A METHODOLOGY TO ENHANCE PROGRAMMING SKILLS OF E-LEARNING STUDENTS

M. T. Pervez, N. Aslam[*], S. Muhammad, N. Naveed, S. Qadri[**]and S. Ahmad

Department of Computer Science, Virtual University of Pakistan, Lahore
[*]Department of Computer Science, NFC Institute of Engineering & Technological Training, Multan,
[**]Department of Computer Science, Islamia University Bahawalpur, Bahawalpur
Corresponding author e-mail: tariq_cp@hotmail.com

ABSTRACT: In this study, we proposed an approach, which helped students to improve their software engineering, designing, developing and coding skills. As a part of this methodology, we developed a comprehensive infrastructure including client-server, sub-versioning control setup and a three tier teaching pedagogy entitled as student-mentor-teacher layer. To test the proposed methodology two experiments were conducted; two types of feedbacks were collected. Adempiere ERP- an open source project was selected as test case. We also highlighted challenges of working on open source projects. Results showed that students who worked on open source projects were more skillful in designing and developing software projects than those who worked on standalone projects. Feedback of student's mentors as well as of teachers recommended using the proposed methodology that helped in improving technical skills by working with highly experienced professionals of open source software community.

Key words: Open source, e-learning, distance learning, methodology,

(Received 9-06-14 Accepted 6-3-15)

INTRODUCTION

Students of conventional universities select and work on the final year projects under the direct supervision and guidance of their teachers. Students of e-learning and distance learning institutions face problems of distance, direct coordination and guidance to work on the projects. Due to these problems of the domain of distance learning, the students remain unable to learn software designing and developing skills. Secondly, the projects offered to the students of conventional and distance learning universities are small, non-professional and do not have industrial implications. They do not have real clients and hence do not have opportunities to learn the techniques and approaches used for professional documentation. Just like medical students who do home job and work with their seniors to improve their knowledge and skills, the IT graduates should also have access to an environment where they can work with the highly experienced IT professionals. Open Source Software (OSS) paradigm can provide an environment where students of distance learning and conventional universities can improve their IT skills by working with competent IT professionals. Software that is available free of cost along with their source code and which may be modified by other developers or community as per their own requirements is called an open source software (Schryen, 2009 and Hissam *et al.*, 2002; Ossher, *et al.*, 2010). Research, development and use of open source software are rapidly being popular for the last few years (Tamura and Yamada, 2008; Polancic *et. al.*, 2004 and Andrew, 2006). Some examples of high quality and most famous OSS systems are Linux, Apache, My SQL, and FreeBSD (Xiong *et al.*, 2009).

Due to lack of required skills, fresh graduates are not able to knob large, medium and even small-scale problems. Students have a lot of spirit, potential, and energy to learn, however, they generally lack technical expertise, experience of teamwork, and project management skills (Ellis *et. al.*, 2008). These deficiencies compel them to behave like trainees and are offered low pay. Consequently, fresh graduates do not have an opportunity to work with senior software developers and analysts.

OSS software applications and tools play a vital role in learning software design and development skills. OSS development thrives with usage of internet that enables software developers to communicate and share their ideas with other developers while being at different geographical locations (Xiong *et al.*, 2009 and Ahmed *et al.*, 2009). Unlike closed source software applications (Ford, 2007 and Bachmann and Bernstein, 2009), OSS applications are developed in loosely controlled environment by volunteers and developers (Xiong *et. al.*, 2009 and Ferenc *et. al.*, 2005). The developers of OSS projects are scattered around the world and contribute through internet and their contributions are freely available to use and edit according to one's own needs (Conley and Sproull, 2009). To get equip students with software designing and development skills; it is demand of current era to have an infrastructure where students can gain and improve the analytical and programming skills. The main goal of our work is to present pedagogy for e-learning students to train and prepare them for joining software industry without being exploited. This setup should mimic mechanism of medical students who gain expertise, experience and skill by working with their senior doctors. To achieve these objectives a project entitled "Integration of Open Source Software Projects in IT Education" was initiated by three Pakistani universities: National University of Computer and Emerging Science, Punjab University College of Information Technology, University of the Punjab, and Virtual University of Pakistan.

