Enhancing Learning Outcomes in Software Engineering course through Problem Based Learning and Peer Assisted Learning

Hannabel Hyacinth Alva [1], Uma B. [2], Shruthi D.V. [3], Saroja C. [4]

[1]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India
[2]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India
[3]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India
[4]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India

[1]hha@mcehassan.ac.in
[2]bu@mcehassan.ac.in
[3]dvs@mcehassan.ac.in
[4]cs@mcehassan.ac.in

Abstract Learning outcomes are brief statements to describe the significance and fundamental learning achieved by learners, which can be demonstrated at the end of a course. Several researchers of Outcomes Based Education (OBE) suggest that the ability to demonstrate learning is a key element of higher education, which involves a performance of some kind in order to show learning that matters. Knowledge gained must be evident through a demonstration process of some kind depending on the course content. Learning outcomes of Software Engineering course fundamentally require learners to value various engineering issues that contribute to developing an evolving software-intensive system. It is necessary to plan and deliver the required system through efficient software engineering processes, based on knowledge of software development models. In addition, group working skills, inter-group negotiations, planning and time management are also required. This paper outlines the use of Problem-Based Learning (PBL) and Peer Assisted Learning (PAL) techniques that were incorporated in delivering the course on Software Engineering to fourth semester students to enhance the learning outcomes.

Keywords- Learning Outcomes, Problem-Based Learning, Peer Assisted Learning

1. Introduction

There is a growing demand for evaluating the knowledge acquired by undergraduates in engineering institutions through a well defined method of OBE.

Hannabel Hyacinth Alva
Information Science and Engineering, Malnad College of Engineering, Hassan
hha@ mcehassan.ac.in

The areas that contribute to an overall professional development include curriculum design, pedagogical and assessment techniques with a focus on OBE. In response to this demand, the delivery of the course Software Engineering requires the course faculty to keep in mind the learning domains of the students. Through a well balanced mode of content delivery, the cognitive and meta-cognitive learning can be achieved. To enhance learning, peer learning and self-learning provide a good impetus in this direction. The course on Software Engineering is important for students of Computer Science Engineering and Information Science Engineering particularly, as it introduces them to the various processes and practices of software development. The conventional approach of delivery of this course as a series of lecture sessions may not make the learning effective, as it lacks experiencing the activities involved in developing a software system. Also the course faculty may not have an exposure to industry practices, and the examples quoted in the text books may be unfamiliar to the learner. Thus the assessment may restrict to mainly focusing on level 1 of the Blooms Taxonomy.

There was a strong consensus from the graduate exit interviews and feedback from alumni and industry experts that the Software Engineering course needs to be delivered in the early semesters of the programme rather than the later semesters. The course was shifted to the fourth semester from the sixth semester. Keeping in mind the industry requirement and the feedback received, the course on Software Engineering incorporated in addition to regular lecture sessions, Problem Based Learning technique, Peer Assisted Learning and awareness of industry practices. This would accomplish the real attempt of applying the principles of project management and team roles that need to be demonstrated during software system design. At the end of each problem based

activity, the team members need to share their experiences and knowledge gained through a short presentation. This helps peer students to improve their understanding of Software engineering concepts and also enhance their confidence in software system design.

2. Literature Review

Software Engineering is the study related to all aspects of software development from design, development to maintenance of software. Software Engineering is required as part of the engineering curriculum especially for computer science stream, to address all the steps required to develop the proposed software application. The general problems that arise when a software is developed are: exceeding timelines and budgets, and decreasing level of software quality. It is required from every member of the software development team that the application is built as per requirements, and is delivered on time within budget.

Software Engineering (SE) is an area with a wide range of concepts and knowledge. Such diversity of topics, requires the application of different teaching and learning techniques for an effective education. Serious Games is one of such techniques, yet its design tends to be complex, currently lacking a map of game design standards that comply with SE education requirements [1]. A number of principles have emerged from recent software engineering researches that have the potential to bring a measure of control to the practice of this discipline. The classroom is a good place to start acquainting students with these principles, and to start getting them into the habit of adhering them as a matter of routine practice [2].

