Engaging Students in Open Source Software for Learning Professional Skills in Academia

Mrs. Manpreet Shrivastava
R.D.V.V. Jabalpur, M.P.
manpreetshrivastava07@gmail.com

Abstract— The digital world makes human life more comfortable. This digital world has been created by the software. It seems to be that availability of this software in academia is not up to the mark. To enhance the role of software in academics there is a need to use open source software. The Open source software is popular as it provides the source code to the end user. Here the source code is available for user to change or modification for improvement from its original design. Most of the time, it is free of charge with the help of General Public License (GNU). This paper is a study of basic information of OSS, various development models for OSS, the use of OSS in the area of academics to improve the education system. This paper also discusses on previous work of the current scenario.

Keywords*- OSS, Proprietary, SDLC models*

*****

I. INTRODUCTION

Software engineering is the core of the computer science curriculum. The most important part of a software engineering course is the semester project. The motive of this semester project is to give students practical industry relevant software engineering knowledge, but the target is not fulfilled due to lack of industrial experience. It also seems that the software programs are not in the access of students. OSS is the solution to this problem. It is an interesting source for software engineering education. By participating in OSS projects students can improve their programming, design and communication capabilities. Open Source Software (OSS) development and use has increased significantly over recent years. Therefore, there is a need to analyze and understand these projects. Software can classify in two basic categories:

The open source software [1] is very popular now days. The most of government uses the Open Source Software. The US DoD is also a frequent user of OSS. The OSS has some special features to attract everyone. The OSS has the ability to give the response as soon as possible for the future threats because it has the ability to modify source code in unrestricted way.

Free and open source software (FOSS) is a mechanism in software world in order to provide the users a right to modify the existing software. This modification can enhance the efficiency of that software. The concept of open source was developed in the late 1980s by Richard Stallman of the Free Software Foundation.

OSS is defined as software whose source code is available and the user is free to use according to the OSS licensing policies. Due to the better connectivity of the world, the development of the OSS has grown quite rapidly. It is a broad cause for the availability of change in the OSS. OSS is always created by a group, so it is also available to all, while there are some software that is made by an industrial institution. These are called closed software.

In today's crucial time, it is very important to understand that concealing information is as much as necessary and also an important issue to share the information. This can be estimated from the fact that the closed source software does not share any information with anyone and because of this the effectiveness of learning is influenced. Before starting the detailed description there is a need to understand some basic terminology related to open source software in order to make proper difference [5]. Before starting the detailed description there is a need to understand some basic terminology related to open source software in order to make proper difference [5]

Table 1 Basic Terminology

Source code	The source code is simply a text file on which the programmer or coder write the sequence of instructions to implement the software. The source code of software may have multiple text files.
Freeware	These software have the properties that they are freely accessible but the source code does not available for users.
Shareware	Distribution of shareware software is free on the basis of trial use. Here the provider expects that the user will purchase the software after they understand the use or need.
Open Source Software	The OSS is the software having the availability of source code for users. The source code may or may not be free. The modification on code might possible or not it all depends on the license agreement.'
Free and Open Source Software (FOSS)	This is similar like the OSS having all the flexibility of OSS but here FOSS is providing the source code without any cost.
Free-Libre Open Source Software (FLOSS)	The basic definition freedom to access of source code of software to modified and update without any permission is FLOSS.

Proprietary software or closed source software

This is the category [1, 2] of software that is conventionally licensed. In these types of software there is a need to pay to the software provider. It means the user must take the ownership of the software before using it. Due to this it become the limited or specific user software. In Proprietary software the source code of software is hidden from user and not editable. It will not possible to make changes in source code because in some way there is a buying just for the access of the software not modification. In short the Proprietary software is only for use after buying. The word "Proprietary" software is also used as an antonym for open source software.

