View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by MURAL - Maynooth University Research Archive Library

SOFTWARE ENGINEERING MEETS PROBLEM-BASED LEARNING

James D. Delaney BEng, MEngSc, Chartered Engineer, Department of Computer Science, NUI Maynooth
George G. Mitchell BSc, MSc, Department of Computer Science, NUI Maynooth
Sean Delaney BEd, MEd, Coláiste Mhuire Marino

A PROBLEM-DRIVEN APPROACH FOR TEACHING SOFTWARE ENGINEERING TO UNDERGRADUATE STUDENTS IS ENTERING ITS THIRD YEAR AT THE DEPARTMENT OF COMPUTER SCIENCE AT NUI, MAYNOOTH.

Real-world Education

There is growing agreement within universities and industry that experiential learning or 'learning by doing' should be emphasised more than the traditional 'chalk and talk' approach. This is confirmed by the on-going review of software engineering undergraduate programmes, which is being conducted as part of a joint international engineering-computing bodies project [1]. International approaches to providing this experiential learning have been documented and they include the *Studio-based approach*, the *Learner-centered approach* and the *Project-oriented approach*. These approaches focus on individual project work under simulated industrial conditions.

A New Teaching Paradigm

The approach adopted in Maynooth to give students a 'real-world' experience in software engineering is based on a learning technique introduced by McMaster Medical School in the 1960s. This approach is called small group *Problem-based Learning* (PBL).

PBL reverses the traditional approach to learning. Traditionally students are presented with course material and are subsequently tested on that material through individual examination. In contrast, PBL presents the student with a real-world problem and the students must explore the solution domain in a self-learning capacity. The students will learn the course material through guided practical application of knowledge to a suitable task. PBL has been successfully applied to civil, mechanical and electrical engineering as well as to medical, dental and veterinary education.

The introduction of PBL into educational courses is related to a constructivist theory of learning. This theory is based on the premise that learners come to know something new, not by passively hearing it, but by actively engaging with it and connecting it to what they already know. In other words, a learner's current knowledge is challenged and through interaction with other people and with the broader environment the learner constructs new knowledge. This is illustrated graphically in Figure 1, which shows that the gap existing between the knowledge the learner currently possesses and the knowledge the learner must acquire is bridged by having to complete a challenging project in a team setting. The lecturers' role is to scaffold the students' learning.

The PBL Course Structure

The PBL course introduced as part of the third year computer science and software engineering degree program is called "*Real World Software Engineering Group Project*" [2]. The educational objective of the course is to develop students' software engineering process skills. These skills include requirements elicitation and analysis, architectural design, testing and project management. The course emphasises the *How* (process) rather than the *What* (product) of the software development process. No single software development process is advocated, but students are guided in identifying key software engineering activities, which they must piece together themselves to successfully complete the project. To achieve this requires their working as a team to investigate and study relevant topics such as project

management, software engineering, databases, network security and client-server programming. Each student documents their individual journey of discovery in the form of a reflective journal in which they note facts, opinions and detailed considerations.

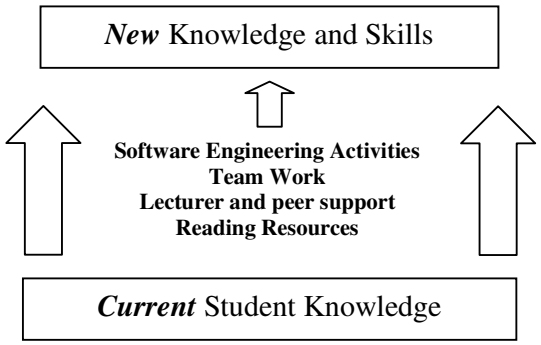

Figure 1: Students build on their existing knowledge by being challenged by new software engineering activities.

Teams of four students are formed. Team members are selected so that academically weaker students will gain the advantage of working with their academically stronger peers. A modular software project is assigned to each group. A typical project might be the development of a complete E-commerce web site or the development of an email access portal.