As a team of VUP, an infrastructure was developed which enabled students to work and participate in an OSS project in an e-learning environment. We worked on various open source software applications like AdempiereERP, OrangeHRM, SugarCRM and PostBooks. However, due to constructive and timely feedback from the international community our major contribution is in AdempiereERP. This study, based on more than two years' experience of working on various OSS projects, presents an approach to enhance software designing and development skills of e-learning students by getting then indulged in the OSS projects. Our methodology may work as a guideline for other educational institutes as well.

MATERIALS AND METHODS

In this section, the proposed methodology to develop infrastructure and engage students in the OSS project is described as under:

Choosing OSS Project: OSS project was selected keeping in view its ranking, activity and the number of full time developers. Ranking ranging from 1 to 500, activity ranging from 99% to 100% and 5 full time developers engaged in the project ware considered while finalizing the OSS project.

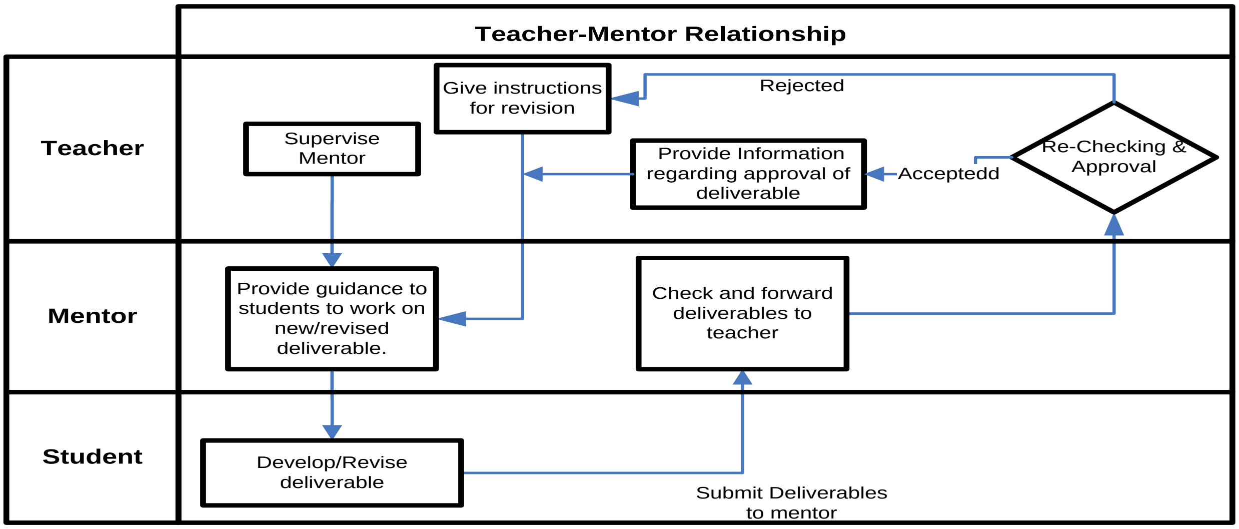
Infrastructure Development: Infrastructure including installation and configuration of a web server to host all software applications (such as Mantis, Visual SVN) was developed. This was required by students to perform various types of tasks and submit them conveniently. Several guides and tutorials such as Visual SVN installation and user guide, Tortoise SVN installation and user guide, Mantis installation guide, Adempiere ERP installation Guide (Prepared by VOSS Com), Postgre SQL (Database), ADempiere project overview tutorial, a guide to install and configure JDK 1.6, Eclipse debugging tutorial, XAMPP/WAMP installation guides, Video training on UML and object model were developed.

Students selection: The participants were selected keeping in view their ability and skills such as cumulative grade point average (CGPA), grades of major subjects such as data structure, algorithms, software engineering, DBMS and C++ language. Students' own choice was also confirmed whether they were willing to work on an OSS project or not.

