and there is no easy way to access it through a website nor to connected it with internet-based commerce systems. The goal of AppleTime's SBA project was to extend the SBA desktop application to Internet and website applications. As stated in AppleTime's project proposal, "Due to the complexity of the software and a limit of resources, our project goal is to have a common reusable framework that will allow integration between an e-commerce website and SBA." Their design supported the following setups to accomplish this: Local - SBA and website running on same hardware, Remote - SBA and website running on different sets of hardware, and Offline - website running on a server and SBA running on non-dedicated hardware. Table 1 highlights the five phases of their project by including a short explanation of the phase and the grade earned.

[Insert Table 1 here]

Team PDQ
Team PDQ chose a project that worked with the FIRST Robotics Competition (FRC). The FRC is a robotics competition for high school students to inspire interest and participation in STEM. The primary objective of team PDQ's project was to develop an enhanced robot controller application to run on the embedded hardware of the FRC. The secondary objective was to develop a PC application to act as a virtual test platform that interfaces with the robot controller application to allow code to be tested and refined without the robot. They chose this project because not only was the design element of their project very interesting, but also they felt that they were supporting a good cause (the FIRST competition). One of the team members had worked with a local high school involved with FRC, and he felt that the existing software being distributed to participants was a very poor example of software design. As a result, the team was highly motivated to produce a design that would be easy to understand and use, and serve as an example of good engineering for FRC participants. The following table (Table 2) highlights the five phases of their project by including a short explanation of the phase and the grade earned.

[Insert Table 2 here]

Student Team Process Narratives
For phases 2, 3, and 4 of the project, each of the teams was required to submit a process narrative. This narrative was to include the steps the team went through to arrive at the current stage of the project, focus on the parts of the phase that were easier or more difficult than what they expected, and reflect on why they believed this was true. Each team took a different outlook on how to complete the process narrative. In the analysis of these narratives, the researchers were looking for instances of team functionality in terms of elements of cooperative learning. Much of the narratives were procedural, which was also important to the instructor. The research here did not focus on the procedural sections of the process narratives. The following paragraphs will show the process and progress of each team through the analysis of each set of process narratives for the three phases.

Team NoBrainer
In phase 2, team NoBrainer began the process narrative by stating, "*I* would describe our initial approach..." (author emphasis). But the writer of this phase of the process narrative moved from the personal use of "I" to using team indicating pronouns throughout the rest of the narrative. The narrative demonstrated that the team started out brainstorming individually, without a meeting to guide their thinking. When they came together, they realized that they had defined the project in very different ways. In reflection, the team decided that from that point forward, they

would always come together as a team prior to assigning individual work. In this phase, team NoBrainer felt that it was difficult to come to agreement on portions of the assignment. Their final comments in this phase 2 process were that they were going to plan a "kick-off" meeting prior to starting any work to "gain consensus" and "assign pieces to each team member." This phase showed team NoBrainer started out as a group of individuals that wanted to divide the work, then put it together for a final project. Through this process, the team learned that they could not just work individually and produce a coherent product. They learned to set up times to negotiate meaning and define goals and outcomes before setting off on individual tasks. Yet they understood the necessity of individual accountability and continued to assign individuals to smaller tasks to be completed outside of team meeting time.

The phase 3 and 4 process narratives were written much more procedurally than phase 2. For each of these, the team detailed their step-by-step actions, without much mention of team dynamics. However, in both the phase 3 and phase 4 narratives, they provided evidence that they were using and depending on collaborative team brainstorming meetings, rather than brainstorming individually. In the phase 4 narrative, the team also discussed negotiation of evaluating when the project was complete. Through their reflections, this team demonstrated the need for more time to work together to figure out how to work as a unit.

