Asian Journal of Information Technology 6 (8): 866-869, 2007
ISSN: 1682-3915
© Medwell Journals, 2007

Adopting Project-Based Learning (PBL) as an Alternative Method to Teach Software Engineering

F.A. Egbokhare and S.C. Chiemeke
Department of Computer Science, University of Benin, Benin City, Nigeria

Abstract: Software engineering is a capstone course for students in final year Computer Science department at University of Benin. The course is a practicsal course and it aims to educate new generation managers, planners, analysts and programmers in the promise and reality of information system development. One of the authors has been teaching software engineering for three consecutive years at the Computer Science department at University of Benin. In the first two years, the course was taught using the traditional classroom approach. Last academic session, a new approach was tried which combined the project based learning with the classroom teaching. In this study, we present the experience of combining the traditional method with the project-based learning approach on teaching software engineering. The waterfall model was adopted to demonstrate the phases of developing information systems and the students were expected to present their reports at the end of every phase. The best report/presentation in each project was then presented to the class as the case study for that phase. This approach not only rouses the curiosity of students but it also challenges them to develop knowledge and creative skills and abilities that will remain with them throughout their career as computer professionals.

Key words: Software engineering, lecture-based teaching, project-based teaching, content, skills

INTRODUCTION

Software Engineering (SE) is a field in computer science that deals with the application of techniques similar to those used by engineers in the development of complex and quality software systems. It also, forms part of the course curriculum in most high institutions that offer computer science and computer-science related courses such as Information and Communications Technology (ICT). Software Engineering has been taught in University of Benin, Nigeria for a quite long time. In our traditional approach, the course mainly covers the theoretical part and it was taught using traditional lecturing combined with homework, mini-case study and tests. SE is a capstone course for students in final year Computer Science Department. One of the positive learning effects of the course is that it prepares the student to work in a professional environment where Information Systems and Software are designed, developed, implemented and maintained in projects.

Effectively imparting the concepts of software engineering has been a major challenge especially in the University of Benin, Nigeria since the poor performance of students over the last five years spoke volumes about their understanding of the course content. Layman *et al.* (2006) discussed similar problems with teaching software engineering at the undergraduate level using the lecture based approach at the North Carolina University and how adopting an alternative teaching method yielded several benefits both to the students and to the course tutors. This indicates that the challenge of teaching software engineering using the traditional lecture-based approach is a global problem which requires a global solution. This study therefore, discusses how a combination of the traditional lecture-based teaching method was combined with a project-based learning approach to effectively impart the content and skills of software engineering at the undergraduate level.

Nacino-Brown *et al.* (1992) defined teaching as an attempt to help someone acquire or change some skill, attitude, knowledge, ideal or appreciation. A teacher (Moore, 1996) is one who is responsible for someone else's learning and takes pains to see that knowledge is acquired by using varying methods if necessary to bring about learning. The goal of teaching therefore, is to bring about the desired learning in the student. Moore (1996) further described successful teaching as the degree to which the teacher has been able to impart the required knowledge on the students. Wirz (2004) observed that most teachers have come to realize that students receive

Corresponding Author: S.C. Chiemeke, Department of Computer Science, University of Benin, Benin City, Nigeria and process information in different ways. Felder (1996) discussed various learning styles adopted by individual students. While, some students tend to focus on facts, data and algorithms others are more comfortable with theories and mathematical models. Some respond strongly to visual forms of information while others get more from verbal forms, written or spoken explanations. While some prefer to learn interactively others function introspectively and individually. Felder however, stressed that it is indeed a matter of style but that if courses are taught exclusively in a manner that is preferred by students or the teachers alone, students may not develop the mental dexterity needed to reach their potential for achievement in school and as professionals. To achieve a major objective of education which is to help students build their skills, it is the responsibility of teacher to identify the best modes of teaching and learning that helps students' understanding. Sheila Tobias (Felder, 1993) identified some negative features that may spur students to switch from their originally intended course of study to other areas. These negative features include:

- Failure to motivate students to get interested in the course
- Relegating students to a passive role in learning
- Emphasizing competition for grades instead of cooperation in learning and
- Concentrating on step-by-step solving rather than conceptual understanding.

