Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (2) 167-170 (2012)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Effective Use of Student Projects in Teaching Software Engineering

S. SHAHZAD, A. KEERIO*, J. A. MAHAR**, Z. HUSSAIN***

Department of Computer Science, University of Peshawar, Pakistan

Corresponding author, Email: ayazkeerio@hotmail.com Cell No. +92 3412802830

Received 13th December 2011and Revised 27th April 2012

Abstract: Software Engineering education aims at providing the students with all the required technical, organizational and social skills to deal with the cumbersome task of development as well as to survive the complicated situations in software development industry. The perspective software engineers need to be prepared for the volatile environment of the industry, where decisions are made in real time and things do not go as smooth and pre-planned as the students experience them in the controlled environment of academic institutions. Being researchers and instructors of software engineering at public sector universities, the authors are able to see the problems due to which the perspective software engineers are unable to thrive in the software development industry. This research has been conducted as a pilot study for a project aiming at reforming software engineering teaching and education in Pakistan. The study highlights the problems in software engineering education in general by providing an analytical review of different models and frameworks for teaching project-based software engineering capstone courses. All of these models and frameworks are designed and tested in order to have improved learning outcome of software engineering courses. The aim of presenting this literature review is to identify a proper teaching model best suited to teach project-based software engineering capstone courses in Pakistan.

Keywords: Software Engineering Education; Software Development Industry; Project Based Teaching.

1. INTRODUCTION

Generally, within universities running graduate and undergraduate Computer Science (CS) and Software Engineering (SE) degree programs, a lot of software projects are being developed each year as a requirement for the degree. Most of these projects are not graded as successful. The reasons vary from lack of technical expertise of student developers to the misunderstanding and misinterpretation of the importance of software development process. Due to this the projects are either cut in scope, left incomplete, or even if completed most of the projects are never submitted to the original customers. The authors, being instructors of SE are working towards reforming SE education. There is a need to investigate into the reasons of this wastage of time and efforts of all the parties involved into software development at the academic level in order to make this effort fruitful and beneficial both for the students and for the software development industry. An exhaustive literature review is being conducted to find out best practiced teaching models and solutions adopted by renowned researchers and members of academia to tackle with this problem in SE education.

The rest of the paper is organized as follows: Section 2 explains research design; section 3 gives a brief background of SE education and explains the importance of software development project in SE education; section 4 provides a summarized view of design, implementation, benefits, and drawbacks of some selected and important SE teaching frameworks and models; section 5 gives some important and useful ideas to improve the learning outcome; section 6 gives a conclusion.

2. MATERIAL AND METHODS

2.1 Research Design

This research has been conducted as a pilot study for a project aiming at reforming SE teaching and education at public sector universities. The aim is to improve the learning outcome and to bridge the gap between industry and academia. This gap exists due to inadequate practical and industry oriented training of the students and the dynamic situations in real life software development.

The study conducts an analysis of the current trends in SE education, especially in training the students for software development projects. As a

*University of Sindh, Jamshoro, Pakistan
**Department of Computer Science, Shah Abdul Latif University, Khairpur, Pakistan
***Quaid-e-Awam University, Nawabshah, Pakistan

general trend, all CS and SE degree programs conclude with a final software development project. This project is supposed to exploit all the theoretical foundations of CS as well as the practical aspects and concepts of SE with which the students have been trained during their studies.

The following sections take a look at what different trends exist in teaching and training software engineers. The case studies presented below highlight the importance of SE education and place of development projects in teaching SE foundation processes.

2.2 Importance of Analyzing SE Education

SE education system is still not producing the quality results. The fresh software engineers face a lot of problems when they enter the industry with knowledge and training they receive in CS ad SE degree programs. There is an ongoing process which has been started to improve the SE education and teaching and to reform whatever is needed according to the requirements of the industry. A lot of research has been conducted in this regard. Professional software engineers, skill development experts and members of academia have given the output of the research in the form of guidelines, frameworks, and information to tackle with different problems occurring in SE education as well as in software development industry.

2.3 Importance of Software Development Projects in Teaching SE

Many of the researchers (Reichlmayr, 2006; Todd et al, 2005; Oakley et al, 2004; Beasley, 2003) agree that using development projects to teach technical and engineering related subjects (including SE) improve the learning outcome expected from the courses, and it also makes the students aware of the wholeness of the overall engineering process, which is otherwise taught to them in parts. Another main objective is to prepare the perspective software engineers for real world software engineering projects. It is also noted that formalities and complexities of small scale software development ventures at university make the students realize the size of the problems in real software development (Johns-Boast, 2009).

