A Novel Approach to Carrying Out Mini Project in Computer Science & Engineering

Padmashree Desai[#], G.H. Joshi[*], Vijayalaskhmi M[#]
[#]*Department Of Computer Science and Engineering*
B.V.Bhoomaraddi College of Engineering and Technology, Hubli-5800 31
[1]padmashri@bvb.edu, [3]viju11@bvb.edu

[#]*Department Of Computer Science and Engineering*
B.V.Bhoomaraddi College of Engineering and Technology, Hubli-5800 31
[2]ghjoshi@bvb.edu

***Abstract*— The student projects done in Computer Science and Engineering suffers from the following problems: poorly managed requirements, poor or no design and poor or no testing. Students generally tend to focus more on coding phase of the project, since the misconception is that a large code constitutes good software. Unlike other branches of engineering, the engineering processes are not visible here. Further, management of large number of student projects poses a challenge in terms of guidance, progress monitoring and assessment. The student projects are observed to be ending up incomplete and are rarely deployed. The feedback from industry side also speaks about the gaps in terms of these skills. A mini project course for fifth semester is designed in the undergraduate programme in Computer Science and Engineering to address the lacunae observed above by tightly integrating it with the theory course on Software engineering. The focus of the mini project is not on what the problem definition is, instead it is on how it is done. The implementation plan addressed the issues and challenges listed above faced by the faculty. Assessment rubrics are written to guarantee proper understanding of the expectations among the stake holders resulting in fair assessment.**

This approach improved students understandability in Software engineering concepts and also the quality of their capstone projects. The paper discusses the design, implementation and assessment details of mini project along with the experience gained.

***Keywords*— Assessment, Learning Objectives, Skills, Software engineering, Mini project.**

I. INTRODUCTION

The IEEE definition of engineering is that "Engineering is that profession in which knowledge of the mathematical, computational and natural sciences gained by study, experience, and practice is applied with judgment to develop economically effective use of matter, energy, and information to the benefit of humankind". The project provide student with an opportunity to gain experience in the practical application of what the student has been studying for the past several years. The project helps to understand concept of planning, designing and implementation for an identified problem

So a mini project course is redesigned in early stage of undergraduate programme in Computer Science and Engineering for fifth semester. The activities in the mini project are designed in line with the software engineering life cycle activities studying in the theory course on Software engineering in the same semester. The course is designed to meet the industry needs [11]-[12].The course is designed to overcome the following problems faced by students while carrying out projects:

- Poor requirements management, design and testing.
- Incompletion of projects for not following schedule.
- Lack of team work.
- Poor documentation and communication.
- Mindset of students that large code makes a project.

The paper is organized into following sections. Section I discusses Curriculum design and implementation, Section II describes different phases of software development, Section III narrates Assessment Methods, Section IV tells about impacts of methodology on students and Section V discusses the Observation and Conclusion.

II. COURSE DESIGN AND IMPLEMENTATION

A course on mini project is designed with the following Course Learning Objectives (CLOs) using Blooms taxonomy [1]:

- **apply** knowledge of Computer Science and Engineering to solve an identified problem
- **recognize** the need for engineering a product or solution
- **use** software development life cycle activities in the project development
- **develop** communication skills, technical writing skills
- **ability** to work in a team.

Mini project laboratory activities follow the traditional Software Development Life Cycle (SDLC) activities [2] of Waterfall model. Traditional waterfall model is easy to understand and as there is a clear separation between phases.

Mini project is carried out in different phases. Each phase is clearly defined by objectives and outcomes. Student team is formed with 4 members. Guides are allocated to mentor the teams. 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. Different activities in a phase are shown in Fig. 1. Evaluation of projects is done according to the assessment rubrics.

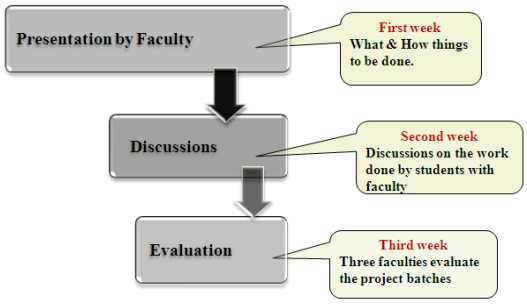

Fig. 1 Activities in each phase

III. Different Phases Of Software Development

A. *Problem identification*

Students are provided with the guide lines to select a problem. Student select a problem, identify features, scope and purpose of the problem. Students give a presentation on this phase and evaluation team evaluates and give suggestion to students whether they can go ahead with problem definition.

B. *Software Requirement Specification (SRS)*

Students analyse the problem and map the features to functional and non functional requirements. Acceptance test plan is prepared accordingly. Students prepare SRS document accordingly.

C. *Software Design*

Usual practice is to prepare a single design and implement the same. But in this approach, sub groups of two members are formed within a team. Alternative designs are prepared by these subgroups. Peer review of these designs is done and final design is prepared. These are two independent designs based on same SRS. These phase helped the students

D. *Module Implementation and Testing*

Modules are implemented and tested using appropriate test cases. Proper coding standards and documentation is insisted during this phase.