Similar, features related to those enumerated above have been observed in teaching the course, software engineering at the University of Benin, Nigeria. This led the course tutors to seek and implement an alternative initiative to restructure the course both in layout and presentation using a combination of traditional lecture-based approach and a project-based learning method. Layman *et al.* (2006) adopted a similar initiative to restructure an undergraduate software engineering course from a lecture-based teaching approach to a lab-oriented course which resulted in some of the highest student evaluations in recent course history.

The Lecture-based teaching method is the traditional and most widely used teaching method especially in post secondary institutions (Nacino *et al.*, 1992; Awotua, 2002). It involves skillful delivery of facts to students according to a preplanned scheme. In its strictest sense, it is characterized by one-way communication in which the teacher presents ideas or concepts, develops and evaluates them and summarizes the main points at the end while the student listens and takes down notes. Students can ask questions during the course of the lecture for clarification. One of the main advantages of this method is that it is used to teach large classes. This and the fact that it is a very suitable method where the teacher has to cover everything in a specified syllabus may have contributed to its popularity. Nacino *et al.* (1992) however, advised that it may not be the most suitable method if the objective is to develop skill. The curriculum of the course software engineering involves teaching the subject matter, ideas and ensuring that students acquire the necessary skills required to solve real-world problems. Thus it was necessary to seek an alternative method to the traditional lecture-based method especially in the area of mastering the required skills.

Project-Based Learning (PBL) is an individual or group activity that goes on over a period of time, resulting in a product, presentation or performance. It is both a teaching and learning method which usually has a time line, milestones and other aspects of formative evaluation as the project proceeds. Donnely and Fitzmaurice (2002) described PBL as both a curriculum and a process. The curriculum consists of carefully selected and designed problems that demand from the learner acquisition of critical knowledge, problem-solving proficiency, self-directed learning strategy and team participation skills. The process replicates the commonly used systemic approach to resolving problems or meeting challenges that are encountered in life and career. Students are engaged in authentic real-world tasks to enhance learning. The model shifts away from short, isolated, teacher-centered lessons to a concept integrated with real world issues and practices. It motivates students to take responsibility for their own work. PBL challenges students to work cooperatively in groups to seek solutions to real world problems (Duch, 1999). These problems are used to engage students' curiosity, think critically and analytically and initiate learning the subject matter. PBL typically begins with an end product or artifact in mind, the production of which requires content knowledge of skills and typically raises one or more problems which students must solve together. The choice of combining PBL with the traditional lecture-based approach was to enable:

- The teacher cover the course curriculum by teaching the content of the phases in the software engineering life cycle model and
- Help Students to develop their skills by carrying out projects on the substance of what is taught.

PBL has successfully been applied as an alternative approach to teaching construction engineering and management (Chinowsky *et al.*, 2006), which is also a professional course. This led to the belief that for a career -base course like software engineering, it will positively influence students understanding.

MATERIALS AND METHODS

We used the Waterfall model which treats the entire software development process as a series of cascading phases (Somerville, 1997; Pfleeger, 2001; Pressman, 2001) to teach the students the concept of software engineering. Waterfall model is the oldest and still the most widely used model in an educational environment. It is of interest to note that in a typical undergraduate software engineering course at the University of Benin, the waterfall model is still the dominant model used to illustrate the software engineering projects. It provides a template into which methods of analysis, design, coding, testing and maintenance can be placed.

RESULTS

In 2004/2005 academic session, the final year students offering software engineering were 150. The students were shared into groups (10 in each group) and each appointed a group leader to coordinate and properly manage group activities. These groups can be likened to teams in real world software engineering projects. The aim of these groups was to teach students to work in teams and also to report and be responsible to a group leader. At the beginning of the course, a case study was given to develop an information management system for the University based on the phases of the waterfall model. Since the course was a full semester course (3-month period), we dedicated one week to teach each phase and a week for the class project which should be prepared using proper software engineering paradigms and tools. At the end of each phase the reports expected from the groups are shown in Table 1:

Table 1: Expected report from students

Phase	Expected report	Comments
Requirement Definition and Analysis	Requirement document or prototype stating the users functional and domain requirements.	-
Design	Architectural and detailed design using appropriate tools.	-
Implementation	Description of hardware, software, database requirements and justification for the choice of implementation language used.	-
Testing	Range of data used must be specified, verification and validation tests must be discussed and related to specified users requirements.	-
Maintenance	-	Since it is a continuous process, the students were asked to demonstrate the different types of maintenance, stating their advantages and disadvantages

Various milestones were set for delivery of the reports to the lecturer's mail box, after which we had general discussions were the various groups presented their work to share any experiences that may benefit the entire class. The best report/presentation in each project was then presented to the class as the case study for that phase.

