Activity Based Teaching Learning in Software Engineering – An Experience

Padmashree Desai#, G.H. Joshi*

#Department Of Computer Science and Engineering
B.V.Bhoomaraddi College of Engineering and Technology, Hubli-580031
padmashri@bvb.edu

*Department Of Computer Science and Engineering
B.V.Bhoomaraddi College of Engineering and Technology, Hubli-580031
ghjoshi@bvb.edu

Abstract-The course on Software Engineering is important for both the undergraduate and postgraduate programmes of computer science. This is particularly so in terms of making the students industry ready by exposing them to the processes and practices of life cycle activities of software development. The traditional approach to teaching the course on Software Engineering has not been able to make a positive impact on learning because: the course instructors mostly lack exposure to industry practices, the course material being referred while teaching the course fails to establish the context as the case studies referred are alien to the students and assessment focusing mainly on memory oriented questions. Thus teaching the course on Software Engineering has got reduced to monotonous lecturing, in the absence of experience and case studies. An attempt to address the issues referred above is made by the authors. The theory course on Software Engineering was redesigned by tightly coupling a Mini project course with it. Different activities are designed along with Class room teaching to make Software Engineering as joyful course for the students. The proposed methodology helped the students to improve their Software Engineering concepts and quality of capstone projects. The paper discusses the approach followed, the results obtained and the experience gained.


I. INTRODUCTION

The Software Engineering (SE) is an important course which is taught for the students of Computer Science and Engineering and equivalent branches at Bachelors and Masters Levels. Usually this subject consists of several classroom sessions, quizzes, assignments, tests etc. Theoretical concepts are present in class room. However one can observe that this subject is more of lectures and less of interactive learning as there is a lack of laboratory sessions or mini projects. Faculty feel teaching Software Engineering easy by referring explanation on the various lifecycle models and phases of software development such as Planning, Requirements, Design, Testing, Validation, and Maintenance etc. But the observation is that the concepts cannot be illustrated easily on the blackboard as it can be learnt through experience. The problems with classroom lectures is that the concepts are not very well illustrated and mostly it becomes monotonous with lecturer speaking for most of the time. Lack of process visualization and there is no hands on experience in the lifecycle activities. The students tend to by heart the contents and seem to give less importance for the subject as it is more verbose and less mathematical or algorithmic. Also, one can observe higher failure rates among students as there is a general tendency to miss important concepts and misunderstand the topics. So teaching software engineering cannot be only achieved through Class room. Major challenge [1]-[2] is to integrate methodology and theory into the practice of software development. Instead of learning methodologies as abstract ideas, Students should be given an opportunity for applying concepts into the implementation of their projects.

These issues are addressed by redesigning the curriculum, adopting teaching learning activities and redesigning Assessment methodology inline with the course objectives. Outcome Based Education (OBE) frame work [3]-[4] is used for curriculum design and implementation.

The proposed method uses multipronged strategy to make the subject more interactive and involvement of the students in the life cycle activities of the software development by integrating Software Engineering theory course with mini projects, interactive sessions in class room, problem solving sessions, games and role plays, quizzes, Use of SE tools and relevant case studies to provide practical knowledge on Software Engineering theoretical concepts.

The paper is organized into following sections. Section I discusses Curriculum design and implementation, Section II describes Teaching- learning activities, Section III narrates Assessment Methods, Section IV tells about impacts of methodology on students and Section V discusses the Observation and Conclusion.
II. CURRICULUM DESIGN AND IMPLEMENTATION

The Theory course is redesigned for Fifth Semester Undergraduate Engineering Programme. Curriculum design and implementation framework is shown in Fig. 1.

![Curriculum Design and Implementation Framework](image)

III. CURRICULUM CONTENT

Curriculum contents are designed by writing Lesson plan well in advance and given to the students in the beginning of the semester. Lesson plan consists of Course contents, Course learning objectives (CLOs) and Topic wise objectives for each chapter. CLOs are written using Blooms Taxonomy meeting to abet a-k criteria [5]. At the end of the course/chapter student should be able to achieve the written objectives. CLOs and Chapter wise learning objectives are shown below.

**Course Learning Objectives (CLOs):**

At the end of the course student should be able to:

- **Explain** the different processes and models used for software development.
- **Identify** the requirements of customer problem through customer interaction and case study.
- **Analyze** the customer requirements using different software Engineering techniques and Prepare the Software Requirement Specification document (SRS).
- **Write** pseudo algorithm / flowchart which helps in building the software.
- **Test** the different modules and System using the Test plan.
- **Develop** a solution for the customer problem.

