AC 2010-903: EVALUATION OF A STUDENT TEAM PROJECT IN AN INTRODUCTION TO SOFTWARE ENGINEERING COURSE FOR AEROSPACE ENGINEERS

Mary Lynn Brannon, Pennsylvania State University

Mary Lynn Brannon, Instructional Support Specialist at the Leonhard Center for the Enhancement of Engineering Education at the Pennsylvania State University, has a Master of Arts Degree in Education and Human Development specializing in Educational Technology Leadership. Her work focuses on projects that measure and assess student perceptions of learning related to their experiences with engineering course innovations. She is a faculty development consultant with previous experience in instructional design, and instructor of the Graduate Assistant Seminar for Engineering Teaching Assistants.

Oranuj Janrathitikarn, Pennnsylvania State University

Oranuj Janrathitikarn is a Ph.D. student in the Department of Mechanical Engineering, the Pennsylvania State University. She has a Bachelor of Engineering degree in Mechanical Engineering from Chulalongkorn University, Bangkok, Thailand, and a Master of Science in Aerospace Engineering from Penn State. Her master's thesis focused on the implementation of the Soar architecture on the six-legged robot. Her research interests are intelligent systems, cognitive architecture, unmanned ground vehicles, robotics, and software engineering

Lyle Long, Pennsylvania State University

Dr. Lyle Long is a Distinguished Professor of Aerospace Engineering, Bioengineering, and Mathematics at The Pennsylvania State University. Before that he was a Senior Research Scientist at Lockheed Aircraft, and served as a visiting scientist at Thinking Machines Corporation and NASA Langley Research Center. Additionally he was a Moore Distinguished Scholar at the California Institute of Technology. He has a D.Sc. degree from George Washington University, an M.S. degree from Stanford University, and a B.M.E. from the University of Minnesota. His research interests are in robotics, neural networks, software engineering, intelligent systems, computational fluid dynamics, and unmanned air vehicles. He is the recipient of the Penn State Engineering Society Outstanding Research Award, the IEEE Computer Society Gordon Bell Prize, and the Lockheed Aeronautical Systems Company award for excellence in research and development. He is a Fellow of the American Physical Society (APS) and the American Institute of Aeronautics and Astronautics (AIAA). Prof. Long has written more than 200 journal and conference papers.

Evaluation of a Team Project in an Introduction to Software Engineering Course for Aerospace Engineers

Abstract

Software engineering plays an important role in many industries, especially aerospace where the aircraft, spacecraft, and ground systems are often very large and complex, and safety and/or the mission require very safe software. In order to prepare aerospace engineering students to be more competitive in the aerospace workforce, Introduction to Software Engineering was developed at a the Pennsylvania State University in 2007. This senior-level course was designed to present software engineering concepts to aerospace students who have some background in computer programming, but no prior knowledge in software engineering. Students majoring in aerospace can select the software engineering course or an electrical engineering course as a requirement for the aerospace major in the eighth semester. In addition students can also take the course as an aerospace engineering elective or to fulfill requirements for minors in computational science or information science and technology. During the first two years, the course provided the materials based solely on lectures and talks from guest speakers. To provide a more real-world experience to students, a student team project was added to the course in the spring semester 2009, where they had to use the software engineering concepts. The pedagogical approach was to incorporate peer learning through teamwork that would involve the students in a problem-based learning experience.

The team project was designed with three objectives: to provide hands-on experience in software engineering through the development of a relatively small software system, to simulate the real working environment in a large company by having the students in the class work together as a team, and to emphasize the communication and collaboration skills among small groups in the software development model which are crucial skills in developing large and complex software systems.

The purpose of this paper is to describe the student team project including a discussion of how it improved the learning experiences of students and to share assessment data of student perceptions of working on a team. Preliminary findings indicate that participation in the team project increased the students' awareness of the importance of software engineering in the Aerospace industry. Individuals who are involved in the design and development of real world projects in software engineering courses and pedagogy may be interested in this paper.

Introduction

The need for software engineering education is important to the economy. The number of software disasters is growing[1]. Millions of dollars are spent on software disasters each year and this will grow as software systems become more complex. The Aerospace Department at the Pennsylvania State University recognized the importance of enhancing the curriculum to support the needs of current aerospace systems. Recruiters tell us that their companies are interested in hiring aerospace graduates that have studied software engineering or systems engineering[1]. A new course in software engineering was developed and first offered in Spring 2007 at the

Pennsylvania State University. Students in the aerospace major are required to take the software engineering course or an electrical circuits course.

