PBL APPLIED TO SOFTWARE ENGINEERING GROUP PROJECTS

DECLAN DELANEY AND GEORGE G. MITCHELL
Department of Computer Science, National University of Ireland, Maynooth, Ireland.
email: decland@cs.may.ie, georgem@cs.may.ie

This paper describes the application of Problem-based learning (PBL) to the design and implementation of an E-commerce web site by small groups of software engineering students. This work is part of a real-world software engineering course, taught to pre-internship students. The use of PBL has gained significant interest since its inception in the late 1950's, and its later adaptation to small team-based learning in the early 1960s. By combining the PBL paradigm along with the experience of teaching a "traditional" software engineering course, and by analyzing feedback from industry, a course, which we believe provides students with new insights into real-world software engineering projects, has been developed.

Initially students were formed into teams of 4 or 5 members based on the weak-strong selection technique. The course began with team-building activities, after which the E-commerce project proposal was presented to the teams. The teams were given complete autonomy over their software development strategies but they were required to work with the clients (mentors) to elicit the project requirements and specifications. Emphasis was placed on the methodology employed (the '*how*') rather than on the end product (the '*what*').

Assessment of the students focused on three main areas, in keeping with the PBL paradigm. Firstly, *implementation skills* were assessed by examining the final product and documentation provided by the teams. Secondly, *teamwork and leadership skills* were evaluated through the use of short anonymous self-assessment and team-assessment questionnaires, as well as by their demonstrated ability to organize meetings and manage their team skills. Finally, *analytical thinking and inter-personal skills* were evaluated through personal journals and a detailed group presentation. The journals outlined their journey through the learning process and demonstrated their ability to identify and analyze critical variables in the development cycle. The presentation was followed by hard-hitting questions based on Bloom's taxonomy from the faculty staff members.

This paper details all aspects of the course development and execution and concludes with an evaluation of PBL and its application to software engineering education.

Keywords: software engineering, Problem-based learning, group projects

1. Introduction

This section deals with two distinct issues:

(1) the current status of group projects in software engineering curricula;
(2) the Problem-based learning (PBL) approach to education.

1.1 Software Engineering Curricula

Although the 1991 publication of the ACM[1]/IEEE-CS[2] Joint Curriculum Task Force report on Computing Curricula [1] never explicitly mentioned *Software Engineering*, many universities have developed undergraduate software engineering degrees since its publication [2][3][4]. In 1998, the IEEE-CS and the ACM established the Joint Task Force on Computing Curricula 2001 (CC2001) to undertake a major review of curriculum guidelines for undergraduate programs in computing. In contrast to the 1991 report, the current report involves the creation of several volumes reflecting the diversity of the computing field, including an explicit Software Engineering Volume. This volume is being produced through the IEEE-CS/ACM SWEEP[3] project [5]. This project is guided by eleven principles which state, "*The professional practice of software engineering encompasses a wide range of issues and activities including problem solving, management, ethical and legal concerns, written and oral communication, working as part of a team, and remaining current in a rapidly changing discipline.*"

1.2 Problem-based learning

Problem-based learning (PBL) is a generic learning technique and has its origins in the late 1950s [6]. We are interested in the *small group PBL* introduced by McMaster Medical School in the 1960s [7].

[1] Association for Computing Machinery
[2] Computer Society of the Institute for Electrical and Electronic Engineers
[3] Software Engineering Education Project

PBL reverses the traditional approach to learning. Traditionally students are presented with course material and are subsequently tested on that material through individual examination. PBL on the other hand introduces the students to a problem and the students must explore the solution domain in a self-learning capacity. Through appropriate guidance the students will learn the course material through practical application. It is for this reason that PBL has proven so successful in medical education, where student doctors apply theoretical course material to the practical diagnosis of patients' symptoms.