Open source software

Open source refers to a program or software [1, 3] where source code is available to the general public to use and / or modify its original design at no cost. Open source code is usually created as a collaborative effort where programmers improve code and exchange changes within the community. The reason for this move is that a larger group of programmers who are not interested in property ownership of financial gains will produce a more useful and error-free product for everyone to use. This concept is based on peer review to find and eliminate errors in program code, a process that commercially developed and packaged software does not employ. [4]

Table 2 Difference between CSS & OSS

Properties	Proprietary / Closed Source Software	Open Source Software
Transparency	No	Yes
Community structure	No	Yes
Creativity	No	Yes
Peer efforts	No	Yes
Ease of use	No	Yes

Advantages of OSS

Open Source Software [6, 11, 12] has the large number of advantages but few of them are listed here. Figure 1 shows the tree structure of open source software.

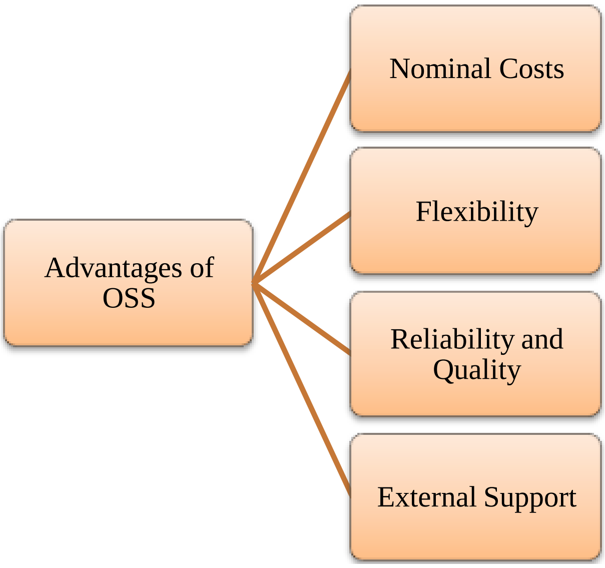

Figure 1 Advantages of OSS

Nominal Cost: In general, there is no need of licensing fees in OSS. Their nominal cost is one of the major advantages that this software has been accepted by the small businesses.

Flexibility: In OSS there is flexibility for user that he can modify the software packages as per their needs. Most of the time it seems to be that there is need of a programmer to add the function to open source software.

Reliability and Quality: Open source software now a day is very reliable due to its security features. As far as the quality is concerned it gradually increases.

External Support: External technical support has been available for most OSS packages. Open source products have their online services. In these services, the online community admits a lot, and here inquiries can include questions and answers through an online blog.

II. OSS IN ACADEMIA

The academics are a major area in which the open source software can play an important role. It seems to be that the most frequent user of any software is a student. But there is a need to pay for most of the software. This system deters the student from learning new approaches. As the brain of student is very fertile for new innovations. Here the open source software provides all the source code to enhance its quality and feature of a software program.

The open source community serves a lot to the academic community. The academics is an especially promising area for shareholders because students often turn to engineering programs when wanting to change the world, solve marvelous riddles or pursue profitable careers. These motivations combine well with the rewards that can be achieved by getting involved with free and open source software.

Basically there was no support for this model, either academic or technical leadership, in our scenario. University leaders do not want to create a large organization focused on this space, they are very careful to eliminate popular innovation. The most advanced projects in the field of public health and safety services are more independent and will hesitate to include them in said organization. Several people expressed concern about how resources were allocated and how priorities were identified for the needs of different institutions.

Thanks to open source software development movement, today significant number of development tools available to us. Most of such tools are free and very well supported.

- VLC Media Player
- MYSQL
- PHP
- Mozilla Firefox
- Office software
- PDFtk
- PsychoPy

Table 3 Application with License name [9, 10]

Application	License
Apache	Apache
Bash	GPL
Cygwin	GPL
gawk (awk)	GPL
GCC	GPL
LaTeX	LaTeX
Linux	GPL
Linux (Red Hat)	GPL
OpenOffice	GPL

These tools are very important for the daily uses of every one. Having some extra feature academic can improved the current scenario.