Undergraduate engineering students must have mastery in engineering theory and concepts. Employers tell us that it is equally important for aerospace engineers to have teamwork skills and the ability to communicate systematically with electrical, computer, software and systems engineers to be successful in the workforce. In order to prepare students for a career in the aerospace industry, it is important for the students to experience and understand real world challenges and problems. Pedagogically, students learn best when they can apply theory to practice and also when they are in a peer learning environment. The course team project, development of a software system for a mobile robot, was designed as a hands-on peer learning assignment to enable students to experience working on a collaborative software engineering team. The scope, timeframes and complexity of most (engineering) projects require the effort of teams of engineers---experts in some aspects of engineering practice working in coordination with other experts[2].

Typically software engineering courses are taught in computer science and engineering programs.[3,4] Since those students in those majors have strong backgrounds in computer programming languages and algorithms, the projects focus mostly on designing and creating complex software systems which require high level of knowledge and experience in programming. Those courses require students to work in small teams of up to four members. Some courses ask students to contact customers from the industries directly to develop the software systems for them. This approach provides more real-world experience to students than working on the in-class projects.

In order to make the projects more challenging to students, the use of robotic projects had been implemented in many universities[5,6]. For example, one software engineering course at The University of Virginia was developed by using studio presentations for the class of over a hundred students [5]. This course required students to work in a small group and two groups were paired to work as a team. This course also included the closed laboratories to provide in-depth training on the particular skills.

In order to develop the software engineering course for students in majors other than computer science and engineering, other approaches should be integrated into the course because those students have limited background and experience in computer language programming. For example, another software engineering course provided the lectures in six areas: Computer Architecture, ADA 95 Constructs, Algorithms, Theory of Computation, Software Engineering, and Introduction to Other Classes[6].

The Department of Aerospace Engineering at Penn State University had offered the Introduction to Software Engineering for Aerospace Engineers for the first time in the spring semester 2007. The teaching method for the first two courses was mainly based on the class lectures, homework, and exams. In the spring 2009, the use of class projects was introduced for the first time. The project was the development of a software system for a mobile robot. A relatively unique aspect of the course is that it focused on giving students experience in working in a software engineering team. The students worked in teams of four whose task was to complete one of the phases of the development process, e.g., design, coding, and testing. These groups coordinated their efforts to form an integrated software engineering team. The number of students in the class necessitated the use of software development teams, which shared the same set of hardware systems. Having two teams also allowed the use of an end of semester competition. The instructor, Professor Lyle N. Long, took special training in software engineering to prepare to teach the courses; he is now a Certified Software Development Professional.

The objectives of the course are to provide hands-on experience in software engineering, and to simulate the real working environment of a large company using team work with an emphasis on communication and collaboration skills. "Collaboration is a process that crosses time and cultures. Increasingly, engineering endeavors involve teams scattered across continents, working toward a common purpose. Corporations are recognizing that synergized, distributed expertise can bring both needed engineering and cultural knowledge to a project"[2]. These objectives are achieved using a variety of active learning methods, including lecture, demonstration, problem solving, collaborative work, formal team work, and peer learning. The assessment plan provided for formative assessment via oral and written reports and tests; and summative assessment with the completion of software system for the final grade. Surveys and focus groups were conducted to obtain feedback from the students on their perception of the learning experience.

Course Description

Introduction to Software Engineering for Aerospace Engineers was designed around two sets of activities: the traditional activities and the team project.

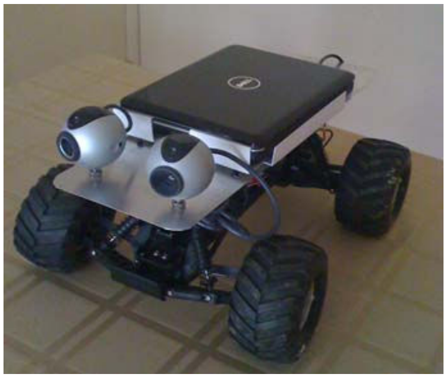
The traditional activities were in-class lectures based on a software engineering text book written by I. Sommerville[7], reading assignments, homework, and exams. The lectures included the illustration of the role of software engineering in aerospace industries through various media such as news and short video clips. The lectures also included talks by the guest lecturers that had expertise in software engineering.

