GROUP PROJECTS IN SOFTWARE ENGINEERING EDUCATION*

Jiang Guo
Department of Computer Science
California State University Los Angeles
Los Angeles, CA 90032
323 343-6674
jguo@calstatela.edu

ABSTRACT

Software engineering is a very important course in computer science program. In our software engineering education, we have adopted group-based projects as an approach to improve students' learning experience. The group-based collaborative projects will not only enhance student problem solving and critical thinking skills, but also will increase student motivation, confidence and resilience. This paper discusses our small-group-based approach in our software engineering course.

1. INTRODUCTION

The Software Engineering course at Cal State LA focuses on the needs of industry. The industry needs information-intensive systems of unprecedented reliability, capability, performance, and scale. As the software grows more and more complex, the advances in software engineering technology will continue to grow and an ever increasing rate. The Software Engineering course at the Cal State LA provides the students with a fundamental understanding of the theory and practice of software engineering – the use of sound engineering principles to economically develop software that is reliable, flexible, maintainable, and that works efficiently on real machines [1]. This course is designed to provide students knowledge and skills of software development to implement large-scale software systems using the current available tools.

Computer science program in Department of Computer Science at California State University Los Angeles received ABET accreditation in 2007. Software Engineering

* Copyright © 2009 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

course plays an important role in the evaluation of output of the computer science program.

In our ABET report, assessment data is derived from various measures including Major Field Test (MFT), Learning outcomes exam, and project analysis in Software Engineering course [2]. The Software Engineering project assessment process is used to evaluate the program's Objectives/Learning Outcomes based on students' group projects implementation. The Objectives/Learning Outcomes directly related to group projects are:

– *Students will have the training to design and implement a large software system and will have the ability to work both individually and collaboratively.*

– *Students will have sufficient oral and written communication skills.*

The assessment of the outcomes is based on the faculty's evaluation to the group project design and students' development skills, and the oral and written communication skills.

2. GROUP-BASED PROJECT IN SOFTWARE ENGINEERING EDUCATION

To take Software Engineering class, the students have to finish prerequisite courses, including Using Relational Databases and SQL, Object-oriented Programming, Programming with Data Structures, Web and Internet Programming, and Data Structures and Algorithms. Through learning these prerequisite courses, students are expected to master a main programming language (such as Java) and database manipulating and operating language SQL and have some basic concepts of web-based programming.

During the lecture hours of Software Engineering class, we cover methodologies and tools for the specification, design, development, testing, evaluation, and maintenance of software systems. Roger Pressman's software engineering book is used in the classes [3]. This course also satisfies the upper division writing requirement. At the end of the course, students are able to:

- Estimate the cost and effort for software projects
- Make schedules for software projects
- Elicit software requirement
- Create data models, dataflow models, and behavior models
- Convert requirement models into software architectures
- Implement component-level design
- Choose proper software testing strategies
- Use software testing techniques

Software Engineering at Cal State LA is a lab-oriented core course. Students are divided into groups of 4 to 5 members each. Each group is responsible for the design and development of a software system collaboratively, including documentation, implementation, and presentation of the group project. Many studies show that students can learn more through participating in group activities, spending substantial time in labs and interacting with faculty [4] [5]. Other studies also show that groups are very helpful to students [6]. So, the software engineering groups are expected to encourage students to be more active learners and to spend more time with peers, and this approach will greatly improve students' software development abilities and increase their interests in pursuing IT careers. Group projects not only enhance student self-study skills in programming, but also improve their written and oral communications skills expected in large software development [7]. At the end of the quarter, students are required to give an oral project presentation and a written report, including requirement specification, design specification, source code and test report. The faculty will assign a ranking (1 – 5) in the three categories: Collaborative development, Oral Communication, Written Communication. From 2002 - 2007, the average scores in Project Development, Oral Communication, and Written Communication varied between 3.6 and 3.9.

Increased student-student interaction is a key of group project success. It is the hands-on software development experience involved with software engineering course that make it so important to the software educational world.

From the group project in software engineering course, the following benefits are expected:

- Improved students' software development experience.
- Enhancement of problem solving and critical thinking skills for students.
- Students trained to master a comprehensive set of computing skills and to apply them quickly in a realistic hands-on environments. Such skills are highly relevant to current practices of the academic research community and IT industry.
- Students trained to have teamwork coordination skills and offer them valuable opportunities for leadership development.

To achieve our goals and objectives, each group of students needs to elect a group leader, who is in charge of coordinating the tasks between group members. In order to provide high quality learning experience for students, the instructor works with group members in the lab twice per week. Once a month, the instructor hosts progress presentation seminars, during which participating students and group leaders give brief presentations. The group activities prepare participating students for interactive learning, practical software development, leadership roles, as well as to function and work in group environment. The approach provides several successful mechanisms for students. These mechanisms include: collaborative group-based learning communities, student-student relationships, project advising from faculty, and oral and written presentations. The group-based collaborative projects not only improve student performance, but also increase student motivation, confidence and resilience [8].

