# Notice of Retraction

After careful and considered review of the content of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

We hereby retract the content of this paper. Reasonable effort should be made to remove all past references to this paper.

The presenting author of this paper has the option to appeal this decision by contacting TPII@ieee.org.

# A Problem-based Learning Approach to Teaching an Advanced Software Engineering Course

Ming Qiu
Software School
Xiamen University
Xiamen, P.R.China 361005
Email: mingqiu@xmu.edu.cn

Lin Chen
Software School
Xiamen University
Xiamen, P.R.China 361005
Email: 1590814chenlin@163.com

**Abstract—Problem-based learning is particularly suitable to a multidimensional software engineering course. This paper describes the use of problem-based learning in a course "Advanced Software Engineering" given at Xiamen University, P.R.China. This approach is conducted a blended learning environment, a combination of a face-to-face learning environment and ELearning environment. A learning manage system based on Moodle was implemented to support blended learning. A set of integrated projects were selected as stimulus to learning. Both inter- and intra-group collaborative learning are encouraged. A survey conducted in the end of the course showed that students accept the problem-based learning quiet well, and their academic achievements were also better than expected.**

***Keywords—problem-based learning; software engineering education; learning management system.***

## I. INTRODUCTION

Software engineering is a multidimensional field that involves activities in various areas and disciplines [1]. Teaching a software engineering course is a balance between theory and practice. The students need to know the theory like methods, techniques, and standard practices. They must also get the opportunity to apply the theory in realistic projects to develop skill and get valuable experience. Hence, problem based learning (PBL) has become a commonly used approach in several educational institutions [2], [3], [4].

PBL is a contextual, collaborative, and constructivist learning environment [5]. One of the key elements of PBL is that it is student-centered. Students make their own judgment on what they should learn. PBL is usually under teacher guidance. The teacher designs the problem simulations and guides students in developing their learning. Problems are solved through collaboration processes inside small groups. Teachers play the role of facilitator or guider who shares information and guides the group through the learning process. This organization fosters discussion and collaborative discovery, placing the focus on the process instead of on the result itself.

This paper describes a way of teaching software engineering courses using PBL approach. A set of integrated projects were chosen as stimulus to the students. Both an face to face learning environment and an E-learning environment were set up to encourage self learning and collaborative learning. The results of the evaluation at the end of the semester show that students are satisfied with the new approach, and their communication skill is greatly improved.

The following section will give brief information about the course and explains the course design. In Section III presents the evaluation results. Finally the last section gives conclusions based on our experiences.

## II. COURSE DESIGN

### A. Course Description

The course "Advanced Software Engineering" is designed for junior students in the undergraduate program in a Software Engineering major at Software School of Xiamen University, P.R.China. It covers the Unified Process (UP) [6]; various components of the Unified Modeling Language; Object-oriented Analysis and Design (OOAD); and communication skills in a team environment. The prerequisite for the course is "Introduction to Software Engineering", which introduces the basic methods, tools and procedures for software development. It includes the discussion of the software engineering lifecycle, the common software engineering paradigms, and their key components (planning, requirements analysis, design, etc).

The primary objectives of "Advanced Software Engineering" are to prepare students for real-world team software development. To achieve this goal, we adopt the PBL approach, in which a semester-long and class-wide project is assigned to student as stimulus to self-learning and collaborative learning. Because many PBL activities are conducted outside the classroom, a learning management system (LMS) based on Moodle [7] is implemented to support blended learning [8] environment (face-to-face learning plus E-Learning). The screen shot of the LMS is shown in Fig. 1.

### B. Teaching Strategy

A majority of PBL approach divide students into groups of size three to four [2], [3], [4]. As a required course, the number of students enrolled in 2008-2009 academic year was 237, which were grouped into 61 teams. For PBL and group work, such a high student-instructor ratio is a big obstacle to success of the strategy. So six teacher assistants were employed to help the instructor. Each team is guided by a teacher assistant. The role of the teacher assistants was mainly to perform the following activities:

978-0-7695-3987-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ETCS.2010.339


Fig. 1. The main page of the learning management system

- Acting as inspector of the group,
- Making comments on artifacts,
- Doing acceptance testing.

The instructor acted as a facilitator or guider. Three projects were chosen as stimulus. The project topics included a teaching affair management system, a dormitory management system and a library management system. All of these projects were asked to be integrated to each other. Therefore, the intrateam communication was encouraged. It can also help to grade teams, because the most integrated product may come from the best team.

### C. *Project Schedule and Performances*

On the first day of class, three project concepts are introduced immediately. Project descriptions were provided to the students, and each team had a week to select which project it wanted for the semester.

The students were asked to follow the UP and develop the project in an iterative and incremental way. The UP describes the tasks that students must accomplish to build software, and gives the instructor visibility into the project. The instructor can query students on their progress down to the granularity of the process description, and students follow the rules outlined in the UP to report status to the instructor.