Mentorship: In the eighth semester, the students, who completed their project in first two semesters, i.e. sixth and seventh, were selected as mentors. They acted as a middle layer between students and the teacher (Fig-1). A mentor checked documents of a student and sent it to the teacher. Teacher then updated mentors about their performance. Mentors remained in contact with teacher while providing any type of guidance to the students. In this way, teacher watched all types of activities of mentors and the students. Fig-1 shows relationship among teacher, mentor and student.

Student Ranking System: One of the core elements to work on an open source project, especially in an e-learning environment, was to encourage and motivate students and monitor their performance. For this purpose, we adopted an approach called as Student Ranking System (SRS) to create competition among them. The idea was to assign some points to various activities performed by students. The SRS was calculated based on points performance of every student and was published at Learning Management System of Virtual University (VULMS) each week. SRS played a vital role in improving performance of students. SRS was used to divide and assign points to various activities as shown table 1. A student was awarded 1 point if he sent activity report daily. If a student submitted his deliverable on or before a due date, 3 points were awarded to him and so on.

Viva voce Examination: At the end of final semester, viva voce examination of students was conducted who successfully completed their work.


Teacher-Mentor Relationship
Teacher
Mentor
Student
Supervise Mentor
Give instructions for revision
Rejected
Provide Information regarding approval of deliverable
Acceptedd
Re-Checking & Approval
Provide guidance to students to work on new/revised deliverable.
Check and forward deliverables to teacher
Develop/Revise deliverable
Submit Deliverables to mentor

Showing relationship among teachers/research officers, mentors and students

RESULTS

OSS projects enabled students to work in team, learn coding style, software engineering and development practices and approaches. We performed various experiments and concluded that students improved their IT, software engineering and coding skills by working on the open source projects.

Performance evaluation of students: First experiment was conducted to evaluate the skills of same 10 students in the domains of Software Engineering (SE), Object Oriented Programming (OOP), Implementation Concepts and coding style. Firstly they worked on independent projects and then worked on OSS projects (Fig-2). The duration of each session was 2 months. Percentage points obtained after working on independent projects and OSS projects showed that the students' skills were much better after having worked on OSS projects.

Table 1: Showing various parameters with their points used in student ranking system

Measures	Points
Daily Activity Report (DAR)	1 point per day
In-time Deliverable Submission (IDS)	3 point per submission
Asking Intelligent Question (IQ)	1 point per Question
E-mail Response (within 24 hrs) (ER)	1 point per Response
Submission Correctness (SC)	Percentage from 5 points
Participation in Chat/Tutorial Session (TS)	5 points per session

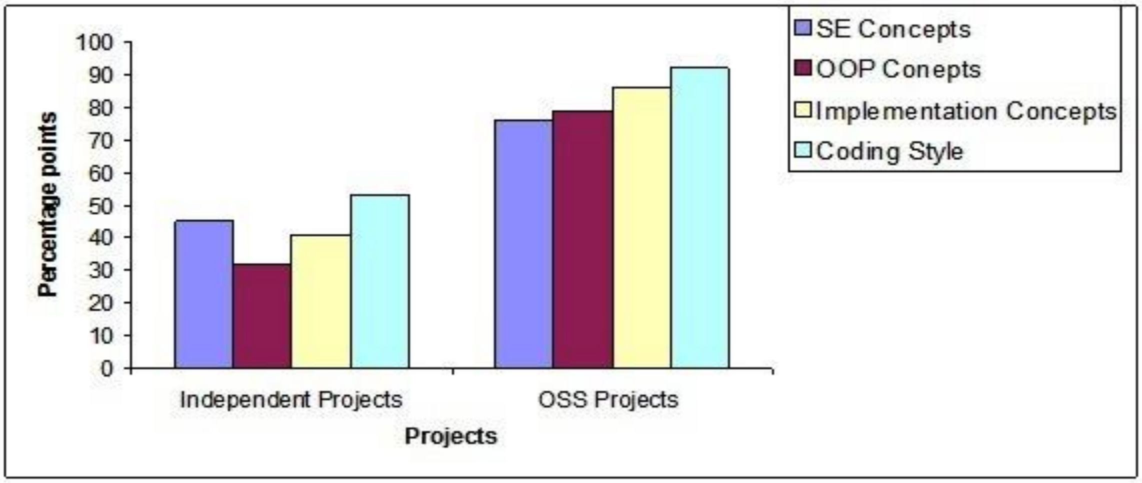


Fig. 2 Experiment to show performance of same students with OSS and independent projects.

