Integrated Project Based Learning in Software Engineering Education

Sohan Singh Yadav[1]
Software School, Xiamen University
Xiamen, China
rjxy1@xmu.edu.cn

Jianbing Xiahou[2]
Software School, Xiamen University
Xiamen, China
jbxiahou@xmu.edu.cn

Abstract- This paper presents an education framework to effectively develop crucial software engineering skills in students of software Engineering major at National Exemplary Software School (NESS), Xiamen University. The goal is to describe a systematic approach towards integrating project based learning in software engineering major, both inside and outside the classroom. An essential part of Software Engineering Education is practical training in principles, methods and procedures under conditions similar to developing real software products. This paper describes the different conventional and traditional approaches at length for Software Engineering Education and proposes integrated project based learning approach is more effective and interesting for teaching and learning SE as compared to the lecture based approach.

I. INTRODUCTION

Software Engineering Education is witnessing a positive change, where innovations and improvements in curriculum, instruction and assessment being directed towards bridging the academia-industry gap by projecting the true nature of software development and facilitate the student in nurturing essential knowledge, skills and attitude that are actually needed by the industry[2, 3, 4, 5]. Software Engineering as a discipline is deeply linked to practical aspects of developing software within cost, schedule and quality requirements. For this the Software Engineer should have technical as well as managerial expertise, while being aware of economic issues. Many learning models and pedagogies are in use and many new ones are being developed for teaching Software Engineering in academic and professional environment. But the current Software Engineering Education is mostly based on the class room learning model that is found rather unfulfilling the requirements posed by the practical and interdisciplinary nature of the subject. The focus is on using integrated project based learning as the teaching instruments to effectively address the learning disabilities [6].

Focusing on real-world, open-ended problems, problem based learning is a pedagogy that has been shown to increase student motivation and interaction [7]. This method requires cooperation among students and promotes higher level thinking that is not always necessary when learning by more traditional, independent methods [8, 10]. Furthermore, the collaborative component provides the advantages of supporting the development of social and team-working skills [9].Education in Software Engineering for technical applications requires the knowledge of the principles, methods and procedures as well as their application in projects of modest size. Software projects should be done in teams and should include requirements specification, analysis, design, construction, quality assurance and project management. Project requirements have to be stated in such a way that the participants are confronted with the true problems of the specific applications.

II. OBJECTIVES OF INTEGRATED PROJECT BASED LEARNING

The main purpose of the integrated project based learning in the software engineering course is to gain some software engineering experience which cannot be obtained with small, method-driven lab exercises with problem statements that are usually tailored to already prefabricated solutions. Moreover, some lab examples are often oversimplified such that they do not seem to be relevant to industrial practice at all. Some topics (e.g. process models) are even covered by lectures only and at a first glance they seem to be rather theoretical and of no immediately visible practical use. In order to reply to these irritations, it proved to be an essential experience to students to take part in a software development team. However, education by integrated projects is not only delimited by the possible duration of a project but also by the kind of particular skills which should be trained.

III. CURRENT APPROACHES TOWARD SOFTWARE ENGINEERING EDUCATION

The traditional and the most prevalent approach to Software Engineering Education is a lecture based paradigm, sometimes accompanied by lab exercises. A major problem with the lecture based paradigm is that the student is usually a passive audience during lectures and doesn't get involved in the learning process. This passive mode of learning is great worry of academia as well as industry.

The traditional and the most prevalent approach to Software Engineering Education are to impart theories and concepts in a class room environment. Research done by learning scientists has proved over and over that this kind of learning is no learning at all. [1, 13]. The traditional model of class room learning enforces a reactive rather than proactive approach. i.e., the participants are expected to react to a solution presented instead of proactively thinking about the

978-1-4244-7662-6/$26.00 © 2010 IEEE

problem on hand. The participants settle for short term objectives like making it through the course instead of leveraging on long term goals of learning. Owing to this mismatch between the education and its application; the present teaching methods prove to be inadequate for the purpose.