Team AppleTime

In the phase 2 process narrative, team AppleTime organized the narrative in five paragraphs. The first paragraph talked about the teaming issues. They met as a team to talk about the deliverables and then they created a plan to divide the work having each team member focus on one of the areas: requirements, project planning, risk assessment, and quality criteria. Because the team felt that the functional requirements section was the most important part of the project, the team member who was going to write up this section gave his initial thoughts at the planning meeting. The team discussed his ideas and provided immediate feedback, and they also all provided more input later when the writer sent a rough draft of the functional requirements. Each of the other three team members proceeded to do their work separately, without input from the team. The process narrative indicated that the remaining four paragraphs of the narrative were written by the person responsible for that section of the phase submission. These four paragraphs were very procedural in nature. However, it is worth noting their language as individuals. The author in charge of functional requirements talked in terms that indicated teamwork, such as "we" and "our." But the other three authors that did not have team input for their sections used words such as "I" and "me." The three short paragraphs used personal pronouns ("I" or "me") four times, three times, and four times, respectively.

Again, the phase 3 and 4 process narratives for team AppleTime were mostly procedural. In both phases, team AppleTime only used pronouns referring to the team as a whole (i.e., "we" and "our"). In phase 4, the team also referred to themselves with their chosen name, "Team AppleTime." This growth of identifying themselves as a team, rather than a group of individuals, demonstrates the holistic growth of the team.

Team PDQ

Team PDQ is unique in the fact that these students did not live in close enough proximity to one another to have many face-to-face meetings. Much of their teaming was done through an online tool called "GoToMeeting." Through their reflection, they have shown that they could become a high functioning team even when not meeting face-to-face.

In phases 2 and 3, Team PDQ provided mostly procedural process narratives. However, both of these narratives provided evidence of seeing themselves as a team. This team used team-indicating pronouns exclusively throughout their phase 2 and 3 process narratives and used them often. They also indicated that they were working together on each of the parts of the project, and that they were not using a method of "divide and conquer."

This team's process narrative for phase 4 revealed many aspects of this team's functionality. They indicated that they were negotiating meaning of the project plan for phase 4 and working together to identify milestones for the project. Team PDQ used their meetings to plan out how to meet their objectives for phase 4. They discussed this planning meeting and in reflection stated, "This [meeting] really helped our team get on the same page."

Although not required, team PDQ submitted a process narrative for phase 5 as well. In this phase, the team had the chance to meet for their only face-to-face meeting. The team indicated that they had "more thorough" discussions and were able to "work out several ideas on a whiteboard, verbally explaining our thoughts as we drew them out." They wrote that one of their frustrations of the distance team meetings was that it was sometimes difficult to listen to a team member's ideas without being able to visualize it in the manner intended. This one face-to-face meeting was extremely valuable to them. The team stated, "We were able to break down barriers in communication and focus on the meat of the work for this phase."

Team PDQ demonstrated good teaming skills through working together to overcome their difficulties with distance, finding an appropriate venue to facilitate their meetings, and showing team unity through referring to the team rather than individuals. This team also demonstrated the value of face-to-face interaction. While teamwork can be successful using distance learning, having periodic face-to-face meetings can be extremely helpful.

Student Team Presentations

Student-team project presentations for phase 3 and 4 occurred over the last seven weeks of class. These short presentations, 10-15 minutes, were a forum for formative assessment of the project phases by the instructor, as well as a means for the team to inform and seek feedback from class-members about the project. The criteria for these presentations were specified in the course syllabus, which was provided on the first day of the course. There was obvious emphasis placed on the team's ability to articulate the process and challenges the team faced as they proceeded through the project phases.

From the course syllabus:

- All of the assignments are part of a multi-phase term project. The deliverables will include text and graphics to explain design ideas, and at least one assignment will involve a presentation to the class, with follow-up discussion and critique.
- Oral presentation. You will have opportunities to present your project work several times during the semester. Each member of the project team should take turns presenting your work to the class.

During the semester, each team was expected to give two separate detailed presentations, one for project phase 3, and one for project phase 4. Each team member participated in at least one team presentation. During the observations of the team presentations, the researchers noted that there was a consensus feeling and representation of ownership of each group's project. One of the indications of this ownership was that pronouns such as "our" and "we", rather than "I", were used by all team representatives during the project presentation.