The team project was implemented in the course to reinforce the use of the software engineering concepts in practice. .The project was started in the second week of the course and the project activities were executed in parallel with the content in the class lectures.

Project Description

As mentioned earlier, the team project was to develop a software system for a mobile robot. The hardware system is a radio-controlled truck with two on-board cameras, as shown in Figure 1. This system was built by the instructor. A Dell Mini 9 laptop was used to control the truck. The mission objective is to control a mobile truck wirelessly to find the simulated landmines in a park on campus. Each team had to develop the software system to control the truck via wireless LAN. At the end of the semester, the two teams competed with each other to find the most landmines within a 50 minute class period.

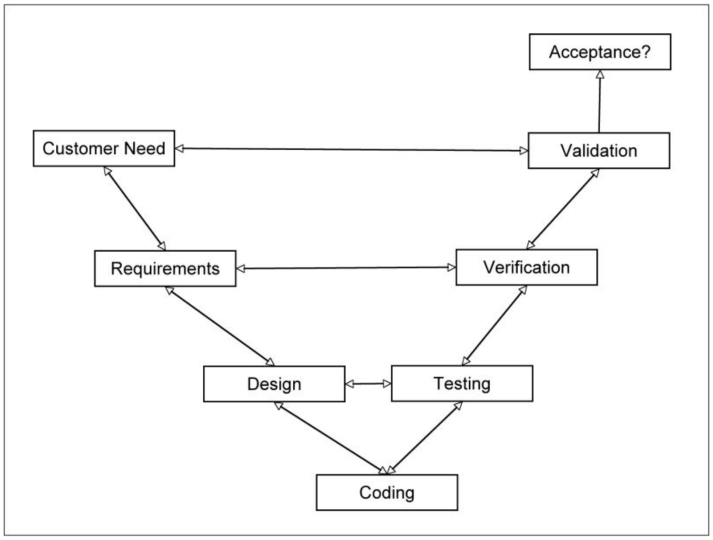
Figure 1: Radio Controlled Truck



Participants

This is a senior-level course with 47 students from diverse ethnic backgrounds (44 male, 3 female). Students were divided into two software engineering teams. Members in each team were separated into six small groups, Requirements, Design, Coding, Testing, Validation, and Verification, according to the V-Life cycle software engineering model (in subsequent years a Costing group was added as well). The role of each group was defined in the **S**oftware **E**ngineering **B**ody of **K**nowledge (SWEBOK)[8]. Each group had four to five people to execute the task according to the V-Life Cycle. Figure 2 shows the V-cycle of the software development model.

Figure 2: V-Life Cycle Model



Activities

After the students were divided into two software engineering teams, the Requirement groups met with the instructor, who acted as a customer, to discuss the system needs. The Requirement groups had to write the formal system requirements and then turned this report over to the Design groups. The role of the Design groups was to design flow charts and object diagrams of the software system and pass them to the Coding groups. The Coding groups would code the software and send the codes to the Testing groups to test them on the hardware system. When the system passed testing, the Verification groups would check the system with the Requirements groups, and the Validation groups would check with the customer. Throughout the execution of the project, each group had to communicate with all others as such communication was critical to the successful development. The groups set up private communication space in the course management system. In addition, regular interaction between three pairs of groups was necessary: Design and Testing, Requirements and Verification, the Customer and Validation.

Presentations

Each team was required to give a brief oral presentation to the customer every two weeks to report the progress of their project. The presentation was closed to the opposing team to keep the technical information of each team confidential. A report, signed by all group members, was submitted for grading after each presentation. At the end of the semester, the final report including all of the code of the software systems was submitted, and a system demonstration occurred during the final competition. The competition was an in-class demonstration. Each team had fifty minutes to set up their software systems. When the system was ready, the truck was taken outdoors to the park (Figure 3). Four members who represented each team were selected to control the truck from an office in an adjacent building by seeing the real-time pictures from the webcams (Figure 4). The remaining students of the class were observing in the park. One representative remotely controlled the truck to find the simulated landmines (Figure 5) and marked the locations on the provided map. The team that found the most landmines within the shortest time was the winner of the contract.