The core aspects of PBL are [8]:

?? Learning takes place within the contexts of authentic tasks, issues, and problems-that are aligned with real-world concerns;

?? In a PBL course, students and the instructor become co-learners, co-planners, co-producers, and co-evaluators as they design, implement, and continually refine their curricula;

?? The PBL approach is grounded in solid academic research on learning and on the best practices that promote it. This approach stimulates students to take responsibility for their own learning, since there are few lectures, no structured sequence of assigned readings, and so on;

?? PBL is unique in that it fosters collaboration among students, stresses the development of problem solving skills within the context of professional practice, promotes effective reasoning and self-directed learning, and is aimed at increasing motivation for life-long learning.

2. PBL and Software Engineering Group Projects

Since PBL is a generic learning technique it has been successfully applied to a number of differing environments; medicine; dentistry; veterinary and more recently engineering. The Computer Science Department at NUI Maynooth decided to apply PBL to teach software engineering process to third year students. This PBL course was entitled *"Real World Software Engineering."*

The educational objective of the course was to teach software engineering process skills through a team-based software project. There is broad agreement that the teamwork aspect of software engineering curricula is inherent to the successful education of software engineers [9]. This led us to integrate the PBL model with the IEEE-CS/ACM curriculum guidelines for software engineering. PBL offered an excellent opportunity for students to practice, apply and develop skills such as problem solving, team building, ability to cope with change and both self and group assessment. This was the first opportunity the students had to experience PBL in a team-based environment.

For this reason the emphasis of the module was on team-participation skills and on the application of software development processes rather than exclusively on the delivery of an end product.

The course was developed for third year undergraduate students who had studied core computer science subjects but only had a limited amount of software engineering knowledge. It was decided to include the group project as part of the third year program for a number of reasons:

?? Industry: feedback from industry indicated that the students found it difficult to apply software engineering skills during their work internships, which begin at the end of third year;

?? Consolidation: it provides a means of consolidating the disparate skills and theoretical knowledge acquired after completing half of the degree program;

?? Preparation: it prepares the students for their individual capstone project during the fourth year program.

3. Logistics and Software Engineering Group Projects

In this section we will examine three key logistical aspects of presenting and developing real-world software engineering group projects:

Team organisation

Choice of problem

Course Delivery

Team Organisation

In keeping with McMaster's *small group PBL,* and the experience of other universities in software engineering courses, each team consisted of either four or five members [3][8]. Team members were selected based on the weak-strong selection technique so that academically weaker students would gain the advantage of working with their academically stronger peers [4][10][11][12]. This made the teams self tutoring and enabled student-centered learning, independent of the lecturers.

The Problem

The choice of the problem for the software engineering project course is fundamental to PBL. Solving the problem must lead the students through an adventure of discovery that covers all the material within the course definition. The team size must be taken into consideration and each team must be given the same problem. For these reasons a modular E-commerce software project was assigned to the students. The deliverables consisted of a software product and software documentation. The product development required investigation of topics such as databases, network security, scripting languages and web site design. To produce the documentation the students were obliged to conduct requirements analysis, design specifications and verification and validation of the product.

The task assigned to the groups was to develop an online E-commerce store complete with the database and including security and billing transactions. The breadth of the problem meant that a complete solution would have been very difficult to achieve within the allowed time without adopting a rigorous software engineering approach and good project management. This emphasized the *How* rather than the *What* of the software development process. This emphasis is common in academic software engineering group projects [13].

Course Structure

The course module was spread over a 12-week period and consisted of 4 hours per week contact time with mentors and 4 hours per week independent work. Each team of 4 therefore had 384 man-hours to complete the project.

The initial encounter centred on motivational exercises. The students were invited to physically divide into their allocated teams. Initially a simple exercise was given to the class and each student had to work on the problem for a short length of time. Afterwards the students were invited to pool their results with other students in their team and come up with a second solution. This was the beginning of a conviction that group work pays. Afterwards a second exercise was carried out, to instil the importance of teamwork, planning and asking questions.

The weekly lecture contact time was concentrated in a single half-day session. This gave the students a relatively long period of time to work together and provided them with an opportunity to meet the clients (tutors) under controlled conditions.