DISCUSSION

Using the combined lecture and project based teaching methods enabled students to interact with the real world while still in training. This increased students' curiosity and meeting project set milestones poised a major challenge that forced students to create and become creative. The course became more interesting because students looked forward to practically implementing each phase and each group wanted to top the class thus students were forced to become knowledge managers rather than the traditional lecture based method where the lecturer was solely in charge. A general assessment of student performance at the end of the course showed that in a class of one hundred and fifty students, 126 (84%) passed the course with good grades, 15 (10%) obtained average scores while only 9 (6%) failed the course as shown in Table 2.

These grades showed remarkable improvement on student performance in the course compared to the previous year were over 50% of the students obtained average scores. The students' assessment of the combined approach is as shown in Table 3.

Table 3 shows that 90% (135) of the students understood the course content better with the combined approach. Students had challenging projects to look forward to and this roused their curiosity and creative abilities. Also, we discovered that there was high motivation to learn 86% (129). This indicates that students had high expectations for the course and to this end they were quite confident and ready for the real-world challenges in software engineering.

Table 2: Grades obtained by student in 2004/2005 academic session

Grades	(%)
A	10
B	116
C	15
D	9

Table 3: Student's assessment of the combined teaching approach

Item	Yes	No
Understanding of course content	135	15
Mental challenge	117	33
Creativity and originality	96	54
Readiness for the labour market	141	9
Team experience	75	75
Motivation	129	21

CONCLUSION

Applying project-based learning and the traditional lecture-based approach in teaching software engineering has proven to be effective because it allows students to experience first hand challenges and opportunities from real-world projects. The approach provides choice for students to focus their learning experience on a particular area of interest. However, Software engineering is a practical course and it aims to educate new generation managers, planners, analysts and programmers in the promise and reality of information system development. The best way to teach the course is to combine theoretical teaching methods with practical learning methods. This approach not only rouses the curiosity of students but it also challenges them to develop knowledge and creative skills and abilities that will remain with them throughout their career as computer professionals.

REFERENCES

Awotua-Efebo, E.B., 2002. Effective Teaching: Principles and Practice Paragraphics. Port Harcourt, Nigeria, pp: 159-210.

Chinowsky, P.S., H. Brown, A. Szajnman and A. Realph, 2006. Developing Knowledge Landscapes through Project-Based Learning. J. Professional Issues in Eng. Edu. Practice, 132: 118-124.

Donnelly, R. and M. Fitzmaurice, 2002. Collaborative Project-Based Learning and Problem-Based Learning in Higher Education: A Consideration of Tutor and Student Roles in Learner-Focused Strategies. Learning and Teaching Center, Dublin Institute of Technology, Dublin 2, Ireland.

Duch, B., 1999. Problem-Based Learning. Retrieved 31st May, 2007 from http://www.udel.edu/pbl/

Felder Richard, M., 1996. Matters of Style. ASEE Prism, 6: 18-23.

Hailiday, J., 1996. Back to Good Teaching. Diversity within Tradition. Cassell. London.

Layman, L., T. Cornwell and L. Williams, 2006. Personality Types, Learning styles and an agile approach to software Engineering Education. ACM. SIGCSE. Bull., 38: 428-432.

Moore, T.W., 1996. The Phylosophy of Education. An Introduction. Routledge and Kegan Paul. London, pp: 66-89.

Nacino-Brown, R., F.E. Oke and D.P. Brown, 1992. Curriculum and Instruction: An Introduction to Methods of Teaching. Macmillan Press Ltd, London and Basingstoke.

Pfleeger, S.L., 2001. Software Engineering: Theory and Practice. Pearson Edu. North Asia, pp: 659.

Pressman, R.S., 2001. Software Engineering: A Practitioner's Approach, (5th Edn.), McGraw Hill, pp: 855.

Richard, F.M., 1993. Reaching the Second Tier. Learning and Teaching Styles in College Science Education. J. College Sci. Teaching, 23: 286-290.

Sommerville, I., 1997. Software Engineering. (5th Edn.), Addison-Wesley, pp: 742.

Wirz Dick, 2004. Student's Learning Styles Vs. Professors' Teaching Styles. Inquiry, Vol. 9.