In general there was noticeable difference in-group recognition between the phase 3 and phase 4 presentations. In the phase 3 presentations, the three groups (NoBrainer, AppleTime, and PDQ) spent the majority of time talking about the specifications and characteristics of their project design. Two of the three groups began their presentation by introducing themselves and giving the title of the group project, "Hi, I am Matt and this is our project..." or "Hello, our system is to provide a way for customers a way to order from their cell phones..." One team, "AppleTime," introduced the team project by acknowledging the team and the project, "Good morning, I am representing the group 'AppleTime' and the core function of our project..."

In the phase 4 presentations, the team presentation appeared to focus more on "selling" their project, and there appeared to be a greater emphasis on the teams articulating the process of collaborating on the design project. A noticeable difference in introducing the projects was observed between the phase presentations. During phase 4 all groups began with an introduction and acknowledgment of their group, "We are team 'NoBrainer,' we have been inspired to present a video that represents our design process." The "AppleTime" group was a bit more humorous about their design process, "So we are Mike and Brad, we are from team 'AppleTime.' I'm not going to go into depth on it, but here is a movie clip from the movie 'Office Spaces,' not the TV program 'The Office.' We were a bit more dysfunctional, anyhow, this clip represents the design process that we experienced in our project." The third team, "PDQ" was a bit more proactive in their reflection, "We were going to provide a fancy video, but we couldn't tied it in to our project or design process, so we will just stick to the plan…we would like to let you know about an on-line team collaborations tool, gotomeeting.com. It has lots of capacity and functions, it saved our lives. Because of logistics, work hours, and distance we could not meet in person as often as we found was necessary. So this tool allowed us to 'meet' as often as we needed and attend to team issues as our different schedules allowed."

Although the five elements of cooperative learning, Positive Interdependence, Individual Accountability, Group Processing, Social Skills, and Promotive Interaction [4] were not enforced or could not be evaluated during these presentations, there was evidence of group reflections and deep understanding of team and collaborative processes. Through the collaborative process individual members of the team were able to participate in the important skill of group dynamics and "the ability to function in multi-disciplinary teams", as stated in ABET Criterion 3(d) [15] in addition to putting their content understanding into practice on a real-world problem.

Conclusion

ABET states in Criterion 3(d) [15] that engineering students must demonstrate “an ability to function on multi-disciplinary teams.” As we enter an era of globalization, industry is placing a greater emphasis on engineers who are able to engage in collaborative working environments, i.e. teams. It is imperative that engineering educators teach engineers the relevant content but also train them to interact and work productively in collaborative teams.

The Project-based learning assignment articulated above may be one way to provide the necessary context for learning about software design and the important skills of group dynamics and process towards a coherent goal. Essential features of the project are: i) Student teams must be allowed to chose the projects and have to opportunity to negotiate with the instructor to refine project scope; ii) Teams must establish and provide justification for quality criteria for the project; iii) Students must progress in permanent project teams, and also be provided other opportunities (such as in-class assignments) to work together to build and practice the necessary skills to complete an assignment; iv) Teams must be allow to present their work to the class at each phase of the cumulative project, detailing project progress and reflecting on design process and team interaction; v) Students must participate in team and individual self-evaluation based on both quality of project deliverables and peer reflection and input at each phase of the project; and iv) The instructor must provide detailed feedback on both the process and content of project design.

Future research in this field should include the observations of students in the classroom to articulate the abilities and outcomes of various strategies of instruction and pedagogy. Further study on this course project will be reported in future publications