Attendance was compulsory for the first hour, after which the students were free to chose a location that best suited their team. This allowed the mentors to present various topics to increase the students' body of knowledge and to guide the students in core issues, such as project management and running meetings. The students were then expected to demonstrate their understanding of this material through practical application.

4. Assessment

Assessment of group projects is a challenge, particularly where the individual contribution of each student is required. For example, a peer assessment approach is advocated by Rosen [14], whereas Younessi and Grant employed the Capability Maturity Model [15]. Maintaining the standard of assessment was critical in the development of this course and as a result it was decided that pure team-based assessment might be a disadvantage to the stronger students and equally be abused by weaker students. A 70/30 mix of team and individual assessment was finally decided upon.

The purpose of the assessment procedure was to assess three groups of skills. These skills were associated with four different deliverable components as illustrated in figure 1. The three groups of skills are outlined in the following subsections.

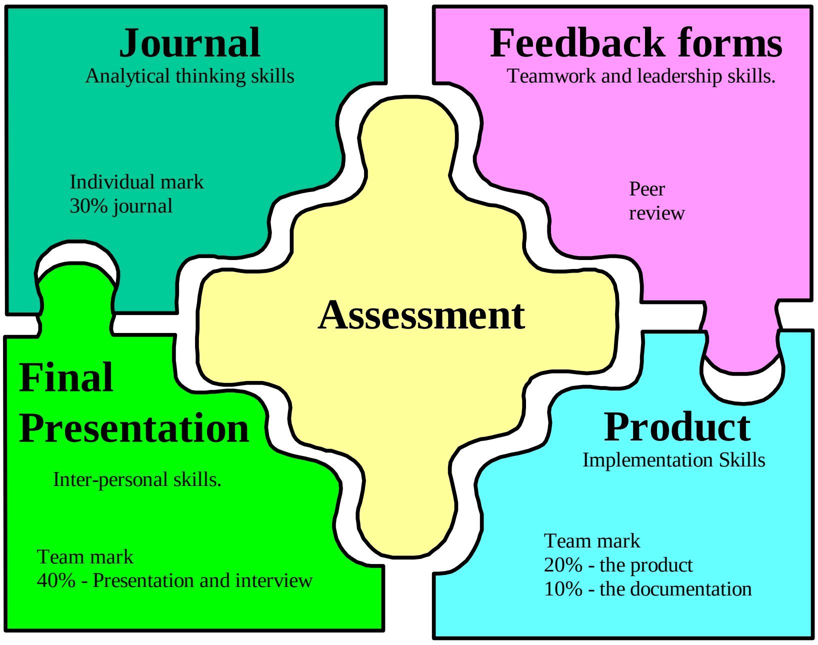

Figure 1. Assessment Components of Group Projects

4.1 Implementation skills

These skills represented the ability to programme an implementation of a given software design specification and to produce software documentation based on best practice. This accounted for 30% of the overall course mark. Each team was awarded a single mark and this went towards the individual's final course mark. The documentation accounted for 20% of the final mark with the software product being worth only 10%. This low mark reflects the fact that most university courses aim to develop software product implementation skills. While these skills are important they are susceptible to a high degree of plagiarism. One possibility we considered was to give no marks for the product implementation skills, but this would have been a demotivating factor.

4.2 Teamwork and leadership skills

These skills reflect how each student operates in a team environment **-** how they contribute to the team, how they organise the team and assign roles and responsibilities and how they integrate any industrial experience with their theoretical knowledge. These skills were monitored every two weeks using two separate anonymous feedback forms: an individual self assessment form and another form to assess the strengths and weaknesses of other team members. This provided an on-going peer review process throughout the software development project. This on-going assessment is core to PBL and is referred to as *formative assessment* [8]. Although no marks were directly awarded for these skills, they formed an important feedback mechanism for the lecturers and were used to gauge team spirit and allow for timely corrective intervention.