OSS students vs. non-OSS students: In this experiment capability of two groups of students, each comprising 10 students was tested. All of them learnt and improved software design skills. They also leant how to implement objected oriented techniques such as aggregation and composition. Results showed that they had better skills than those who worked alone (Fig-3). Students, who worked on OSS projects showed very good performance in software engineering, object oriented programming, implementation and coding style competitions. They had much better concepts in all these areas of software engineering.

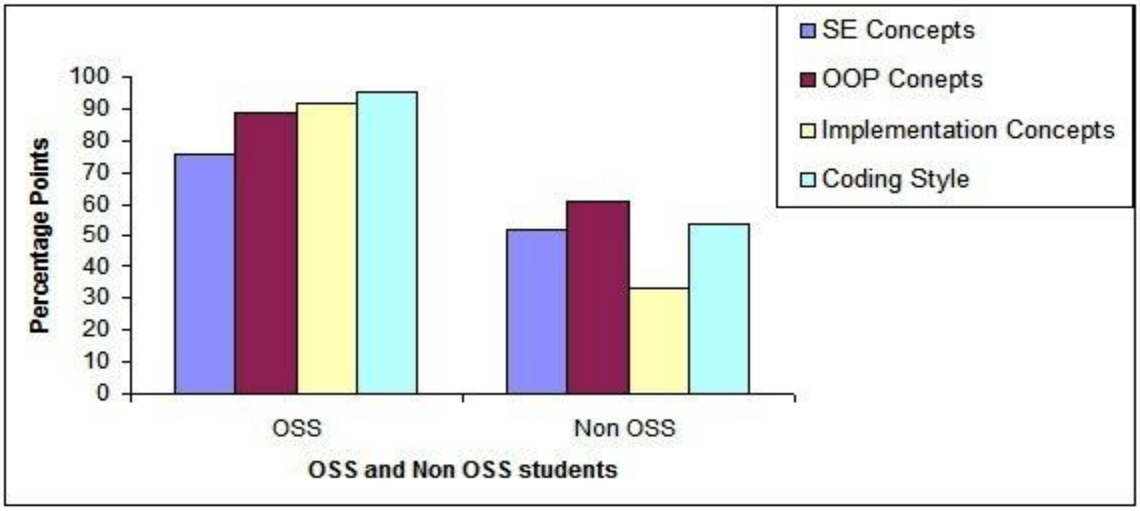


Fig. 3 Performance of OSS and non OSS students in software engineering, implementation and coding style competitions.

Success Rate of Students: A total of forty-three students belonging to various sessions worked on the OSS project (Fig-4). In first session, eight out of twelve students successfully completed their work and passed viva voce examination. In the second session, seven out of eight students successfully completed their work. In the third and fourth session, eleven and eight students successfully completed their work. This experiment showed that students who worked on OSS project also passed their examinations in good grades.

Student's Feedback: The purpose of collecting students' feedback was to know the reasons why some students fail/left work on the OSS project, to improve the performance of teachers by collecting flaws observed in students during working on the OSS project and provided a guideline to other educational institutes interested or planning to involve their students in OSS projects.

Below are given some important questions from the feedback form sent to students and their replies.