Figure 3: Truck with team outdoors

Figure 4: Truck with web cams

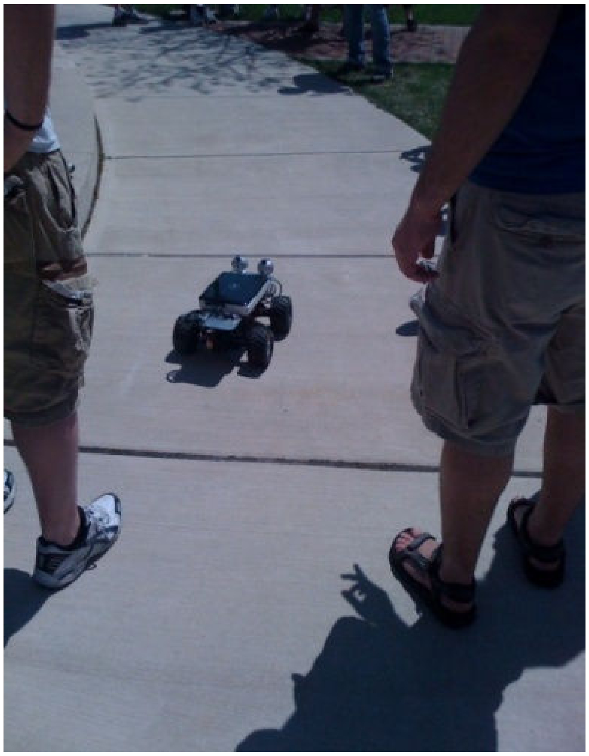
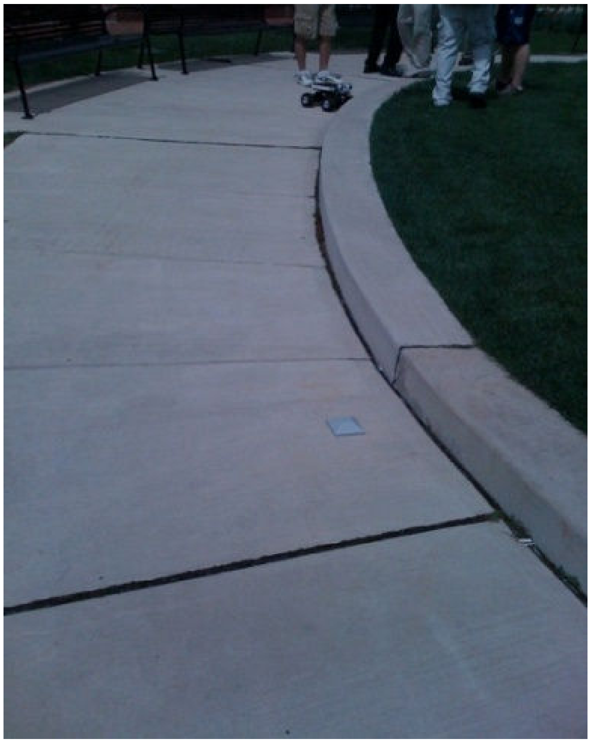


Figure 5: Truck and land mine



Assessment of Team Project Effectiveness

Both formative and summative assessment techniques were utilized to assess the effectiveness of the software engineering team project. Formative assessment included the bi-weekly progress team reports. In addition an online course survey of student perceptions of learning, a focus group, and a student performance questionnaire were used to obtain feedback on the team project. Summative assessment focused on the grading of the final project with all supporting materials. The online survey was completed by 30 students and 12 students participated in the focus group. These were used for both formative and summative purposes.

The students completed a Software Engineering Project Survey and a Course Objectives Survey at the end of the course. The surveys focused on students' perceptions and attitudes about working in a team environment; students were asked to rate whether they felt the course objectives were met. The survey items, along with frequency data and descriptive statistics for each item can be found in Appendix A. Students' consent to use the survey and focus group data was obtained as per university policy.

The students' perceptions of the team project were positive. Students gave honest and constructive feedback regarding the team experience. Approximately 87% of the students agreed or strongly agreed that the team project experience helped them to have a better understanding of software engineering in practice. Approximately 70% of the students agreed or strongly agreed that the course lectures supported and influenced their participation in the team project.