4.3 Analytical thinking, problem solving and inter-personal skills

These skills were assessed using a number of components. The students' journals were assessed by the lecturers on three separate occasions providing another means of formative assessment. The overall journal mark was 30% which represented the students' entire individual assessment mark. On conclusion of the project each team was requested to prepare a twenty minute presentation of their work which was shown to faculty staff. After each presentation students were rigorously questioned on all aspects of the project. These questions were built upon Blooms Taxonomy [16] to evaluate all six competencies and associated skills. Each presentation was given a single team mark which represented 40% of the final course work.

5. Conclusions and Recommendations

This paper has outlined the application of Problem-based learning to a new software engineering group project university degree course. The importance of team work within software engineering curricula was placed in context and the core elements of PBL were explained. The organisation and assessment procedures for the PBL-based software engineering project course were also outlined.

The new module aims to present the students with a realistic software development problem within a controlled university environment and is to be assimilated into the existing CSSE (Computer Science and Software Engineering) degree.

The experience the students gain from this course can be directly applied to their later careers in industry or elsewhere.

References

[1] Web Site of the IEEE Computer Society Educational Activities board: http://www.computer.org/education/

[2] S. L. Duggins and B. B. Thomas, An historical investigation of graduate software engineering curriculum, Proceedings of the 15th conference on Software Engineering Education and Training (CSEET), Kentucky, USA, February 25th -27th 2002.

[3] J. S. Collofello and M. Hart, Monitoring team progress in a software Engineering Project class, 29th ASEE/IEEE Frontiers in education conference , San Juan, Puerto Rico, November 10th -13th , 1999.

[4] M.J. Oudshoorn and K.J. Maciunas, Experience with a Project-Based Approach to Teaching Software Engineering, Proceedings of the Southeast Asian Regional Computer Confederation 5th Annual Working Conference on Software Engineering Education, pp. 220-225, Dunedin, New Zealand, November 1994.

[5] The Software Engineering Education Project (SWEEP) website: http://www.acm.org/serving/se/SWEE.htm

[6] D. T. Vernon and R. L. Blake, Does problem-based learning work? A meta-analysis of evaluative research, Academic Medicine, *68*(7) 550-563, 1993.

[7] M.G. Camp, Problem-based Learning: a paradigm shift or a passing fad? Medical Education Online, 1:2, 1996.

[8] D. R. Woods, Problem-Based Learning: Helping your students gain the most from PBL, ISBN: 0-9698725-1-8, 1995.

[9] A. J. Kornecki, I. Hirmanpour, M. Towhidnajad et al, Strengthening Software Engineering Education through Academic Industry Collaboration, 10th Conference on Software Engineering Education and Training (CSEET '97), Virginia Beach, VA April 13th - 16th, 1997.

[10] J. Polack-Wahl, Group projects: Woman and Men Can Work Together in the Computer Science Realm, Conference Proceeding of the 1999 International Symposium on Technology and Society, IEEE Network, July 1999, pp. 245-249.

[11] Y. Ouyang and S. S. Wang, Real-World Object-Oriented Design Experience for Computer Science Students, 29th ASEE/IEEE Frontiers in Education Conference , San Juan, Puerto Rico, November 10th -13th , 1999.

[12] A. Jones A. and M. Birtle, An individual assessment technique for group projects in software engineering, Software Engineering Journal, Vol. 4, No. 4, pp226-232, July 1989.

[13] P.N. Robillard, Team Project in Software Engineering, Canadian Conference in Computer Engineering, Waterloo, 7th-8th May 1997.

[14] C. C. H. Rosen, Individual Assessment of Group Projects in Engineering : a facilitated Peer Assessment Approach, Proceedings of the 9th Conference on Software Engineering Education, pp 68-77, 1996.

[15] H. Younessi and D. D. Grant, Using the CMM to evaluate Student SE projects, Proceedings of Software Engineering: Education and Practice, Dunedin, New Zealand, pp 386-391, January 24th -27th 1996.

[16] "Tools for Teaching," Barbara Gross Davies, Jossey-Bass; San Francisco, ISBN: 1-55542-568-2, 1993