Many new models, often criticizing the conventional methods have emerged. Some teaching methods are based on principles of goal driven learning, learning by doing, story centric learning and few others recommend self-direction, cognitive learning, team based or collaborative learning. Each technique has its own pros and cons. But none of these techniques has been found a best fit for Software Engineering Education in both the academic and the industry.

IV. INTEGRATED PROJECT SETTINGS

An integrated project based learning approach model can be used for all the projects carried out in the software engineering course. The instructor sets project carefully and make sure that knowledge learnt in classroom during lecture can be applied to the integrated project. Student works in a team and each project team consists of 4 to 6 students, depending on the prospective workload. The size of the project team is set in advance by the instructor. The actual team membership can be partly decided on by the students themselves. The team members elect one person as their team leader who is responsible for project-coordination in an organizational sense. Further resource allocation lies in the responsibility of the team as a whole. The instructor acts as a mentor and can be contacted in case of any problems during the development process. It must be ensured that projects possess some general properties which are subsequently discussed. Main focus of the integrated project is as given below

Requirements engineering: Requirements engineering proved to be one of the most crucial phases in a software development process [11]. Each project should challenge the students in the elicitation, modeling and validation of requirements. Students have to select and apply correctly the appropriate elicitation, specification and validation methods and techniques.

Software design: The students should be able to cross the bridge from the analyzed problem domain to the architecture and design of the solution domain. Again, the choice and correct application of proper design methods and techniques are of essential training value.

Implementation: Students attending a software engineering course usually already proved to be able to formulate algorithms during a prior course in programming techniques. For the purpose of the software engineering course it remains that students have to show that they are able to transform given design units into code.

Teamwork: Software engineering is teamwork. Students should gain experience in cooperating with others in a larger context. They have to fulfill special roles in the development process.

Project management: The students should be able to establish a realistic project plan. The group leader has to act as a responsible project manager who is able to coordinate the different tasks of the team members.

Project documentation: Documents emerging from the software development phases have to conform to given templates and standards (e.g. requirements specification [12]).

V. INTEGRATED PROJECT BASED LEARNING AT NESS, XIAMEN UNIVERSITY

Convinced by the benefits of integrated project based learning approach over the traditional class room teaching, the authors experimented with the integrated project based approach towards Software Engineering Education. This method intends to create an effective learning environment, while being suitable to all types of learning needs and overcoming impediments in other methods. In order to establish such abilities we first lay a rather traditional lecture/lab-based foundation. On these grounds, a carefully designed integrated project based learning approach bridges the gap between dry knowledge and lively experience. This scheme can be very efficient and effective but also quite rewarding for both students and instructors. Proposed approach discusses important characteristics of projects as well as processes, central issues and future prospects.

1) Philosophy

Integrated project based learning approach emphasize on the application aspects of the tools, techniques and most necessarily, the concepts learnt so far. This approach is suitable for software engineering education because the subject itself is not just a bundle of concepts, techniques or models; instead the subject has a broader perspective and uses concepts from many domains to solve a problem. Projects are more relevant to the kind of challenges professionals, both young and experienced, face and bring in the elements of practical way of solving problems and resolving complex situations. Software Engineering is just this kind of subject, and software engineering education needs integrated project based learning model that help the participants to understand and manage the true nature of the present software industry, complex, evolving and challenging.

2) The project, Faculty and the Participants

Projects are the main teaching instruments in this approach. A rightly designed project can increase the effectiveness of the learning environment. The types of Software Engineering project that can be used include Requirements engineering, software design, Implementation, team work, project management and software documentation, which promote a product or a firm and can exhibit industry and technological trends.

Project has one major goal and many minor teaching goals. They can't focus on just one issue/topic, as that would not reflect the real world situation where things are never isolated. The integrated project based approach can be a hypothetical one, a real life project experience, or an adapted version of the real project. It is important to have a good understanding of the approach by the facilitator and the participants. All participants should be willing to put sincere efforts for analyzing and presenting the analysis. Technically, the participants must have taken a first course in programming and should have some experience of software projects, though academic.