Course Duration	12 weeks
Contact time with mentors	4 hours per week
Independent learner study time	4 hours per week
Total available project time for each team	384 hours

Table 1: Breakdown of course times

The course is presented based on what we call *Pairwise Mentoring*, where two mentors present the course in tandem. Table 1 explains the breakdown of group project time. The weekly contact time with mentors is concentrated in a single half-day session to give the students a relatively long period of uninterrupted time to work together and to provide them with an opportunity to meet the clients (mentors) under controlled conditions. Attendance is compulsory for the first hour, after which the students are free to choose a location that best suits their current software process activity. The mentors exploit the compulsory hour to guide students in core issues, such as project management, how to run meetings and analysis of progress made.

In summary, therefore, the core principles of PBL are [3]:

- Learning takes place within the context 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;
- The PBL approach stimulates students to take responsibility for their own learning, since there are few lectures and no structured sequence of assigned readings;
- PBL fosters collaboration among students, stresses the development of problem solving skills within the context of professional practice, promotes effective reasoning, and is aimed at increasing motivation for life-long learning.

Assessment

Assessment of small group PBL projects is a challenge, particularly where the individual contribution of each student must be graded. The PBL paradigm stresses the importance of assessment and that pure team-based assessment might be a disadvantage to the stronger

students and equally be abused by weaker students. The authors therefore implemented a mixture of team and individual assessment techniques, assigning 70% of the marks for group work and 30% for individual work.

The assessment procedure aimed to assess three groups of skills. To assess each skills group, various assessment techniques were employed and staff members other than the mentors were involved. The skill groups and assessment techniques can be summarised as follows:

1. Implementation skills: ability to implement a software design specification and to produce software documentation based on best practice. The assessment was based on the delivered product and relevant documentation (30% of final mark).

2. Teamwork and leadership skills: ability to operate 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. The assessment was based on a group presentation and individual interview (40% of final mark).

3. Analytical thinking, problem solving and inter-personal skills: ability to think through a problem, analyse a situation, deal with pressure and communicate with the clients. The assessment was based on an individual reflective journal and feedback forms (peer and self evaluation forms) (30% of final mark).

Benefits of the approach

Benefits of Small Group Problem-Based Learning		
For Students	For Industry	For University
• Finding work placements is easier • Better-prepared for final year capstone project • Acquire essential management skills • Suits learners who prefer solving real-world problems over a period of time rather than being assessed in a single end-of-year exam	• Resourceful, self-motivated students • Experience of team environment • Experience of working to tight time deadlines • Easier assimilation into the workplace	• Motivates students to take more interest in computer science and software engineering subjects • Provides a more authentic assessment tool • Allows students to apply theory in a practical way • Issues arise in the process that might not arise in conventional classes

Concluding remarks

The success of small group PBL in developing software engineering skills is reflected in the positive feedback received from both the students and from industrial partners who participate in the 3rd year work-placement program. Students express their enjoyment of the course despite the amount of work involved, and found themselves better prepared for their subsequent industrial work placements. Examples of the work conducted by the students include development of the first university ePrints archive in Ireland, development of a web portal for an Irish government project (Muintir na Tire) and development of the Dublin Neighbourhood Watch website.

Contact Details

Declan Delaney, tel: (01) 7083354 email: decland@cs.may.ie

Bibliography

[1] IEEE-CS/ ACM SWEEP (Computer Society of the Institute for Electrical and Electronic Engineers / Association for Computing Machinery Software Engineering Education Project) - The Software Engineering Education Project (SWEEP) website: http://www.acm.org/serving/se/SWEE.htm

[2] J. D. Delaney and G. G. Mitchell, 'PBL applied to Software Engineering Group Projects,' International Conference on Information and Communication in Education, ISBN 84-95251-70-0, pp. 1093-1098, 13-16 November 2002, Badajoz, Spain.

[3] Working Group on Problem Based Learning, Ellis A.(chair), "Resources, Tools, and Techniques for Problem Based Learning in Computing," ITiCSE 1998, Dublin, Ireland, pp41-56.