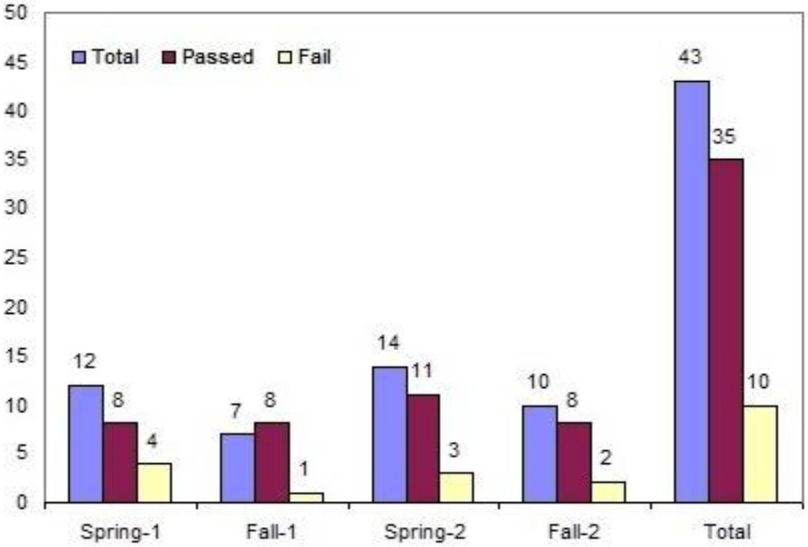


Fig. 4 Enrolment of students in OSS project and their rate of success.

Question 1: Do you think, working on an Open Source Project, is beneficiary for you and other students, how?
Reply 1: "Yes, because OSS project one has to play with professionally written code, which helps students to be familiar of professional coding techniques".
Reply 2: "If a student who is already working in some software development company, then it will be more beneficent for him in improving his skills."
Reply 3: "Yes, it is beneficial. For me, it was helpful in clarifying various OOP and programming concepts and techniques of a high level software".
Question 2: Do you think that there are any disadvantages of working on Open Source Project?
Reply 1: "Yes, I feel some disadvantages of working on OSS project. Student because of working on existing software can not show their innovative capability".
Reply 2: "OSS project advantages and disadvantages depend on the nature of student environment. If he is a regular student then this project is good for him and if he is working in a company other than software Development Company then it.
Reply 3: "No! I don't think that there is any major disadvantage of working on OS project".

Mentor's Feedback: In the final semester, a student selected as mentor acted as an intermediate layer between the student and teacher. Students used to send their deliverables to the mentor; mentor then after putting feedback on the students' work sent the document to the concerned teacher. The teacher evaluated the work/performance of both the mentor and the student and putts his own comments on the document, which was returned back to the mentor. The mentor returned the document back to student. The purpose of involving students as mentors was to equip them with some managerial as well as teaching skills and experience. Overall the mentors gave good results.

The purpose of the feed back of mentors was to pinpoint their difficulties especially in an e-learning environment and skills they got during mentorship.
Below are given some important questions from the feedback form sent to mentors and their replies.
Question 1: Write down few pros and cons of Mentorship?
Reply1: "Mentorship is very useful by various perspectives like it gives confidence, helps to clear concepts of various subjects such as DBMS, programming etc., also gives hands on experience of leading the people. The only problem I faced was the time management especially by the job holder students because they don't have enough time to participate in an OSS project at regular basis".
Reply 2: "Mentorship is a very good approach from the perspective of new students. It provides them an opportunity to act like a teacher".
Question 2: What knowledge and skills you have learnt by working as a mentor in the Open Source Project?
Reply 1: "Mentorship is very useful by various perspectives like it gives confidence, helps to clear concepts of various subjects such as DBMS, programming etc., also gives hands on experience of leading the people. The only problem I faced was the time management especially by the job holder students because they don't have enough time to participate in an OSS project at regular basis".
Reply 2: "Mentorship is a very good approach from the perspective of new students. It provides them an opportunity to act like a teacher".

Teacher's achievements: Teachers acted as developers of the OSS project and guided students in developing and improving various coding techniques and UML models. The achievements of teachers were as follows:

1. They learnt the skills to guide students of an e-learning university. Students were distributed throughout the country and were not in immediate contact of a teacher. Therefore, teachers developed and designed various tutorials and guides of installing the OSS projects like Mantis, visual SVN Postgre SQL etc.
2. They also learnt and improved coding and developing skills by contributing in the OSS project. The most exciting experience was the reverse engineering of various modules of the OSS project and interaction with the community of the OSS project.
3. They had an opportunity of a vast exposure and creating links with international developers.

Teacher's Feedback: As a feedback, teachers recommend other educational institutes to adopt a similar methodology and indulge their students in an OSS project. There were many benefits of involving students in an OSS project like they learnt object oriented techniques, for example, the practical experience of creating relationships of associations, composition and aggregation.