3) Teaching Strategy

Project always bring a challenge and the participants apply their conceptual learning to solve the challenge. The methodology for using integrated project for software engineering education is to present the project to the course participants and then hold discussions, where the participants may discuss their solutions/reactions to the challenge, with other participants. The project can be assigned by the faculty or can be chosen by student from the given list. The most effective learning in an integrated project based approach comes through the discussions, team work and proper focus on every stage of the project development. Project should be guided by the faculty, while not biasing the participants and encouraging creative, justified and proactive solutions. The project should aim to bring out the teaching goals while enhancing the analytical and communication skills of the participants.

4) Observations

Most important observation being the positive change from earlier perception of participants, that Software Engineering is a theoretical and hence unimportant subject. Many participants have now started applying the Software Engineering concepts (especially design and coding standards) in their ongoing projects. The quality of class participation increased gradually. Also, the industry has reciprocated to the approach by showing their interest in contributing to the project pool.

VI. CONCLUSION AND FUTURE WORK

The integrated project based leaning approach provides a common framework for learning the concepts, skills, tools and techniques in presence of a context such that the instructor and student are engaged in a meaningful discussion and the learning becomes student centric and active. We have extended the concept of integrated project based learning by focusing on the design and development of a real word project that can be utilized as a real solution. Course instructor in a role as an external guide or overseer, our approach is unique in that incorporates the instructor into the development team as an experienced but mutual partner. By giving the team control of the project from the beginning, students take personal ownership of the project, becoming responsible for their own learning. Thus, this approach motivates intentional learning and improves learning the course material, teamwork, communication skills and supplementary knowledge of topics that help the project achieve success.

REFERENCES

[1] Kemi Jona, "Rethinking the Design of Online Courses", Keynote paper, ASCILITE 2000

[2] "Guide to the Software Engineering Body of Knowledge-SWEBOK", 2004. A project of the IEEE Computer Society, Professional Practices Committee

[3] Shaw M.(editor). Software Engineering for the 21st Century: A basis for rethinking the curriculum, Technical Report CMU-ISRI-05-108, Carnegie Mellon University, Institute for Software Research International, Pittsburgh, March 2005.

[4] [Sivan A. et al "An Implementation of Active Learning & its Effect on the Quality of Student Learning", Innovations in Education and Training International, Vol. 3

[5] Varma V., Garg,. K., Case Studies: The Potential Teaching Instruments for Software Engineering Education", Proceedings of the 5th International Conference on Software Quality, Australia, 19-20 September, 2005

[6] Senge PM, "The fifth discipline: the art and practice of the earning organization" NewYork: Currency Doubleday, 1990.

[7] Barg, M. et al. "Problem-Based Learning for Foundation Computer Science Courses". Computer Science Education, 10, 2000, pp. 1-20.

[8] Gokhale, A. A. "Collaborative learning enhances critical thinking." Journal of Education 7, no. 1, 1995.

[9] Johnson, D. W., Johnson, R. T., and Smith, K. A. "CooperativeLearning Returns to College: What Evidence Is There That It Works?" Change, July/August 1998, pp. 27-35.

[10] Smith, K. A., Sheppard, S. D., Johnson, D. W., and Johnson, R. T. 2005. "Pedagogies of Engagement: Classroom-Based Practices". Journal of Engineering Education, Vol. 94, No.1, 2005.

[11] H.F. Hofmann and F. Lehner, "Requirements Engineering as a Success Factor in Software Projects", IEEE Software, Vol. 18, No. 4, July/August 2001, pp. 58-66

[12] IEEE Recommended Practice for Software Requirements Specifications, ANSI/IEEE Std. 830-1993, IEEE, New York, 1993

[13] Whitehead, A. N. (1929), The aims of education. New York: Macmillan.

[14] A Case Study Initiative for Software Engineering Education Vasudeva Varma, Kirti Garg, Vamsi International Institute of Information Technology, Gachibowli, Hyderabad, India