3. SOFTWARE PROJECTS AND PLATFORMS

As presented in Figure 1, this approach provides undergraduate students with hands-on learning experiences in project-based project groups. These project groups are created by students' interests. The instructor doesn't force students to join a specific group. The students join a group totally depends on their interests. They need to work collaboratively with peers. The project topics are selected by students. If students can't find their preferred topics, the instructor assigns a topic to a group. This approach integrates activities of all participants: participating group members, group leader and the faculty. The faculty and undergraduate students work closely to implement the projects.

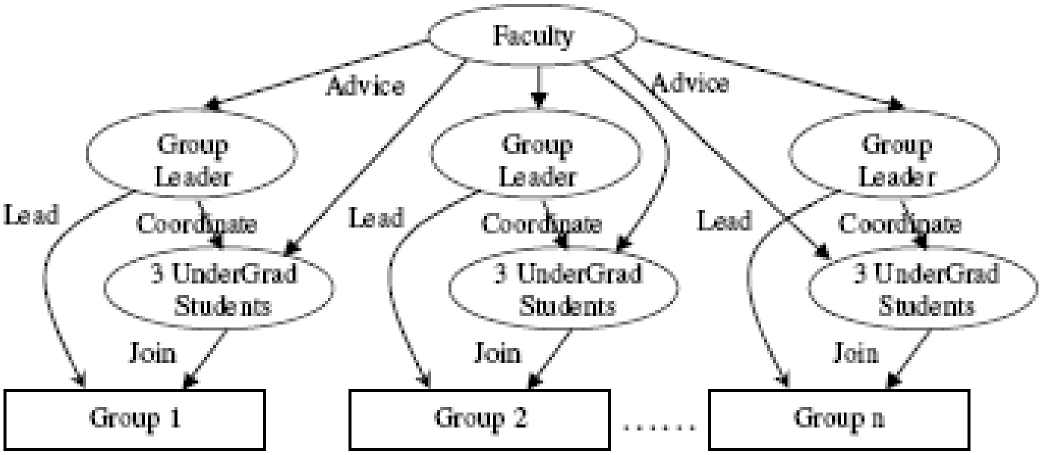

Figure 1. Organization and Management of Group Projects

Group projects developed in this course not only involves choosing appropriate analysis, design, and testing approaches to perform certain requirement analysis, software design and testing tasks, but also involves using best software tools to implement development process. Fortunately, most of the projects are web-based systems. These web-based projects will be developed by using cutting-edge powerful software development platform J2EE techniques, such as n-tier architecture, JSP, Servlet, Filter, etc. Most software tools that we use are open source software. For n-tier web-base systems, we use following platforms:

- Database: we use MySQL as database server. It is downloadable at http://www.MySQL.com, we use "community version, window(x86) version". The installation is straight forward.
- JDBC: MySQLConnector/J "mysql-connector-java" is a good JDBC. It can be downloaded at http://www.MySQL.com. This connector provides ability for Java programmers to operate and manipulate database.
- IDE: Eclipse is a very popular and free IDE. It is downloadable at http://www.eclipse.org. This tool is very powerful and developed in pure Java. So, it can be easily installed on Windows and Linux operating system environments.
- Web Server: we use Apache Tomcat. It is at http://www.apache.org. This web server also provides J2EE API to use Servlets and JSP. It is fully J2EE specification satisfied, so the applications developed under this environment can be easily moved to WebSphere or WebLogic, which is very expensive.
- Frameworks: Apache Struts is available at http://www.apache.org and Spring is available at http://www.springframework.org/. Struts is a very useful web framework. For most students' projects, it is powerful enough. Spring is much more powerful and complex. It also provides other features, such as IoC, AOP.

Although students know some basic concepts of web-based programming, normally they need a couple weeks training to use all above tools fluently. In our lecture hours, we use sample-based approach to teach students how to install and use these tools. Thereafter, students can focus on their projects.

4. DOCUMENTS AND STANDARD TEMPLATES

Written communications skills are important for students IT careers. So, documentation for the group projects must be developed. Students have to submit four documents: (1) Requirement Specification, (2) Design Specification, (3) Code, and (4) Test Report. To help students finish these documents, we provide documentation templates based on IEEE standards [9] [10].

- Requirement Specification mainly includes: Operating environment, Design and implementation constraints, Database considerations, Hardware interfaces, Software interfaces, System function 1..n, Use case model, Performance requirements, Safety requirements, and Security requirements.
- The major parts of Design Specification include: System Overview, Design Considerations, Architectural Strategies, External databases and/or data storage management and persistence, Concurrency and synchronization, System Architecture, Detailed System Design and Graphical User Interface Design.
- Code is the source code of the project. We require students to follow the Java coding convention.
- Test Report is straight forward. It includes: report of test, cases problem tracking and test summary

5. SAMPLE PROJECTS

In recent years, students finished many projects in our Software Engineering course. Through working on those projects, students improved their oral presentation skills, written presentation skills and collaborative development skills. They also enhanced their skills of software requirement analysis, architecture design, component design, algorithm representation, system integration, and system testing through the group project development. Here is a list of some projects.