A lot of effort is put in preparing students for the software industry. Employers are usually unsatisfied with the experience and knowledge that recent graduates possess. One way to bridge this gap is to develop a method that includes not only teaching software engineering theory but also allows students to practice the principles in a simulated industry environment and develop skills useful to the discipline [3].

Peer-assisted learning is one of many educational strategies that can assist students' learning and their professional development. Its value could be enhanced if combined with a student centred learning strategy such as team-based learning. It can also provide students with teaching experience which could help their professional development [4].

Contemporary higher education is actively introducing learning technologies, that put the student at the centre of the educational process, and technologies taking into account the learning needs of students and creating conditions for their development. Problem-based and project learning are those technologies, which potential has not yet been fully employed by higher education [5].

Many approaches have been proposed in order to foster OBE through an enriching learning experience in Software Engineering course in addition to preparing employable graduates. On one hand, it is necessary to ensure that the course provides the knowledge required for carrying out all activities of software development and on the other to assure that the knowledge gained is delivered in a manner such that the future professionals will be able to correctly address problems during their professional career.

3. Need for a shift from conventional delivery of SE Course

The emphasis of the Software Engineering course is on gaining knowledge to be able to analyse, model, design, implement and evaluate a simple software system effectively. Software engineering is a practical subject and requires applying the concepts learnt as part of the curriculum. It is generally put to practice by a team of computer system specialists, when developing the solution to a software problem, by carrying out the various software development processes learnt during the study of the course. Teams consist of peers and are expected to systematize their work, and learn skills needed to organise work.

The development of a software solution requires a good combination of diverse skills. Programming skills are vital and to augment it, others include, analytical skills, ability to design a good GUI, testing skills, documentation, team spirit and commitment. Each team member is assigned a specific role based on the expertise of the individual. Since every software project is a team activity, at the completion of a project, each team member will gain an improved experience in all aspects of the project. Hence the breadth of knowledge in many areas grows and the depth of knowledge in some areas is mastered. Such a skill set, is of high demand in the job market, and would certainly enhance employability.

4. Alignment of SE course to practice

For a course such as Software Engineering to be successful at the fourth semester level, it is necessary to focus on an approach where in the learner is required to solve a problem, and to be successful, it requires students to have the fundamental skills necessary to solve the given problem. Hence the course was delivered using a combination of Problem Based Learning, Peer Assisted Learning, Self Learning and Industry Interaction. The basis for considering these methods was to nature interpersonal skills, personal skills, technical skills and organizational skills.

A. Problem Based Learning

One of the main objectives of engineering education is to prepare an employable workforce. The problems that

students generally solve in the classroom are different from those at a typical workplace. Also the procedure followed by industry in designing a solution to a problem is well structured. An attempt was made towards making classroom problems emulate real-world workplace problems in order to better prepare graduates. Through problem solving exercises, students will also be able to address projects better.

Each problem solving exercise was a team work and careful selection of team members was necessary to help build on a successful design experience. Team of four members each were formed during the first week of the semester considering both fast learners and slow learners. A team leader was identified with a simple group discussion technique. The teams were assigned simple real world problems. One such example was to develop a "District Digital Library" application to cater to the needs of potential users. Another example considered was the typical "ATM system". These problems were simple to address, and enabled the teams to understand and analyse the problem, generate scenarios, and produce documentation in addition to working as a team. As part of problem solving, the teams were also introduced to the concept of milestones, deliverables, the risk of schedule slippage and the critical path. Students were also exposed to Industry practices followed in collecting requirements for the software to be developed, which is the vital success factor for all software.

To enhance understanding, students were also required to consider developing solutions to problems related with the tourism industry and the transportation sector. These enabled students to be engaged problem solvers, understand and analyse the problem considering the conditions required for a good solution and in the process become self directed learners. The teams were asked to focus on requirements specification and object-oriented system design. This enabled the students to understand the importance of using specific terms while framing the functional and non-functional requirements and to learn the use of various Unified Modelling Language (UML) notations. These problems also reinforced the concepts associated with Agile development technique and introduced learners to terms such as sprints, sprint planning, sprint review, estimation, taskboard. The importance of practising ethics was also highlighted to bring about awareness of intellectual property such as patents and copyright. Each group documented their work and the activities culminated with each of the teams presenting their case study to the rest of the class.