References

1. K.A. Smith, S.D. Sheppard, D.W. Johnson, and R.T. Johnson, Pedagogies of engagement: Classroom-based practices. Journal of Engineering Education, 94(1), 87-100 (2005).
2. R.M. Felder, D.R. Woods, J.E. Stice, and A. Rugarcia, The future of engineering education: Teaching methods that work. Chemical Engineering Education, 34(1), 26-39 (2000).
3. M. Prince, Does active learning work? A review of the research. Journal of Engineering Education, 93(3), 223-246 (2004).
4. D.W. Johnson, R.T. Johnson, and K.A. Smith, Cooperative learning returns to college. Change, 30(4): 26-35 (1998).
5. K.A. Smith, Teamwork and project management. BEST Series, McGraw-Hill, New York, NY (2004).
6. D.W. Johnson, R.T. Johnson, and E.J. Holubec, Cooperation in the classroom. Interaction Book Company, Edina, MN (1998).
7. D.W. Johnson and R.T. Johnson, Making cooperative learning work. Theory into Practice, 38(2), 67-73 (1999).
8. J. Dewey, Experience and education. The Macmillian Company, New York (1938).
9. B.G. Barnett and A.D. Brill, Building reflection into administrative training programs. Journal of Personnel Evaluation in Education, 3(2), 179-193 (1990).
10. E.M. Boyd and A.W. Fales, Reflective learning. Journal of Humanistic Psychology, 23(2), 99-117 (1983).
11. K. Ayas and N. Zeniuk, Project-based learning: Building communities of reflective practitioners. Management Learning, 32(1), 61-76 (2001).
12. R.K. Yin, Case study research: Design and methods. Sage Publications, Thousand Oaks, CA (1994).
13. P. Darke, G. Shanks, and M. Broadbent, Successfully completing case study research: Combining rigour, relevance, and pragmatism. Information Systems Journal, 8, 273-289 (1998).

14. R.K. Yin, Case study research: Design and methods, Third edition. Sage Publications, Thousand Oaks, CA (2004).
15. Criteria for accrediting engineering programs effective for evaluations during the 2006-2007 accreditation cycle <http://www.abet.org/forms.shtml>. Retrieved July 15, 2007.

Acknowledgments
The authors of the paper would like to thank the students of the Spring 2007, SEng 5802 class, as well as Caglin Akillioglu.

Authors

Tamara J. Moore is an Assistant Professor of Mathematics Education at the University of Minnesota. Tamara received her PhD in Engineering Education, as well as both her B.S. in Mathematics and M.S. in Mathematics Education from Purdue University. Prior to earning her doctorate, Tamara taught high school mathematics for seven years. Her research interests include the learning and assessment of complex problem solving, the learning of engineering teamwork, and the integration of engineering education into the K-12 mathematics and science classroom to encourage students to choose STEM disciplines of study.

Anne L. Kern is a doctoral candidate in Curriculum and Instruction, Science Education at the University of Minnesota. She has accepted an Assistant Professor position in the College of Education at the University of Idaho, Coeur d'Alene campus. Her research interests include teaching and learning in K-12 science, as well as extending "best practices" in instruction and pedagogy.

John E. Collins spent 30 years in industry before returning to the University of Minnesota, where he completed his Ph.D. in 2002. He is currently Director of Graduate Studies for the professional Masters program in Software Engineering at Minnesota, where he teaches in the areas of software engineering and artificial intelligence. His research interests include automated negotiation and rational decision processes.

Figure 1. Team *NoBrainer* Messaging Sequence from phase 3 of the project design.