IV. TEACHING – LEARNING ACTIVITIES

An Innovative approach to SE consists of following activities shown in Fig 2 which are incorporated in the theory course. Fig 2 shows that SE is tightly coupled with Mini projects, Active learning pedagogy tools are used in the class room and Workshops conducted by industry experts to provide real time practices used in the industry.

![Teaching-Learning Activities](image)

Chapter wise Learning objectives:
At the end of the chapter student should be able to:

Chapter Title: Software Testing Tactics
- **List** different testing tactics (L1)
- **Distinguish** between Black-Box testing and White-Box testing (L2)
- **Write** Cyclomatic complexity for a given flow graph (L6)
- **Explain** object oriented testing methods (L2)
  - **Apply** appropriate testing method for the Problem (L3)
- **Design** set of test cases using different testing methods for a given problem (L6)

L1-L6 indicates the levels of blooms taxonomy.
A. Mini Projects

Teaching Software Development Cycle (SDLC) in a class room is a difficult task as visualization of different phases of process is difficult. Students learn the concepts if they experience and practice it. So Software Engineering theory course is tightly coupled with the Mini project course [6]. The focus of the mini project was not on what the students were doing; instead it was on how they were doing. Syllabus of Software Engineering course consists of software processes and models, Software requirement specification, Software design, Detailed design and Testing. Teaching these topics in the theory class is easy but visualization of the process in the classes is not possible for the students. So author correlated each of the topic in the SE course to Mini project laboratory activities. One can observe that Waterfall model is the best model for the beginners to understand Software Development Life Cycle and follow it. Traditional Waterfall model[7] is used for the project development in the mini project laboratory is shown in Fig. 3.

![Fig. 3 Waterfall model for software development](image)

Mini projects are done in a team. Team strength is 3-4 students. Guides are allocated to student teams. An evaluation team of three faculty members evaluate the projects. Laboratory plan is prepared accordingly for the students to experience the different phases of SDLC. The laboratory plan is prepared for thirteen weeks of semester. Each phase in project plan takes three weeks such that during first week awareness program conducted by faculty makes course expectations clear to the students. During second week of the project plan phase, the students are expected to clarify their doubts with the faculty. During third week, evaluation of the activity is carried out by an evaluation team consisting of three members of the faculty. Laboratory plan is shown in Table I. Evaluation of projects is done as per the assessment rubrics written for each phase activity. The assessment rubrics helped in fair evaluation across the batches. This activity helped the students to understand requirement analysis, writing functional and non functional requirements, preparing test plan, coding and perform testing based on test plan. The mindset of the students that the large program makes a project is removed from their mind.

<table>
<thead>
<tr>
<th>Table I: Laboratory Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Different Phase</strong></td>
</tr>
<tr>
<td>Problem identification</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SRS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Design</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Module Implementation &amp; Testing</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Integration and System Testing</td>
</tr>
</tbody>
</table>

B. SE Tool

Demo of SE tools such as Project Management, designing are done in the mini project Laboratory. Students are encouraged to use these tools in different levels of software development. Project management tool help students to create a project plan. They learn to find the critical path, setting milestones, allocating resources, resource leveling, generating calendars etc. Using Rational Rose[8] students could generate UML diagrams like Use Case diagrams, Class diagrams, etc and synthesize the JAVA or C++ code.

C. Class room activities

Different active learning tools are used in the theory class to encourage students’ involvement in the course study. This helped to overcome the drawback of monotonous teaching and learning.
D. Case Studies

Text book case studies[8] are too lengthy to understand at fifth semester level. Many times course teacher fails to establish the context as the case studies referred are alien to the students. Mainly due to the lack of domain knowledge for students to understand the case studies. Here Author has used case studies which are developed by senior students either as mini projects or capstone projects. These case studies helped to establish the context easily at various stages of teaching theory in the class room. Different scenarios made clear to the students.

And also student teams are asked to visit the nearby industry to collect the information about their production and present that as a case study.

E. Class Room Sessions

Presentation slides are used in the class with interesting examples, analogies from civil engineering and mechanical engineering etc.