OPEN SOURCE TOOLS IN EDUCATION

Some productivity suite has been used by every educational staff and students. In this manner the word processing tool MS office is very popular. In OSS the Libre office, Open Office, AbiWord are famous. These tools work same as the MS office but these are free to access.

The Web browser is also a very useful tool to use the internet. Internet explorer is a product by the Microsoft but Firefox is a popular internet browser used by everyone. The Firefox is an OSS.

The most educational open source software are KGeography, Scilab, QCAD, Moodle. All the tools are famous in their respective area.

ROLE OF OSS IN ACADEMIA

There are several roles which will work for the students. The student can take advantage using OSS. Some of the major points have been discussed here. Figure 2 shows the classification chart of the Roles of OSS for Student Community

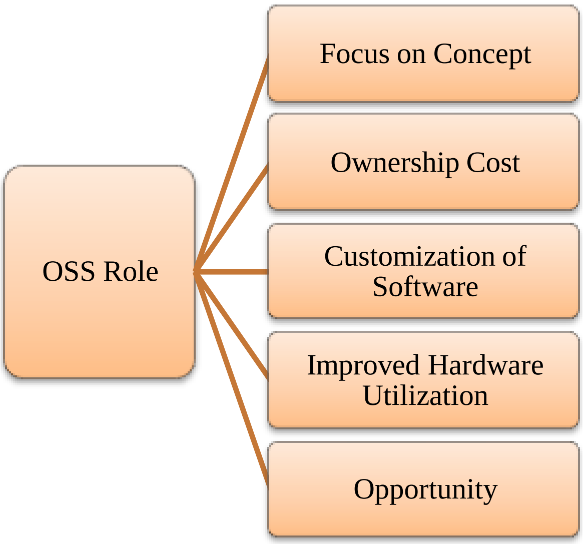

Figure 2 Roles of OSS for Student Community

Focus on Concept: Using OSS student doesn't need to pay attention on the feature of the product or software. Here the student can only focus on the concept of the software. The logic of students will more improve.

Ownership Cost: The cost of commercial software is very high which is not fulfilled by the students. The OSS provides the cost effective ownership of software.

Customization of Software: Customized software can be changed very easily. Most of the software is installed once in your system. No change can be made in their source code. Because of which no new changes can be made in software. Here, you can change source code with the help of open source software.

Improved Hardware Utilization: Open source software is designed to be easy to use in older hardware too. There is no need to change the hardware repeatedly. This property is very useful for students.

Opportunity: The world is expanding towards open source software. One thing is certain that this area is expected to get maximum openness.

III. OSS Development Model

The software development has been done in a specific system development life cycle (SDLC). There are many models which has been proposed with different properties. Normally requirement gathering, analysis, development, testing and maintenance are the basic phases of SDLC. [1, 8,13]

Open source software development is a little bit different. So the development approach is also different. One of the models has been proposed by the DoD USA which is shown in the figure 3. In this model five phases has been shown.

Developer is the coder who has the concept of the Open source software. This OSS has been also implemented by the developer. After developing the software Developer will submit to the trusted developer.

Trusted developer is an authority who will receive all the development provided by the developer. This is third party person so it increases the trust. He has the authority to make modification after the correction or updating the submission will goes to trusted repository.

Trusted repository is also a faithful authority which contains all the OSS. This will provide the software to distribute. The trusted repository is a digital library in order to provide the reliable and long term digital resources.

Distributor is a author to serve the product to the end user and provide the benefits.

User is person who accesses the Open Source Software. During the use if any problem occurs or some update needed then user can report to trusted developer, trusted repository and Distributor directly to enhance the efficiency of software.