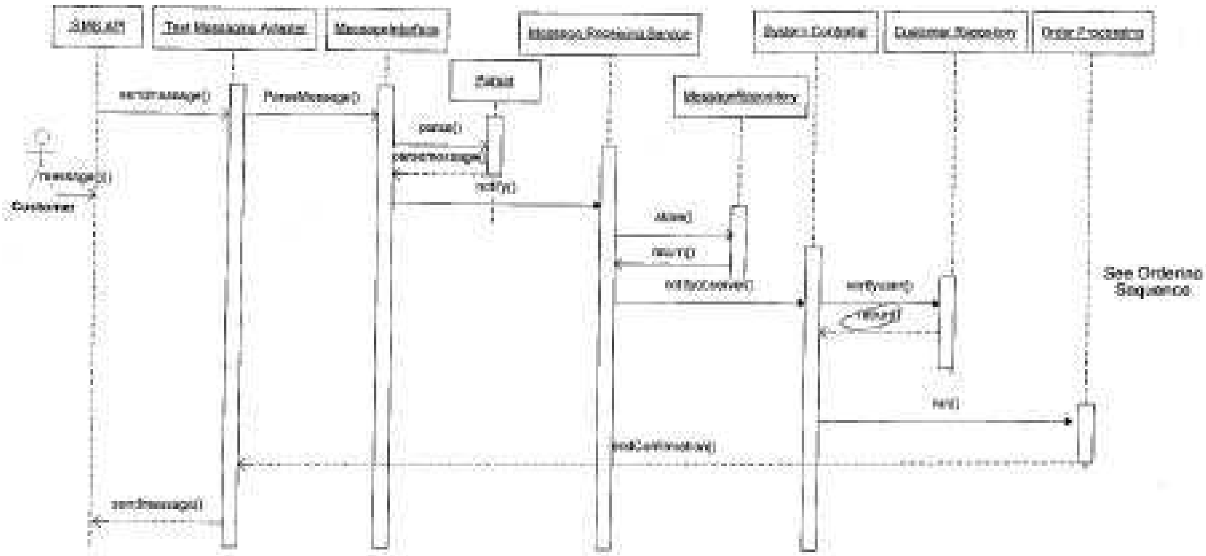

Figure 2. Team *NoBrainer* Order State Diagram with instructor comments from phase 5 of the project design.

\

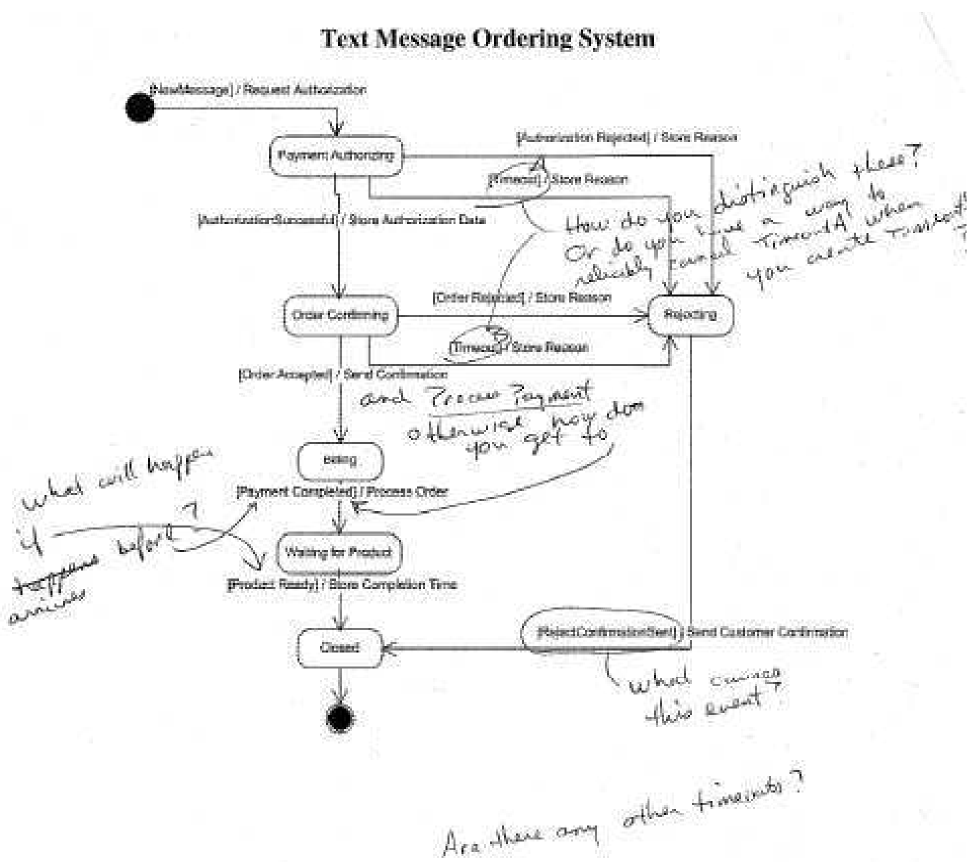

Table 1. A description of the project phases for team AppleTime including grades for each phase.