The deadlines are also determined at the first class. The first three phases of UP (inception, elaboration, and construction) are asked for during a 16-week semester. The instructors divided the semester into five iterations, each of which consists of three weeks. Both elaboration and construction phases are composed of two iterations. Inception phase consists of a single iteration. At the end of each iteration, some performances must be handed out and be evaluated by the instructor. There are listed in Table I.

TABLE I. PERFORMANCES DONE AT THE END OF ITERATIONS

| ***Performance*** | ***Description*** |
|---|---|
| **Deliverable:** | |
| D1: Team Description | Presentation of the team and team roles |
| D2: Phase Plan | Phase plan and important milestones |
| D3: Iteration Plans | One for each iteration |
| D4: Weekly Reports | Project status and progress |
| D5: Vision | Project's scope |
| D6: Use-Case Model | Functional requirements |
| D7: Domain Model | Conceptual model |
| D8: UI Design | UI prototype |
| D9: Design Model | Static & dynamic model |
| D10: Implementation | Code and unit test |
| D11: Test Cases | Black-box test cases |
| D12: Final Report | Project retrospect and peer evaluation |
| **Presentations:** | |
| P1: Inception | Project scope; Initial project plan |
| P2: Elaboration | Analysis & Design; UI prototype |
| P3: Construction | Project retrospective; prototype |
| **Two prototypes:** | |
| T1: UI prototype | After elaboration phase |
| T2: Functional Prototype | After construction phase |

### D. *Tracking Progress*

The progress of the students was observed in the weekly meetings of teams with their teacher assistant. in addition teacher assistants' constantly stayed in touch directly with the team leaders. Each member in the team was required to submit weekly progress reports to their team leader and teacher assistant respectively. The leader of each team, besides his/her weekly progress report, was asked to submit another weekly report describing the work done by the individual members

and the overall progress of the team. Teacher assistants are responsible for checking the consistency of these reports. Furthermore, during open hours, students discussed with the

TABLE II. EVALUATION

| | | Team A | | Team B | |
|---|---|---|---|---|---|
| | weight | mark | credit | mark | credit |
| **Performance:** | | | | | |
| D1: Team Description | 1 | 4 | 4 | 5 | 5 |
| D2: Phase Plan | 6 | 3.5 | 21 | 4 | 24 |
| D3: Iteration Plans | 15 | 3 | 45 | 4 | 60 |
| D4: Weekly Reports | 15 | 3.5 | 52.5 | 4 | 60 |
| D5: Vision | 1 | 4 | 4 | 4 | 4 |
| D6: Use-Case Model | 15 | 4 | 60 | 3 | 45 |
| D7: Domain Model | 6 | 4 | 24 | 3 | 18 |
| D8: UI Design | 20 | 4 | 80 | 5 | 100 |
| D9: Design Model | 30 | 4 | 120 | 3.5 | 105 |
| D10: Implementation | 40 | 4 | 160 | 4 | 160 |
| D11: Test Cases | 20 | 3 | 60 | 4 | 80 |
| D12: Final Report | 10 | 4 | 40 | 4 | 40 |
| P1: Inception | 1 | 4 | 4 | 4 | 4 |
| P2: Elaboration & Design | 1 | 4 | 4 | 4 | 4 |
| P3: Construction | 1 | 4 | 4 | 4 | 4 |
| T1: UI prototype | 20 | 4 | 80 | 4.5 | 90 |
| T2: Functional Prototype | 30 | 4 | 120 | 4 | 120 |
| **Audits:** | | | | | |
| Integration | 30 | 3.5 | 105 | 4 | 120 |
| Artifact Traceability | 20 | 4 | 80 | 5 | 100 |
| Missed deadlines | -2 | 1 | -2 | 0 | 0 |
| Warning | -1 | 0 | 0 | 0 | 0 |
| Bonuses | 1 | 0 | 0 | 0 | 0 |
| Total 282 1065.5 1143 Individual (average) | 282 | | 1065.5 266.38 | | 1143 228.6 |

instructor the difficulties that they faced along the project, and the instructor gave some advice on what should be learned and how to learn it.

### *E. Grading*

A credit system was introduced to differentiate the grades of students. All credits are given to teams - not individuals. A team earns a certain amount of credits for each ”performance“ in the course. Table II is an example of credits earned by team a and team b. The actual amount of credits for a particular performance is a product of its weight and the quality rating given by the instructor and teacher assistants. At the end of the course, each team proposed a credit distribution for its members, based on their contributions to the project. For teacher assistants have the sufficient information about the students in his/her group, they can adjust the credit distribution if there are complains in the team. The credit system works quite well in 2008-2009 academic year. Most students think the credit system is fair.

## III. EVALUATION

At the end of the semester, 57 of 61 teams created complete deliverables by specified due dates. Students were able to learn from one another and from their mistakes. Soft skills of personal communication and collaboration were practiced.