The students gave feedback on how to improve the team project. The students recommended that more guidelines up front coupled with more time to work on the project would have been helpful to the success of the teams. The students also wanted more industry samples and examples in order to help the teams understand their roles and responsibilities. The reaction to the team environment was not surprising given the concern that students have with team dynamics and cohesiveness. To manage the workload more evenly, students recommended that individual mini-team assignments would have been helpful to scaffold the learning into the larger team project. Most of the students felt that collaboration was not difficult. One student stated, "Almost everyone knew somebody from each of the subgroups on a personal level, so communicating something to another group was not hard." When asked to describe how the team project changed the students' perception on software engineering, one student commented "it gave me a better understanding on how communication works within a team project and across different teams." Another student said, "The project showed me how important communication is for large team projects". A total of 90% of the students agreed or strongly agreed that working on a real world simulation team project helped them to better understand the details and scope of such a project.

When asked what challenges they encountered working on a team, the students reported the following as the top three challenges: communication across the sub-teams, accountability within the team, and scheduling team meetings. One student shared that he learned the importance of staying in contact with other members of the sub-team to try and solve problems, such as needing a coding member at every meeting to work through the code. Given the characteristics of the students, most being Aerospace Engineering majors, a majority of the students said that the team project increased their awareness of the importance of software engineering in aerospace applications and concurrently increased the individual students' interests in software engineering in general.

Focus Group Results

The focus group data provided rich and deeper feedback. Focus group discussions allow in-depth exploration of the reasons why the participants think the way they do. Questionnaire results reveal usually only what people think, not why[9]. The focus group protocol (Appendix B) consisted of three discussion segments. The students' primary comments are discussed below. Focus groups are conducted to gather information from students in a very directed and specific arena to generate student's opinions, attitudes and experiences. The focus group method provides a quick and effective method to obtain experiences from participants. It can provide content rich qualitative information and reveal insights that are difficult to capture with other methods.[9]

The first discussion segment asked the students to think about the experience working on the team project.

What did students like best about the project?

Each sub-team is dependent on the other sub-teams. The previous sub-team finished its work and handed it to the next sub-team in the team. The students could see the transition of the work from the previous sub-team to the next sub-team, the dependability, and the responsibility of each sub-team.

What could be done better for future students in the team project?

Students wanted more programming background. Most of the work on the project depends on the coding sub-team. If there were more people in the coding sub-team or the members in the coding sub-team had more programming skills, the product would be a lot better than relying on only four people.

And what did the students like least?

Both teams had to share the same hardware. The verification and validation sub-teams thought that there should have been more communication between both sides of the V-cycle.

The second discussion segment asked the student to think about the learning experience, i.e. the time you spent on the assignments, class preparation and project management.

“Time spent on this class depends on which sub-team each person is in. For example, the validation sub-team spent only one hour to finish it while the coding sub-team spent many hours. The time requirement is not uniform among all sub-teams in the teams."

The third discussion segment asked the students to describe their experience with the team project in one word or phrase; and then in your own words explain how you would recommend this course to another student. Students said, “innovative”, “modern course”, “two thumbs up”, take it”, “the project was fun”, “surprised with your result” and “challenging and a little bit unorganized”. Most students agreed that the goals of the project, to provide hands-on experience in software engineering and to simulate the real-working environment in a large software company, were met.

Conclusion and Reflection

The 2009 Spring semester was the first time the instructor integrated a team project in the software engineering course. This method was chosen because the instructor believed there is a need for students to experience the practice of software engineering development. The results from the survey and the focus group indicate that the team project was very effective in teaching the concepts of software engineering to students and in demonstrating a real-world working environment. In the future, the course team project will be improved upon based on recommendations from the students. A major change will be to devote more class time to the project. The project scenario will be changed because the source code for the mobile truck is now public. As the project continues to evolve, the instructor will incorporate current and timely software engineering project activities to show the students the value in working on a team. What made this project innovative is that software engineering is integrated into the total course allowing students to be immersed in working on a software engineering design team which simulates an actual team in a software engineering company.

A student commented in an email to the professor, “I interviewed for a systems engineering position. I told them about your class and what I have been learning in it and they were very impressed. I mostly talked to one of the senior engineers on the project and he said what we have been doing in class is exactly what they are doing on their team. He said taking that class will look very good for me getting the job. Just thought I would mention that because I thought it was really cool how closely your class matches up with what happens in the real world.” The student got the job.

Acknowledgement is made to the Leonhard Center for the Enhancement of Engineering Education at the Pennsylvania State University for its support of this project.