F. Games & Role Plays

Interesting games[9] are played in the class room to understand the importance of Customer–Developer Interaction, Requirement gathering, designing. The role play involves students taking on different roles for an activity such as Customer, Developer, and Software tester. This provided them involvement and a mechanism to practice the different activities and roles.

G. Criticizing /analyzing the requirements /design

Incomplete /Wrong requirements are given for students to analyze these and classify the requirements as Functional or Non functional requirement and write the correct requirements. Incomplete/Wrong Design such as Architecture, Data flow diagram, flowchart are given to students to identify the flaws and produce the correct design.

This activity helped to improve their analytical skills.

H. Problem Solving

Different types of problems[10] are solved in the classroom relating to different case studies. Problems based on Testing tactics that is how to write test plan, writing test cases to test different paths of software module, writing test cases to test the behavior of the software are done in the class. Problem statement is given by the Faculty. Students are motivated to understand and solve the problem. Sufficient time is provided to the students to tackle the problem and one of the student will be called to solve the problem on the board. This activity improved their involvement and participation in the class.

I. Workshops

As faculty lack industry experience, workshops[11]-[12] are conducted from industry experts to provide real time implementation details to the students. Industry experts conducted workshop on requirement management and testing methodology. This helped the students to write test cases properly for the modules and implementation of the same. The workshops bridge the gap between Industry and Academics.

V. ASSESSMENT METHODS

Assessment criteria [13]-[14] are defined to meet the stated objectives both in theory and laboratory. Evaluation of mini projects for their continuous involvement in the development process is measured through Assessment rubrics. 50% weightage is given for mini projects and 50% weightage is given for SE theory. Theory question papers are designed to address higher level of blooms taxonomy. Continuous Internal Examination(CIE) is done by conducting regular exams and quizzes/assignments. CIE for SE theory is done as per the Table II.

<table>
<thead>
<tr>
<th>S No.</th>
<th>Assessment Criteria</th>
<th>Weightage in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Minor exams</td>
<td>40%</td>
</tr>
<tr>
<td>2</td>
<td>Quizzes/Assignments</td>
<td>10%</td>
</tr>
</tbody>
</table>

Two minor exams are conducted to find their understanding levels in the subject. Two quizzes and one assignment are conducted to check their aptitude level and mental ability skills in the theory.

VI. RESULTS

At the end of course feedback is taken to understand impact of the approach followed in teaching this course. The feedback is taken for the class strength of 140 students. This curriculum is successfully implemented for two consecutive years 2008-2009 and 2010-2011. The results are really promising. Feedback form is shown in the Table III.
Results of Feedback is shown in the Fig. 4. This indicates that:

- 88% of the students Strongly agree or agree for understanding of the objectives of the course in the beginning.
- 90% of the students strongly agree or agree that Class room activities / Case studies were useful for them to understand the SE concepts.
- 92% of the students strongly agree or agree that Mini project helped to understand the process of Software Engineering and visualize the software Development process.
- 96% of the students strongly agree or agree that Workshops / demos of Software Engineering tools helped to understand Real applications.
- 92% of the students Strongly agree or agree that Activities helped to improve the Communication skills and ability to work in a team.

The approach followed helped to improve their performance in Continuous Internal Exams (CIE) and Semester End Exams (SEE) both in theory and mini project course. Performance of students in CIE is shown in Fig. 5. This indicate that 80% of the students have got marks above 50. Performance in SEE is shown in Fig. 6. This indicate that students passing percentage in Software Engineering theory is 96% and in mini projects results is 99% which is better compared to previous year results.

### VII. CONCLUSION

The innovative approach followed to teach SE has provided good results. Students gained hands on experience on process of SDLC as mini project is tightly coupled with Software Engineering theory course. The students were able to appreciate the class room learning as they were simultaneously realizing the concepts learned in class room through different activities.
such as games and role plays, criticizing /analyzing the requirements /design, demo of SE tools and workshops. Use of case studies which are the project’s of students done in the previous years helped to establish the context easily. Earlier method of monotonous teaching using board-chalk teaching is overcome by activity based learning method.

The process helped the students to improve their capstone projects. This approach encouraged the students to improve their communication skills, team coordination, problem solving and analytical skills. Performance in Semester End Exams is improved. In all the activities, the assessment of student learning focused on their understanding and application more than on memory.

REFERENCES

[4] Chandra R. Sekhar, Omer Farook and Essaid Boutilica,”Continuous Improvement Process Based on Outcome Based Education”, Purdue University Calumet.