E. *Integration and System testing*

All modules are integrated and tested for the test plan written during SRS phase. Demo of Rational Rose and Testing tools are done during project development.
Different phases run as per the lab plan shown in Table I

TABLE I
LABORATORY PLAN

Different Phase	Activity	Week #
Problem identification	Presentation by staff on awareness program	1
	Discussions on Problem statements & finalizing them	2
	Evaluation of process of problem identification.	3
SRS	Presentation on Software Requirement Specification & analysis by staff	4
	Discussion on SRS with students	5
	Evaluation of students SRS	6
Design	Presentation on software Design by staff	7
	Discussions on Design	8
	Evaluation of students design	9
Module Implementation & testing	Presentation on implementation Details and coding standards	10
	Time allotted for students to implement	11
	Evaluation of midway implementation	12
Integration and System Testing	Integrate all modules and perform testing & test according to test plan	13

IV. Assessment

The main challenge for the faculty is monitoring and assessing large number of students' project. Assessment rubrics are written to guarantee proper understanding of the expectations among the stake holders resulting in fair assessment.

A. *Assessment rubrics and Assessment Matrix*

The assessment rubrics [13]-[14] help to differentiate between excellent, good and bad project. Assessment rubrics are written to map the objectives of the phase. Assessment rubrics for SRS are shown in Fig. 2. As rubrics are lengthy, reading and remembering these rubrics pose another challenge during evaluating the projects. So Assessment matrix is written for each

phase. It is "one page document to perform assessment of a phase in a project with quantifiable measures written according to rubrics". Assessment matrix for SRS phase is shown in Fig. 3. Assessment rubrics contain questions and expectation of marks range for each of rubrics written for the phase. These documents reduces burden of reading rubrics every time and evaluation is easy.

Functional Requirements with use case diagrams	
Excellent	Requirements should be clearly defined for the identified features and should satisfy quality characteristics such as complete, correct, unambiguous, verifiable, consistent, Ranked for importance and / or stability, modifiable, traceable
Good	Few Requirements does not satisfy quality characteristics of SRS
Moderate	Requirements are not identified for the features listed for the problem definition
Test plan for acceptance testing.	
Excellent	Write test cases for identified inputs, outputs and features
Good	Write few test cases for identified inputs, outputs and features.
Moderate	Test plan is not clearly defined and unable to write teat cases
Presentation and communication skills	
Excellent	The organization of contents is appropriate for its purpose. Uses rich and varied and appropriate vocabulary. Confident and relaxed in the whole presentation Engaging to audience Handle difficult questions with ease and confident Illustrative explanation
Good	The organization of contents is appropriate for its purpose. Selects words appropriate for an audience and uses correct grammar. Confident in most parts of the presentation Answer questions correctly and concisely
Moderate	Basic organization of contents and preparation. Use simple vocabulary or makes consistent errors in grammar. Confident in some parts of the presentation Unable to answer the questions with accurate supporting evidence.

Fig. 2 Rubrics for SRS

S. N	Parameter	Marks
Name of the student		
1	Is the team able to tell the process model used for project development and justify.(marks 01-02)	
2	How many requirements are clearly defined for the identified features(marks 01-05)	
3	How many test cases for acceptance testing are written as per the template?(marks 01-05)	
4	How many non functional requirements are written with measurable parameters? (marks 1-2)	
5	Is Effort estimation done ? (marks 0-1)	
6	Does the student have clarity about Software Requirement Specifications? (marks 01-05)	

Fig. 3 Assessment matrix for SRS

Assessment rubrics and matrix are written for all phases of mini project. These helped the faculty members to perform fair assessment.

V. RESULTS

Feedback is taken for 34 project batches each batch consists of 4 students at the end of the semester to know the impact of the approach followed. Feedback form contains questions relating to the learning objectives. Feedback form is shown in Fig. 4. Analysis of feedback is done and graph is shown in Fig. 5.

Sl No	Description
1	Objectives of the mini project were made known to us in the beginning of the semester
2	Phase wise presentations made by faculty members were useful to understand the life cycle activities of software development
3	Mini project helped me to improve problem solving skills
4	Mini project helped me understand the process of engineering a software
5	Mini project helped to improve presentation communication and documentation skills
6	Mini project helped me to develop the ability to work in a team
7	Assessment criteria was made known to us well in advance
8	Assessment of mini projects was transparent
9	Regular lab slot for mini project helped us to do the mini project in a timely manner following the life cycle activities