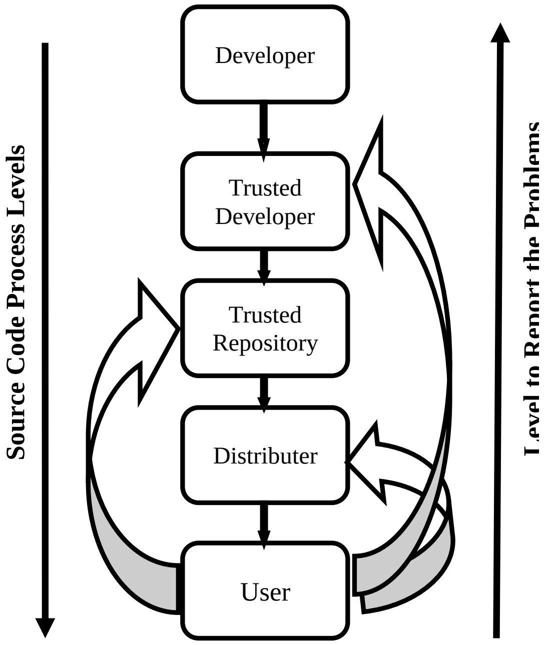

Figure 3 Open Source Software development model by DoD USA

As shown in figure 3 users have the rights to report the problem regarding error or abnormal code execution to the higher authority. User can directly report to distributer, trusted authority, trusted developer. In this scenario the solution will generate as soon as possible.

One of other model has been proposed by the author Jorgensen. [14] In his model following steps has suggested. Figure 4 shows the Jorgensen Model. The major steps have been discussed below:

Code: Code is basic entity which is provided by the coder. This code will review and also improved if necessary.

Review: The review is an important phase for all codes .This code is already given by the code phase.

Pre-commit test: After the completion of review process the testing will take place. The testing will done in the sequence manner.

Development release: Whenever the code need to provide to the end user as an application there is a need of development release platform.

Parallel debugging: During the use of application we may get flaws. In this phase those flaws will recover as debugging.

Production release: This is a final version release which gives us stable product.

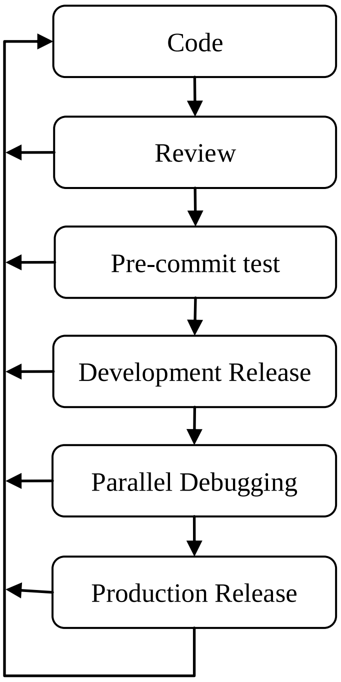

Figure 4 Life-cycle model by Jorgensen

Figure 5 shows the Rinette Roets Model [15]. This model also worked as a Jorgensen but in another way. Steps are similar like Jorgensen Model but are a comparative analysis with the SDLC. This model shows the comparative analysis of the Software Development Life Cycle and the Open Source software Development life cycle. It seems to very clear the SDLC and OSS-SDLC both work in different manner.

There are five steps in software development life cycle model. Here there are five steps (Planning, Analysis, Design, Implementation, and Support) which have been categorized in three major steps. These three steps are similar to the Open source Software development life cycle. In other words Planning, Analysis, Design of SDLC is equivalent to the Initiation of OSS SDLC

Similarly the step Review, Contribution, Pre-Commit Test and Product Released of OSS SDLC are the equal step of Implementation of SDLC. The final last step of support of SDLC is work as Parallel Debugging and Development Release of OSS SDLC.