It was observed during these exercises, the teams enhanced their understanding of Software Engineering concepts, in addition to improving on their presentation and communication skills. The application of knowledge gained to software problems arising in the real world, helped strengthen the Learn– practice- improve cycle.

B. **Peer Assisted Learning**

Informal peer education has occurred at all times. Peer Assisted Learning (PAL) is characterised by specific role-taking: one person has the job of tutor, where as others are in the role of tutees. It strongly focuses on curriculum content. It is important to note that, those helping their peers to learn belong to similar social groupings as that of the learners, and do not possess extensive knowledge of subject matter. Researchers term this experience as "social congruence". The PAL motivation highlights the fact that students, whether tutors or tutees, share the similar knowledge base and learning experience. This helps the peer tutors to communicate with their learners to explain concepts using a suitable delivery mechanism. Additionally, learners feel at ease with peer teachers as they belong to similar social roles. Benefits of PAL have been reported in the field of medical education. Other significant observations are, PAL also lightens the teaching burden on the faculty and provides role models to other students. Research also suggests that participating in PAL promotes leadership skills, enhances motivation and confidence, and brings an interest in academic careers also.

The PAL technique considered was a 15 minute activity implemented once a week. As part of Peer Assisted Learning, four credible peers played the role of tutor and reinforced the concepts of requirements analysis and a few key concepts of Object-Oriented design using UML with their peers. The rationale for selecting this part from the curriculum was to enable peers to be aware of the various UML notations that are part of system design, which is an intrinsic part of all projects carried out by students as part of the laboratory courses in higher semesters. The use of scenarios during design process was very well taken and exhibited by the students through a few case studies

In addition to peer learning, a self learning exercise on the topic of International Organization for Standardization was specified. The aim was to understand the basic concept of standards, how the various standards are developed and their benefits.

C. **Experience of Software Development by Alumni**

Industry standards constantly evolve within limits to meet new challenges. These may be specific to an organization or a developer community. Hence it is necessary for Industry personnel to adapt best practices for each situation. In software development, a best practice is a clearly defined method that contributes to a successful software product. In the entire software industry, several best practices are followed and one commonly followed is the Agile project management methodology. To be abreast with current practices, processes and frameworks in Industry, is of vital importance for prospective employees. The software industry in particular requires a

team of individuals who are positive in their approach when addressing problems.

Alumni are one of the important assets of an Educational Institution. They represent the alma mater to the world. To help undergraduates be aware of Industry practices, it is easy and convenient to utilize alumni experiences and knowledge gained through employment in Industry. The living, breathing experiences of alumni not only reinforce concepts learned as part of the course, but also inspire the learners to practice them well.

To be aware of industry practices, two alumni offered to contribute towards enhancing student learning of the course. The alumni were engaged in different domains of the software industry and were involved in software development process activities for over six years and shared the current practices followed in the industry for software design and development. An emphasis was made on Agile development as this was the norm in industry. The alumni shared some of their experiences starting from client interactions as part of requirements elicitation through testing and documentation. They made aware of the probable hiccups during analysis and advised the learners to be aware of best practices. Through this exercises, it also helped strengthen the relationship between the institution and its alumni.

5. Outcomes

Through Problem Based Learning, students actively learned alongside lecture sessions. The case studies considered as part of PBL related to the real world and this enabled better understanding of software engineering concepts, as they learnt by practicing the tasks themselves. The students were able to analyse the given problems and model the structure of software system and express clearly the functional and non-functional requirements and use UML notations at the end of the course.

To determine the course experience and the effectiveness of learning, students were required to complete an online questionnaire on completion of the course (Fig.1).