3. Frameworks and Models for Project-Based SE courses

3.1. Small teams, Role-Based, Real World Project

Johns-Boast and Flint (Johns-Boast, 2009) have presented a teaching framework using real-world projects. They propose to assign different roles to team members that exist in any software development company. Teams are composed of a mix of senior and junior students, with senior students acting as team leaders (mentors, coaches, managers) and juniors be doing all the technical tasks needed in the development. During the course of the project different tutorials are arranged for the students to guide them in team building, software development process and other managerial tasks which are the routine in the industry.

The assessment and grading of students towards the course is done at different stages during the project. The students are asked to submit written reports and demonstrate the process that they are performing.

Johns-Boast and Flint's (Johns-Boast, 2009) experience highlights some important facts to be considered when designing and conducting project based software engineering courses:

- In role-based teams, if grading is done on individual basis then the students lost the idea of team work and concentrate only on their own role, whereas when they know that their work will be graded as the overall team performance then they try to do their best, not only in playing their own assigned role but also help others to perform best. In this way they learn not only the process of their role but they are able to learn the overall software development process.

- Grading effort of individual students towards the project is very difficult.

- Students concentrating only on the tasks of their role also fail to understand the dependencies between the different phases of software development process.

- Some roles (team members) have too much to do (for example, programmers) and some have to work only sometimes during the project (for example, the client or customer), thus there is an uneven distribution of work load among the team members.
- If the project clients are not for real then the students lost interest in the development as they know that the software is not going to be used ever.

- Making teams of senior and junior students is also beneficial for knowledge management and knowledge distribution among the team.

- The project should be selected very carefully so that all the important aspects of software engineering can be demonstrated clearly. The project should also let the students make use of their

technical skills but also should provide them with an opportunity to see the team organization and communication which is a very important aspect of real life software development industry.

3.2. Large teams: company based

David *et al.,* and Mohamed (Borman *et al.,* 2011) have tested their framework on large teams of up to 30 students, whom they call the employees hence giving the name "Company-based" to their framework. They argue that the problems of non- effective teaching in project based capstone SE courses are that the projects are maintained in small teams which hinder in presenting the actual industry situation to the students. For this reason they propose to organize the students in large teams to demonstrate the real world situation in the software development industry from "organizational, process, and communication perspective" (Borman *et al.,* 2011). A similar study has been conducted by one of the authors (Shahzad *et al.,* 2009), where Extreme Programming (XP) practices have been used to teach XP software development process to a combined class of graduate and undergraduate students. That project based course emphasized mainly the knowledge management factor and the learning perspective of the XP practices that were implemented.

Zaigham 2007 has proposed a teaching framework in which the team size is gradually increased from individual to pair in the first year of studies, and to a team of ten members in the final year of SE/CS degree program. (Mahmood, 2007) has also discussed a complete course structure, including the process of selecting an appropriate project, forming student teams, software development phases, and also the grading system for the SE training course(s).

3.3. Phase-Based Training

Researchers and members of academia have also attempted to provide the students with expertise in the individual phases of software development by concentrating on a single phase at a time instead of training the whole process at once. It is argued that software development is not about only coding but also requires a number of other social and technical skills. As, according to Hoare (Hoare, 2006), the purpose of software testing is not only to find out the errors in the code but also to analyze the technical, process related and organizational skills of the development team. Jamaludin *et al.,* Jamaludin *et a.l,* 2008) promote the criticality of the skill development in SE and designates requirements engineering (RE) process as the first step towards skill development of the students. They argue that if the students fail to develop their required technical and organizational skills in the initial stages of the process, they will not be able to catch up with the difficulties and required skill levels in the upcoming phases of the development.

3.4. Project Management Teaching Framework

Managing a software project is as critical as the actual development. According to Boehm (Boehm, 1991) problems in management is one of the greatest factors contributing to project failure. Many studies have been presented highlighting the importance of project management and its role in SE education. (Keenan, 2006), (Huang, 2008), (Jones, 2008) and (Javed, 2007) are some to mention who have worked towards developing teaching and training models in the context of software project management.

4. RESULTS AND DISCUSSION

4.1 Evaluating Teaching by Evaluating Learning

The following section describes the concept of retrospective learning and process reviews.

4.2 Final-Year Projects: Personal Review

The authors conducted an informal ethnographic survey, aimed at understanding the development and management related problems faced by final year students. The students were asked about their skill levels, their choice of development tools, their personal view about the process that they followed in each individual phase of the development, and about the problems and difficulties that the students faced in managing the project.