Fig. 4 Feed Back Form

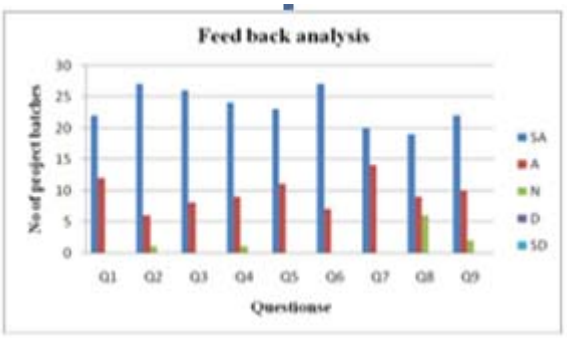

Fig. 5 Feedback Analysis

Feedback indicates that all the project batches either strongly agree or agree that mini project helped to meet the Course learning objectives. Performance in Continuous Internal Exams (CIE) and Semester End Exams (SEE) are also showed marked improvement as shown in Fig. 6. A CIE and SEE curve shows a Gaussian distribution. This process of implementation of mini

projects is conducted for two years 2008-2009 and 2009-2010 consecutively. This has provided a better understanding of carrying out projects among students and also helped faculty to evaluate projects correctly. Results of two years are shown in Fig. 7. Results of 2009-2010 are improved because of the approach followed.

Fig. 6 CIE and SEE performance

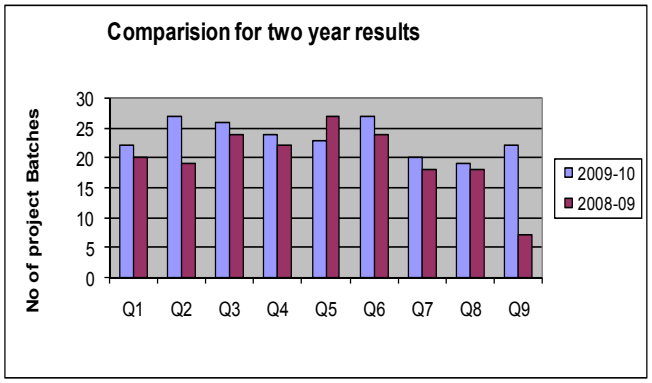

Fig. 7 Comparison of two years result

VI. CONCLUSION

The approach followed has made tremendous change in the students' understanding and implementation of projects. The quality of deliverables produced by students has improved. Students could complete their projects in the scheduled time. Students' problem solving skills, communication skills and ability to work in teams are improved. It provided strong foundation for capstone projects and also marked improvement in the quality of capstone projects.

REFERENCES

[1] Richard M. Felder Department of Chemical Engineering North Carolina State University "Designing and Teaching Courses to Satisfy the ABET Engineering Criteria", Journal of Engineering Education, 92 (1), 7-25 (2003).

[2] Dym, C. L., and Little, L., "Engineering Design: A Project-Based Introduction", 2nd ed. John Wiley, New York, N.Y., 2003

[3] Tockey, S., "A Missing Link in Software Engineering,", IEEE Software, vol. 14,no. 6, pp. 31-39, 1997.

[4] Lethbridge, T.C., "A survey of the Relevance of Computer Science and Software Engineering Education", Proc. 11th SEI Conference on Software Engineering Education, Atlanta, GA, Feb. 1998

[5] Jackson, U., Manaris, B., and McCauley, R., "Strategies for Effective Integration of Software Engineering Concepts and Techniques into the Undergraduate Computer Science Curriculum",: ACM SIGCSE Bulletin, vol. 29, no. 2, 1997.

[6] Chandra R. Sekhar, Omer Farook and Essaid Bouktache,"Continuous Improvement Process Based on Outcome Based Education",Purdue University Calumet.

[7] Johnson, H.A., "Integrating Software Engineering into the Traditional Computer Science Curriculum", ACM SIGCSE Bulletin, vol. 29, no. 2, 1997.

[8] Petkovic, D.; Todtenhoefer, P.; Thompson, G.: "Teaching Practical Software Engineering and Global Software Engineering: Case Study and Recommendations." ,Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference (October, 2006, San Diego, CA).

[9] Bloxham, S., and West, A., "Understanding the rules of the game: Marking peer assessment as a medium for developing students'conceptions of assessment", Assessment and evaluation in highereducation, 29, 6, 2004, 721-733..

[10] Graesser, A., K. Lang, Horgan, D., "A Taxonomy for Question Generation", Questioning Exchange, Vol.2, No.1, 1988, pp.3-15.

[11] Workshop on Best Practice in Software Engineering: "The Role of Industry in Software Engineering Education and Training". Held during the 19th Conference on Software Engineering Education & Training (CSEE&T2003), Turtle Bay, Hawaii, April 2006.

[12] International Workshop on Informatics Education: "Bridging the University/Industry Gap". A Workshop held within IFIP Conference on Education for the 21st Century - Impact of ICT and Digital Resources which was part of the 19th IFIP World Computer Congress,Santiago, Chile, August 2006

[13] Petkovic, D.; Thompson, G.; Todtenhoefer, R., "Assessment andComparison of Local and Global SW Engineering Practices in aClassroom Setting.", Proceedings of the Thirteenth Annual onferenceon Innovation and Technology in Computer Science Education,Madrid, Spain, June 2008.

[14] Magdeleine D. N., Alwis, W. A., Henk, G. S., " Peer assessment in problem-based learning: Students' view ", IAEA Annual Conference, Cambridge, UK, 2008, 1-9..