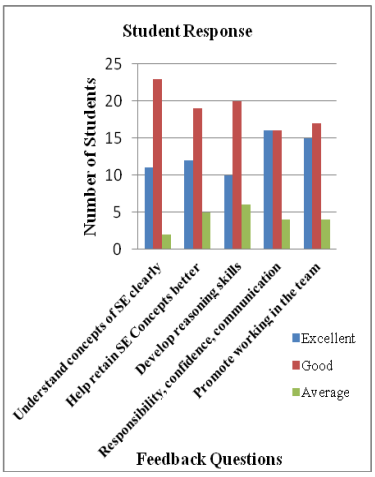

Fig.1: Student Responses

The feedback obtained from the students clearly indicated that rather than just attending lecture sessions, problems helped them to retain the information in a meaningful and lasting way. Notable developments in competences such as responsibility, confidence and communication skills of individual students were also observed over the semester. Students collaboratively worked in teams using software engineering approaches and communicated successfully as effective team members contributing in innovative design solutions. They also prepared technical documentations and made presentations of the software development case studies, considering aspects such as planning and scheduling. The students were also able to appreciate the larger engineering issues that are the basis for developing an evolving software-intensive system. This approach was also appreciated by colleagues of the department, and from faculty of other colleges.

6. Conclusions

The main goal of engineering education is to prepare students for industry. In order to better prepare graduates for any industry, it is required to reduce the gap between classroom and workplace problems considered during the undergraduate course. By incorporating PBL and PAL into regular lecture sessions as part of the Software engineering course, at the end of the semester, all the students exhibited an enriched learning experience. On comparison with the learning outcomes of the earlier batch of students where the delivery mechanism was limited to lecture sessions predominantly, the learning outcome of these students increased significantly. The students also performed remarkably well in carrying out mini projects as part of laboratory exercises. They were able to analyse engineering problems better and design significantly well structured solutions. They functioned effectively in teams and made effective presentations.

References

1. Nuno H. Flores, Ana C.R. Paiva, Pedro Letra (2016) Software Engineering Management Education through Game Design Patterns, 2nd International Conference on Higher Education Advances, Social and Behavioural Sciences, 436 – 442.
2. Fairouz Tchier, Latifa Ben Arfa Rabai, Ali Mili (2015) Putting Engineering into Software Engineering: Upholding Software Engineering Principles in the Classroom, Computers in Human Behavior, 48, 245-254.
3. Aldo Dagnino (2014) Increasing the Effectiveness of Teaching Software Engineering: A University and Industry Partnership, Conference on Software Engineering Education and Training (CSEE&T).
4. Sausan Al Kawasa, Hossam Hamdy (2017) Peer-assisted Learning Associated with Team-based Learning in Dental Education, Health Professions Education, 3, 38-43.
5. Turlo Yevgeny Mikhailovich, Alyabieva Alexander Yurievna (2016) Application of Problem-Based Learning Technology in Technical Education, 11th International Forum on Strategic Technology: Humanitarian and Economical Aspects of Engineering Sciences (IFOST).
6. Nuha H. El-Khalili (2013) Teaching Agile Software Engineering Using Problem-Based Learning, Information and Communication Technology Education, 9(3), 1-12.
7. Oscar Pedreria, Felix Garcia, Nieves Brisaboa, Mario Piattini (2015) Gamification in Software Engineering – A systematic Mapping, Information and Software Technology, 57, 157-168.
8. Izzat Alsmadi and Bilal Abul-Huda (2011) Improving understandability in teaching of Software Engineering and Connectivity with the Industry, EDUCON–Learning Environments and Ecosystems in Engineering Education.
9. Fadilah Puteha, Maniam Kaliannanb, Nafis Alamc (2015) Learning for professional development via peers: A System Theory approach, Social and Behavioral Sciences, 172, 88 – 95.
10. Ana M. Moreno, Maria-Isabel Sanchez-Segura, Fuesanta Medina-Dominguez, Laura Caevajal (2012) Balancing Software Engineering Education and Industrial Needs, System and Software, 85, 1607-1620.