- Empty Pockets.com Online Store
- Video Rental Business
- Automated Online Auto Auction
- Traffic Lights Control
- Dating System
- MyGrocery.com
- Snowboard Software Simulation
- Ant Program Generation
- Gas Station Simulation System
- Book Trading System in Library
- Net CAD System
- Music Sharing System

Some are very good projects, we select two good projects as samples.

5.1. Fantastic Integrated Courseware

In order to give students more flexibility and allow them to study and review lecture notes and do homework and assignments at home or any other location, an integrated web-based courseware - Fantastic Integrated Courseware - was developed by our students. The core functionalities of this system are its abilities to help students and faculty improve learning and teaching processes. This courseware is designed to facilitate university faculty and student coursework activities so that they have user-friendly interface to effectively interact with one another. Currently, some faculty members are using personal websites or Wikis to manage teaching materials. This project is to provide an efficient and well-organized web-based system to satisfy faculty members' and students' needs in their daily academic activities. This software enables faculty members to build their website quickly and easily. It also allows students to download lecture notes, submit homework and discuss questions in forums. Instructor and students have their own calendars which show events marked according to themselves respectively. Instructor have right to add new event, edit and delete the posted events. Instructor can associate assignments with calendar events so that students, who are enrolled in that class, can see the due dates of assignments through their calendars.

This courseware is fully modularized. All the modules are scalable and new features are easy to add. Right now, it has following modules: user login, files uploading, calendar, class information management, forum, email, and search.

5.2. NetCAD System

Computer-aided design system, or CAD system, has been widely used by countless users including mechanical designers, civil engineers, architects, and so on. CAD application software helps those users keep from tedious traditional design process with pens, erasers, and scales by providing convenient, interactive, automated design process assisted by computers. However, since most of CAD application programs run as stand-alone programs, there are still some inconveniency and incapability in fields of team designing and data management. To overwhelm those problems, our students developed a NetCAD System as efficient, secure, and manageable networking CAD system by applying concurrent multiple network connection, user authentication, and data-centered architecture.

The NetCAD System was designed to support collaborative design process. It consists of two application programs. One is the client program called NetCAD Client and the other is the server program called NetCAD Server. A client program and the server program communicate with each other via TCP/IP stream socket. The NetCAD Server supports concurrent connections with multiple clients; therefore, multiple client programs can access to the server program simultaneously.

6. CONCLUSIONS

Students can learn more through participating in group activities, spending more time in labs and interacting with peers and faculty. Software engineering group project based approach can encourage students to be more active learners and to spend more time with peers. This is the essential of our approach to increase the motivation, confidence, collaboration, self-study abilities of computer science students. The hands-on project development involved with this approach is very well applied to the education of software engineer – a core course of computer science program. Our approach can enhance undergraduate student learning by engaging students in structured, project-based collaborative groups advised by the faculty.

REFERENCES

[1] Andriole, S., Freeman, P., Software Systems Engineering: the Case of a New Discipline**,** Software Engineering Journal, Volume 8, Issue 3, May 1993.

[2] Self-Study Questionnaire for Review of the Computer Science Program, Department of Computer Science, California State University Los Angeles, June 2006.

[3] Pressman, R., Software Engineering: A Practitioner's Approach, McGraw-Hill, 2004.

[4] Astin, A., Oseguera, L., Degree Attainment Rates at American Colleges and Universities, Los Angeles, *Higher Education Research Institute*, UCLA, 2002.

[5] Astin, A., Achieving Educational Excellence: A Critical Assessment of Priorities and Practices in Higher Education, *Jossey-Bass*, San Francisco, 1985.

[6] Astin, A., Minorities in American Higher Education: Recent Trends, Current Prospects, and Recommendations, *Jossey-Bass*, San Francisco, 1982.

[7] Guo, J., Liao, Y., Software Engineering Education and its Roadmap for California State University Los Angeles, *Proceedings of the 2005 International Conference on Frontiers in Education: Computer Science and Computer Engineering*, 2005.

[8] Guo, J., Using group-based Projects to Improve Retention of Students in Computer Science Major, Journal of Computing Sciences in Colleges, Volume 23, Issue 6, June 2008.

[9] IEEE Recommended Practice for Software Requirements Specifications, 830-1998**,** Software Engineering Standards Committee of the IEEE Computer Society, October 1998.

[10] IEEE Recommended Practice for Software Design Descriptions, 1016-1998, Software Engineering Standards Committee of the IEEE Computer Society, September 1998.