Bibliography

1 Long, L., (2008), “The Critical Need for Software Engineering Education”, *The Journal of Defense Software Engineering,* January 2008

2 Sheppard, S. D., Macatangay, K., Colby, A., Sullivan, W. M. (2009), Educating Engineers: Designing for the Future of the Field, pp. 7-8, The Carnegie Foundation for the Advancement of Teaching, Preparation for the Professions.

3 Sindre, G., Stalhane, G., Brataas, G., Conradi, R., “The cross-course software engineering project at the NUNU: four years of experience” *Software Engineering Education and Training, 2003. (CSEE&T 2003). Proceedings. 16th Conference* 2003, pp. 251-258

4 Gustafson, D. A. (1998). "Using robotics to teach software engineering." *Frontiers in Education Conference, 1998. FIE '98. 28th Annual*, 551-553 vol.2.

5 Knight, J. C., Horton, T. B. (20005). "Evaluating A Software Engineering Project Course Model Based On Studio Presentations", *Proceedings of the 35th Annual Frontiers in Education Conference, 20005, (FIE "05. S2H-21*

6 Lundqvist, K., and Srinivasan, J. (2006) "A First Course in Software Engineering for Aerospace Engineers." *Software Engineering Education and Training, 2006. Proceedings. 19th Conference on*, 77-86.8

7 Somerville, I, (2006), **Software Engineering,** Addison-Wesley

8 Bourque, P., and Dupuis, R. (2004). "Guide to the Software Engineering Body of Knowledge 2004 Version." *Guide to the Software Engineering Body of Knowledge, 2004. SWEBOK.*

9 Kontio, J., Lehtola, L., and Bragge, J. (2004) "Using the Focus Group Method in Software Engineering: Obtaining Practitioner and User Experiences." *International Symposium on Empirical Software Engineering, ISESE 2004. Proceedings. August 2004,* , 19-20

Appendix A: Frequency data and descriptive statistics from project survey

Each survey response was coded from 1-5, “strongly disagree being 1 and strongly agree being 5. Means and standard deviations are calculated using the coded responses. The coding system was used for the Software Engineering Project subscale.

Software Engineering Project (SEP) Subscale

Item N=30 (100%)	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Mean	Standard Deviation
1. Working in the team project helped me to understand software engineering in practice	1 (3.3 %)	0	3 (10.0%)	19 (63.3%)	7 (23.3%)	4.033	.808
2. The course lectures supported my participation in the team project.	1 (3.3 %)	6 (20.0%)	5 (16.7%)	15 (50.0%)	3 (10.0%)	3.433	1.04
3. The course lectures influenced my perception of the team project	1 (3.3 %)	3 (10.0%)	6 (20.0%)	17 (56.7%)	3 (10.0%)	3.600	.932
4. Homeworks supported my participation in the team project	1 (3.3 %)	11 (36.7%)	11 (36.7%)	5 (16.7%)	2 (6.7%)	2.866	.973
5. Homeworks influenced my perception of the team project.	0	10 (33.3%)	8 (26.7%)	10 (33.3%)	2 (6.7%)	3.133	.973
6. I can relate and apply the course fundamentals to the team project.	0	0	5 (16.7%)	12 (40.0%)	13 (43.3%)	4.266	.739
7. The team project increased my awareness of the importance of software engineering in aerospace applications.	0	0	5 (16.7%)	10 (33.3%)	15 (50.0%)	4.333	.758
8. The team project increased my interests in software engineering.	0	3 (10.0%)	7 (23.3%)	7 (23.3%)	13 (43.3%)	4.000	1.050
9. The team project made me consider a Master’s and/or Ph. D. degree in software engineering.	5 (16.7%)	10 (33.3%)	9 (30.0%)	4 (13.3%)	2 (6.7%)	2.600	1.132
10. The textbook is useful in working on the team project.	3 (10.0%)	6 (20.0%)	8 (26.7%)	11 (36.7%)	2 (6.7%)	3.100	1.124
15. Working as a large team caused difficulty in collaboration.	0	8 (26.7%)	11 (36.7%)	3 (10.0%)	8 (26.7%)	3.366	1.159