4.3 Learning Lessons from a Project Failure: Project Retrospectives

Knowledge management and learning from experience play a vital role in today's SE practices (Dingsøyr, 2004). Taking a retrospective review of projects after completion enables software engineers to learn from their own past experiences. In this way they are able to find out their own mistakes and can prevent themselves from repeating the same mistakes again in any other software development venture. (Dingsøyr, 2004) promotes a project as a common place or stadium for software engineers to learn by practice and learn from peers. Retrospectives not only promote learning but also provide an opportunity to evaluate a project as a success or failure (Nelson, 2005). (Razmov, 2007) suggests that software engineering projects are best place for teaching software engineering to prospective software engineers.

5. CONCLUSION

Software development requires expertise in a number of dimensions. Teaching this multi-faceted area of knowledge also requires lots of effort on behalf of the instructor. Software development process can only

be demonstrated by actually doing it. Teaching software development using software development projects is hence proven to be very beneficial. Different frameworks and models have been defined after research and experience of members of academia and practitioners from the industry. The choice of a particular model depends upon factors like the required skill level of students, strength of students to be trained, and availability of resources. The analysis of all the frameworks and models presented in this paper also facilitates to define a framework according to the SE education needs in Pakistan.

REFERENCES:

Bealsey, R.E. (2003) Conducting a Successful Senior Capstone Course in Computing. Journal of Computing Sciences in Colleges, October 2003, Vol. **(19):** Issue 1, 122-131

Boehm, B. W., (1991) Risk Management: Principles and Practices, IEEE SW, 32-41.

Broman D., S. Kristian and A.B. Mohamed (2011) The Company Approach to Software Engineering Project Courses. IEEE Transactions on Education (ISSN 0018-9359)

Dingsøyr T.(2004) Postmortem Reviews: Purpose and Approaches in Software Engineering. SINTEF Information and Communication Technology, SP Andersens vei 15b, 7465 Trondheim, Norway

Huang L., and L.Y. Gping, (2008) Interactive learning environment for designing for SPM teaching Wicom. 1-4.

Jamaludin N., A. Akmal, S. Sahibuddin, and N. H. Hidayat. (2011) Challenges of a project-based learning approach towards requirement engineering. In Proceedings of the 10th WSEAS international conference on Software engineering, parallel and distributed systems (SEPADS'11) World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 66-71.

Javed T., and Manzil-e-Maqsood, (2007) Practicum in SPM- An endeavor to effective and pragmatic SPM education, ACM. UK.

Johns-Boast L., and S. Flint (2009) Providing Students With 'Real-World' Experience Through University Group Projects. 20th Annual Conference for the Australasian Association for Engineering Education. AAEE. Australasian.

Jones J. and T. Mark (2008) A Case Study of Classroom Experience With Client-Based Team Projects, journal of Computing Science in Colleges, Vol. **(23):** No. 5. 45-48.

Mahmood, Z. (2005) A Framework for Software Engineering Education: A Group Projects Approach. International Journal of Computing and Information Technology, Issue 3 Vol. **(1):** 34-56.

Nelson R. (2005), Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between. MIS Quarterly Executive journal Vol. **(4):** No.3.79-84.

Oakley, B., R.M. Felder, R. Brent, and I. Elhajj, (2004). Turning Student Groups into Effective Teams. Journal of Student Centered Learning, Vol**. (2):** No.1, 9-34.

Razmov V. (2007) Effective Pedagogical Principles and Practices in. Teaching Software Engineering through Projects.37th ASEE/IEEE Frontiers in Education Conference.

Reichlmayr, T.J. (2006) Collaborating With Industry – Strategies for an Undergraduate Software Engineering Program. In Proceedings of the 2006 international workshop on Summit on software engineering education, International Conference on Software Engineering Shanghai, China. SSEE'06. 13-16.

Shahzad S. and W. Slany (2009) Knowledge Management Issues in Teaching Extreme Programming in Proceedings of I-KNOW '09 (2009) 278-288.

Thomas P. W. (2005) A Company-Based Framework For A Software Engineering Course. In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '05). ACM, New York, NY, USA, 132-136. DOI=10.1145/1047344.1047399 Http://Doi.Acm.Org/10.1145/1047344.1047399. 2005.

Todd, R.H. and S.P. Magleby, (2005). Elements of A Successful Capstone Course Considering The Needs Of Stakeholders. European Journal of Engineering Education, May 2005, Vol. **(30):** No 2, 203-214.