Challenges of the OSS Project

Student's response in distance/e-learning mode: In a distance or e-learning mode when students did not find their teachers in front of them, they showed laziness and usually respond late. Therefore, to monitor performance and activities of the students in such a mode is really a challenging job and teachers feel lack of control and grip over the students.

Load shedding: This was one of the noteworthy issues faced while supervising students. Lack of uninterrupted power supply (UPS) especially on the end of students was also one of the major reasons of responding late.

OSS Community Response / Patch Acceptance Response: OSS developers or community (owners) also worked in an e-learning environment. Therefore, because of lack of direct communication between the developers and the community of an OSS model, efficiency affected even if there was a dedicated team at the owners' end.

Setting up Infrastructure: Immense efforts were required to setup infrastructure and resolving various configuration issues. We spent almost one month to install and configure AdempiereERP.

DISCUSSION

E-learning, has reshaped the frontiers and techniques of distance learning by enlarging its scope and deepening its interconnectedness (Dabbagh and Bannan-Ritland, 2005). It has removed the barrier of distance and made it very easy to learn courses from well known instructors of reputable universities which was previously not possible (Navarro and Shoemaker, 2000 and Shahin *et al.* 2008). Researchers have proposed a methodology for e-learning students to improve software engineering skills by working with highly experienced professionals. The proposed methodology is based on e-learning pedagogy and has helped final year students of the universities to improve their software designing and development skills. Previously proposed methodologies such as "An Activity-Level Internet-Based eLearning Model For University Level Courses" (Shahin *et al.* 2008), "A Theory-Based Design Framework" (Dabbagh, 2005) and The Funnel Model (Madar and Willis, 2014) did not focus on the pedagogy of learning software designing skills. Shahin et al. (2008) proposed an e-learning model which described a framework for university level regular courses. They did not describe the pedagogy of working on final year projects. Dabbagh (2005) presented a theory based framework. It is a very good approach for learning regular university courses but it does not provide a single step of completing final year project and improving coding skills. Madar and Willis (2014) proposed methodology that was focused on the pedagogy of improving software development skills of students by indulging them on OSS projects. This approach describes the steps of selecting open source software, configuring and setting up infrastructure and guidelines to work on open source project. Currently, no methodology is available which describes an approach of get involved in the open source projects. We tested the proposed methodology in two ways. First, skills of the same 10 students were tested after having worked on independent and OSS projects (Fig-1). The points obtained after working on independent project and OSS projects showed that students obtained more knowledge and skills by working on OSS projects. The same has been shown by He *et al.,* (2013) in the sense that hat OSS community and developers share the latest knowledge and improve technical skills. In the second experiment (Fig-2), two groups of students were developed. One group worked on independent projects and the group worked on OSS projects. The group who worked on OSS projects secured more points. Results support the study of Alexandre (2003) in the sense that working on OSS projects enable a user to improve all software engineering practices without having a paid instructor.

Conclusion: The purpose of the study was to develop an approach for getting engaged students of e-learning institutions in the selected OSS projects to improve programming skills by coordination and working with the highly skilled professionals around the world. The proposed study was a comprehensive approach to select an OSS project based upon the predefined parameters. This method also guided in the selection students, mentors and set up the required infrastructure. Feedback of the students who contributed in the OSS project recommends the proposed methodology. There were a lot of returns of working on an OSS project. Firstly, students have an opening to gain experience of a team work by working with highly experienced developers and analysts. Secondly, they enjoy a genuine taste and observation of implementing the object oriented programming techniques like generalization, specialization, polymorphism, method overloading, method overriding, composition and aggregation etc. They also learn techniques to develop and design different UML models and diagrams in a professional way. One of the primary benefits of deciding to work on an OSS project was that students didn't have to worry about the licensing problems of tools and technologies they want to use for the development purposes.

Acknowledgement: We are very grateful to ICT and RnD, Ministry of Science and Technology, Pakistan for her generous financial support for this study.

REFERENCES

Ahmed, F., P. Campbell, A. Jaffar and L. F. Capretz. Managing support requests in open source software project: The role of online forums. In Proc. of 2nd IEEE International Conference on Computer Science and Information Technology.590-594. (2009).