18. Working in a team project helped me to understand the course material better.	0	2 (6.7%)	5 (16.7%)	12 (40.0%)	11 (36.7%)	4.066	.907
19. The work load for the course was more manageable because I was able to work with a team.	0	2 (6.7%)	5 (16.7%)	12 (40.0%)	11 (36.7%)	4.066	.907
20. The pace of the team project was appropriate for the amount of work involved.	1 (3.3%)	5 (16.7%)	6 (20.0%)	11 (36.7%)	7 (23.3%)	4.366	.668
21. Working on a real world simulation team project helped me to better understand the details and scope of such a project.	0	0	3 (10.0%)	13 (43.3%)	14 (46.7%)	4.366	4.41
N=29 (100%)							
23. I think that the project is a good way to learn software engineering.	0	0	2 (6.7%)	13 (43.3%)	14 (46.7%)	4.41	.627
27. Because of the team project, I feel better prepared for a job in software engineering	0	1 (3.3 %)	5 (16.7%)	15 (50.0%)	8 (26.7%)	4.03	.778

Appendix B: Focus Group Protocol

Focus Group Questions
AERSP 440, Introduction to Software Engineering for Aerospace Engineers

AERSP 440 Course Focus Groups Spring 2009

The purpose of the focus group session is to solicit student perceptions and experiences in the AERSP 440 course during this past spring semester. The data collected will have no identifying information. Your feedback and ideas are very important to us and will help us improve and enhance the course in the future. This process will help us to understand your perceptions, both good and bad, of your learning experience this semester. We are primarily interested in perceptions on your experience with the team project. If you speak about the contents of the focus group outside the group it is expected that you will not tell others what individual participants said.

Please let us know what you think.

First discussion segment:

Think about your experience working on the team project. On an index card, list 3-5 items that you liked best about the team project. Next, list 3 items that would have bettered prepared you for AERSP 440. Finally, list 3 items you liked least about the project.

Second discussion segment:

> Think about your learning experience in AERSP 440, the time you spent on assignments, class preparation and project management. Think about how you managed your time. List 5 items that the AERSP 440 professor could do to improve these experiences.

Third discussion segment:

> Please think about this question and then write your answer on the index card. If a fellow student, who will be taking AERSP 440, asked you to describe your experience with the team project, what would you say in one word or phrase? Then, in your own words please explain how you would recommend this course to another student.

Appendix C: Software Engineering Team Project Assignment

Course Project Description

AERSP 440: Introduction to Software Engineering for Aerospace Engineers

Instructor: Prof. Lyle N. Long
TA: Oranuj Janrathitikarn
Spring, 2009

The project for this course will be the development of a software system for a mobile robot. Mobile robots are beneficial in many applications especially for interplanetary exploration and operation in extreme environments. A radio-controlled truck with an on-board computer will be controlled by a desktop computer via wireless LAN through the internet [Ref. 1]. Therefore, the truck will be able to operate remotely anywhere in the world with a wireless internet connection coverage (). In addition, the truck will have an onboard camera to display the real-time environment. At the end of the semester, the truck will be tested in a remote environment. The project evaluation will focus on the quality of the software as well as the documentation, team communications, and final test. This is *not* a programming project, it is a software engineering system project.

This course project is designed to:

1. Provide hands-on experience in software engineering through the development of a relatively small software system for a mobile robot;

2. Simulate the real working environment in a large software company by having the students in the class work together as a team. Each student will work in a small group functioning as a part of a software development model. The instructor will be a customer who indicates the system needs;

3. Emphasize the communication and collaboration skills among small groups in the software development model which are the most crucial skills in developing a large and complex software systems.

There will be two completely separate teams (blue and white), competing against each other. Students on each of these teams will be separated into five (or more) small groups according to the V-Life cycle software engineering model based on their interest:

□ Requirements
□ Architectural Design
□ Coding
□ Unit Testing
□ System Testing

Each of these sub-groups will have roughly four people each. After the customer needs have been discussed, each group will work together to produce the software system, design, and documentation according to the software engineering approaches [Ref. 2]. The communication log between each group must be recorded via Penn State Wiki (https://wikispaces.psu.edu).

Every week, each group must give a brief oral presentation about their progress in the class. At the end of the software development cycle, the final product will be demonstrated and the report must be submitted.

1. Long, L.N., Sharma, A., and Souliez, F., "Client-Server Java Programming for Wireless Mobile Robots," AIAA Paper 2003-0459, AIAA Aerospace Sciences Meeting, Reno, NV, Jan., 2003.

2. Sommerville, I., *Software Engineering*, 8th Edition.