The students were asked to complete a survey to find out to what extent the students accept the PBL approach. The students were asked to express their opinion on 1-5 Likert scale, checking 5 if they strongly agree, 4 if they agree, 3 if they neither agree nor disagree, 2 if they disagree, and 1 if they strongly disagree with the statement.

The survey was anonymous and voluntary, and it was conducted using the Moodle LMS. The questionnaire was divided into four parts:

- Questions about course relevance;
- Questions about course interactivity;
- Questions about tutor support;
- Questions about peer support;

207 of 253 enrolled students have completed the questionnaire. The survey results listed in Table III are encouraging for the instructors. For the first group of questions, the students value the relevance of the course to their professional practice at 3.72 and 4.28, respectively. It is clear that students consider what they learned in the course will do well to their professional career.

The second group was centered on the course interactivity. The results of this second group of response indicate that students are satisfied with the blended learning environment, based on a combination of face-to-face environment and online learning via a LMS.

The third and fourth group of questions was related to the tutor and peer support. The relative low mark in group four can be explained by the fact that there is competition in marks exist among students, and peer support is not as active as be imagined to be..

On the other hand, the PBL students in 2008-2009 academic years have been more motivated, and have work harder than those in 2007-2008 academic years. The fail rate of 2008-2009 academic years is lower than that of 2007-2008 academic years.

## IV. CONCLUSIONS

It is challenging to develop a course based on PBL. One of the challenges encountered was selecting a course project student that could be completed in one semester. The students' level of programming ability varied. While some students complained that the project is too simple, some others were unconfident that they can get through the project.

Another challenge was knowing the status and progress of each group. Many team activities and interactions among team member happen outside the classroom. Because of peer pressure in the class, the students tend to hide problems in their weekly reports. It is also difficult to tell from the early deliverables such as vision, use case model and domain model, because total disconnection to implementation can sometimes be mistaken as lack of details, which is a technical problem, not an indication of a dysfunctional group. Moreover, even though the students are asked to start programming early, one single student in a team may sometimes take over early-stage team assignments and thus mask team problems.

PBL is a new approach for teaching undergraduate software engineering principles as well as collaborative skills.

The students were taught to collaborate in team roles, and then the teams must collaborate with other teams to accomplish the project successfully. This approach to education has a closer connection to industry development environments. There is a true integration of individual-level and team-level approaches.

TABLE III. QUESTIONNAIRE RESULTS

| *Question.* | *Avg.* |
|---|---|
| **Relevance:** | |
| My learning focuses on issues that interest me. | 4.02 |
| What I learn is important for my professional practice. | 4.19 |
| I learn how to improve my professional practice. | 4.28 |
| What I learn connects well with my professional practice. | 3.72 |
| **Interactivity:** | |
| I explain my ideas to other students. | 4.27 |
| I ask other students to explain their ideas. | 4.36 |
| Other students ask me to explain my ideas. | 4.22 |
| Other students respond to my ideas. | 4.13 |
| **Tutor Support:** | |
| The tutor stimulates my thinking. | 3.94 |
| The tutor encourages me to participate. | 4.11 |
| The tutor models good discourse. | 4.24 |
| The tutor models critical self-reflection. | 4.31 |
| **Peer Support:** | |
| Other students encourage my participation. | 3.26 |
| Other students praise my contribution. | 3.64 |
| Other students value my contribution. | 3.14 |
| Other students empathize with my struggle to learn. | 2.09 |


## ACKNOWLEDGMENT

This work was supported by the Excellent Course Plan of Xiamen University.


## REFERENCES


[1] O. Barzilay, O. Hazzan, and A. Yehudai, "A multidimensional software engineering course," IEEE Transactions on Education, vol. 52, no. 3, pp. 413–424, Aug. 2009.

[2] R. Garcia-Robles, F. Diaz-del Rio, S. Vicente-Diaz, and A. Linares-Barranco, "An elearning standard approach for supporting PBL in computer engineering," IEEE Transactions on Education, vol. 52, no. 3, pp. 328-339, Aug. 2009.

[3] I. Richardson and Y. Delaney, "Problem based learning in the softtware engineering classroom," in the 22nd Conference on Software Engineering Education and Training, Feb. 2009, pp. 174–181.

[4] N. Linge and D. Parsons, "Problem-based learning as an effective tool for teaching computer network design," IEEE Transactions on Education, vol. 49, no. 1, pp. 5–10, Feb. 2006.

[5] J. H. C. Moust, H. J. M. V. Berkel, and H. G. Schmidt, "Signs of erosion:Reflections on three decades of problem-based learning at maastricht university," Higher Education, vol. 50, no. 4, pp. 665–683, 2005.

[6] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process. Addison Wesley, 1999.

[7] "Moodle," http://moodle.org, 2008.

[8] N. Hoic-Bozic, V. Mornar, and I. Boticki, "A blended learning approach to course design and implementation," IEEE Transactions on Education, vol. 52, no. 1, pp. 19–30, Feb. 2009.