PHASE	Description	Grade
Phase 1	Phase 1 consisted of an executive summary, an overview, the project goal, the project description, the project requirements, and the member experience. The detailed requirements were to create a service and interface that would: ▪ Expose and simplify core SBA functionality to the web ▪ Allow remote websites to use the SBA Database ▪ Allow local websites to use the SBA Database ▪ Allow the website to work when SBA is offline while providing a sync process to update the SBA Database ▪ Allow dynamic XML message configuration and processing The instructor commented that there was not a clear bound on what the first iteration would be, so the team had to negotiate this with the instructor prior to the next phase.	90/100
Phase 2	Team AppleTime focused their phase 2 submission on the scope, requirements – which were much more detailed in this phase, the user scenarios, the areas of risk, the quality criteria, the detailed project plan, and the change cases. The instructor wrote that this phase lacked some justification for the team's areas of risk and did not organize the project plan in step-by-step procedures. Otherwise, his comments were positive.	95/100
Phase 3	In the domain model phase, team AppleTime had errors in their UML model. The instructor discussed problems with distinguishability of elements, associations seemed to be backward, and diagrams appeared to depend on non-existent references. He indicated that the team had a design that separates "get" services and "set" services, but did not justify why this was done or what value it added.	94/100
Phase 4	Team AppleTime had a less detailed phase 4 submission than the two other teams in this study. The had six sections that covered all of the requirements of the complete model: (1) overview, (2) domain model – which was significantly more detailed than in phase 3, (3) design changes – which consisted of only minor changes, but had better explanations, (4) software package structure, (5) class dictionary, and (6) quality criteria.	100/100
Phase 5	For phase 5, team AppleTime refactored 3 elements: ▪ changed createLocal, updateLocal, and deleteLocal methods to protected level of visibility. Added a setLocal method to the BaseSetSCARService, ▪ removed callSet() methods from DataPortal and inheriting classes, and ▪ reimplemented a true bridge pattern with the DataPortal and SCARSerializer interface. The instructor gave only positive feedback on this phase of the project.	100/100

Table 2. A description of the project phases for Team PDQ including grades for each phase.

PHASE	Description	Grade
Phase 1	This phase consisted of the project description and goals, scope, milestones to completion, a list of resources (which detailed their individual personal experience), requirements for the project, and benefits. The detailed requirements were broken into 2 parts: the robot controller and the robot hardware simulator.	100/100
Phase 2	Phase 2 focused on the scope of the project and the plan. There was little change in the project description for phase 2 as compared to phase 1. They defined the system components, the requirements including the Use Cases (initialization, autonomous operation, user controlled operation, and virtual hardware creation), the quality criteria and identified risks, and the anticipated changes. Team PDQ also developed a sophisticated project plan with details about effort, assumptions, and sub-milestones.	96/100
Phase 3	The main product of phase 3 was the preliminary domain model, which was a use case diagram and description. The team also included a glossary, an assessment of quality, risk and areas of change, as well as strengths and weaknesses of the design. The instructor scored this phase low. His comments indicated that the team did not provide a framework for the participants of the FRC to extend, which is what they promised in the previous phases. He felt that the team phase submission did not help him understand what the team model did or how the user could use it.	83/100
Phase 4	Because of the problems PDQ had in phase 3 of the project, their phase 4 submission was very detailed. They included three diagrams and "walkthrough" descriptions of each diagram for their complete UML model. A detailed design of the classes and an assessment piece were also included. They explained each of their steps thoroughly and noted that they figured out that communication issues were one of their problems in phase 3.	100/100
Phase 5	Team PDQ refactored 3 elements in phase 5. They chose to refactor a sensor, the autonomous table, and the controller from phase 4. They also noted that it was obvious that their design was not complete at this point, but due to the restraints and scope of the class, chose to stop iterating at this point. The instructor agreed with this assessment of the status of the project, and was very positive about the final product of team PDQ.	100/100