Alexandre, G. "The (LA) TEX project: A case study of open source software." TUGboat 24:132-145. (2003).

Andrew J. T. The Development and Use of Open-source Spacecraft Simulation and Control Software for Education and Research. 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06). 330-336. (2006).

Bachmann, A. and A. Bernstein. Software process data quality and characteristics: a historical view on open and closed source projects. Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops. (2009).

Conley, C. A., and L. Sproull. Easier said than done: an empirical investigation of software design and quality in open source software development. In System Sciences, 2009.HICSS'09. 42nd Hawaii International Conference on (pp. 1-10). IEEE.

Dabbagh, N. Pedagogical models for E-Learning: A theory-based design framework. International Journal of Technology in Teaching and Learning,1(1), 25-44. (2005).

Dabbagh, N. and Bannan-Ritland, B. Online learning: Concepts, strategies, and application. Prentice Hall. (2005).

DeKoenigsberg, G.. How Successful Open Source Projects Work, and How and Why to Introduce Students to the Open Source World. Software Engineering Education and Training. (2008).

Ellis, H., R. A. Morelli, G. W. Hislop, T. Coll and D. Dublin. Work in Progress - Challenges to Educating Students within the Community of Open Source Software for Humanity. Frontiers in Education Conference. S3H-7 - S3H-8. (2008).

Ferenc, R., I. Siket, and T. Gyimothy. Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction. IEEE Trans. Software Eng., 31(10):897-910. (2005).

Ford, R. Open vs. Closed: Which Source is More Secure? ACM QUEUE 5(1). (2007)

Hassan, G.. “Designing concurrent, distributed, and real-time applications with UML”, Proceedings of the 28th international conference on Software engineering, Shanghai, China. (2006).

He, Z., Peters, F., Menzies, T., and Yang, Y. Learning from open-source projects: An empirical study on defect prediction. In Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on (pp. 45-54). IEEE. 2013.

Hissam, S.A., D. Plakosh and C. Weinstock.“Trust and vulnerability in open source software”, “Software, IEE Proceedings”, 149:47 – 51. (2002).

Madar, M.J. and Willis, O. Strategic Model Of Implementing E-Learning. International Journal Of Scientific & Technology Research 3:5 (2014).

Navarro, P. and Shoemaker, J. In M. G. Moore & G. T. Cozine (Eds.), Web-Based Communications, The Internet and Distance Education. University Park, PA: American Center for the Study of Distance Education: The Pennsylvania State. (2000).

Ossher, J., S. Bajracharya, and C. Lopes. Automated Dependency Resolution for Open Source Software.“Mining Software Repositories (MSR)”.7th IEEE Working Conference. 130 – 140. (2010).

Polancic, G.., R.V. Horvat and T. Rozman. Comparative Assessment of Open Source Software Using Easy Accessible Data.26th International Conference on. Information Technology Interfaces. l.1: 673 – 678. (2004).

Sahraoui, S.M. ICT governance in higher education: Case study of the rise and fall of open source in a Gulf university. International Conference on Information and Communication Technologies and Development.17-19. (2009).

Schryen, G.. A. Comprehensive and Comparative Analysis of the Patching Behavior of Open Source and Closed Source Software Vendors, IT Security Incident Management and IT Forensics, Fifth International Conference on IT Security Incident Management and IT Forensics. 153-168. (2009).

Shahin, G..O., Singh, D., and Wah, T. Y. An Activity-Level Internet-Based eLearning Model For University Level Courses. InProc. of the 24th South East Asia Regional Computer Conference (2008).

Tamura, Y. and S. Yamada.A Method of Reliability Assessment Based on Deterministic Chaos Theory for Open Source Software. Secure System Integration and Reliability Improvement. Second International Conference on Secure System Integration and Reliability Improvement.60-66. (2008).

Xiong, C.J., Y. F.Li, M. Xie, S.H.Ng, and T.N. Goh. A Model of Open Source Software Maintenance Activities. Industrial Engineering and Engineering Management. 267 – 271. (2009).