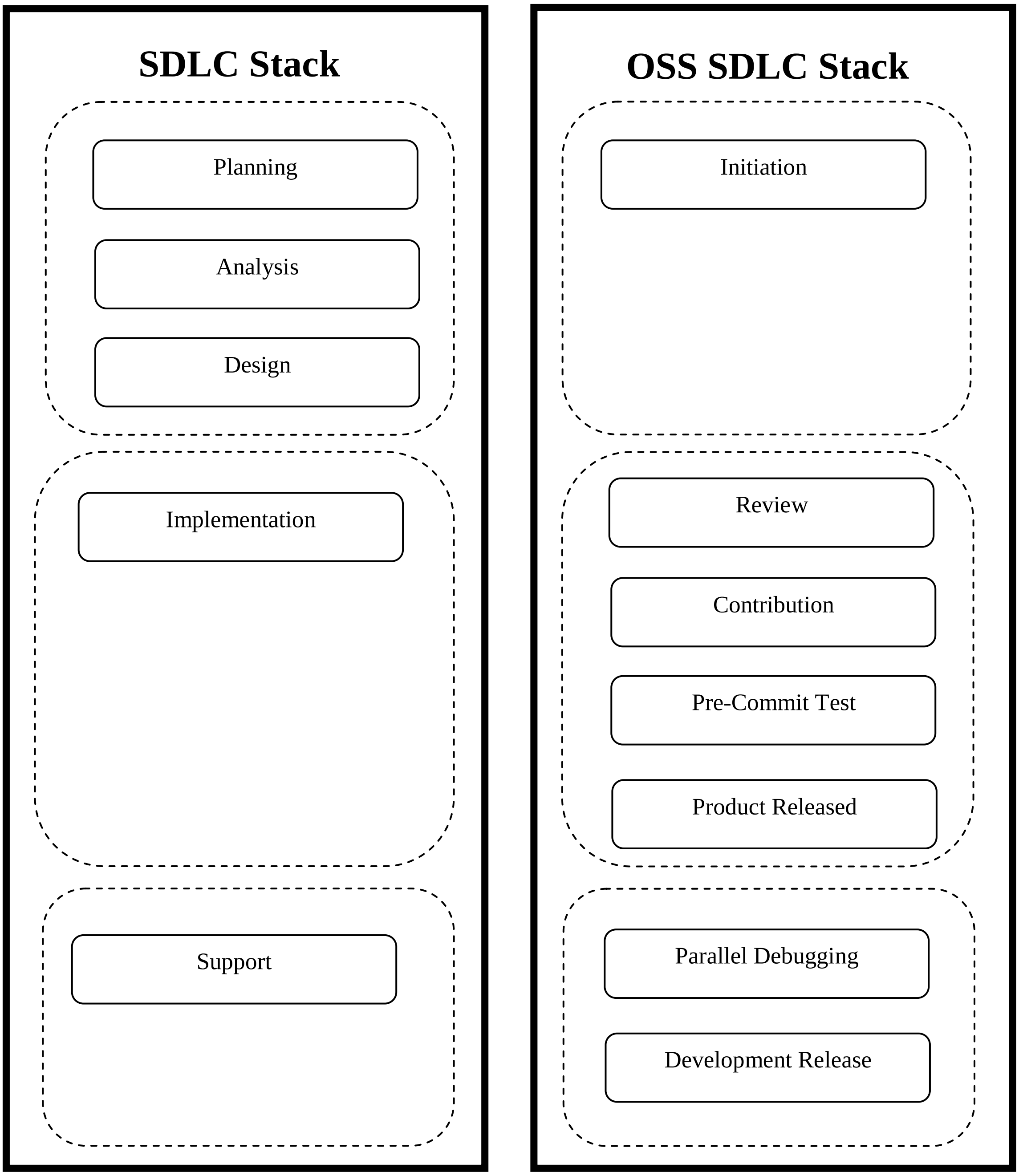

Figure 5 Life-cycle model by Rinette Roets

IV. Literature Survey

In the field of open source software development [17] (OSS), OSS developers are primarily responsible for the supervision and management of implementation activities. Based on the qualitative analysis of Internet-based artifacts from a few OSS projects, the current literature of the Office of Internal Oversight Services shows that the generation of requirements for the development of the operational support system is largely informal and ad hoc. But the sometime due to unavailability of experimental study involving practitioners themselves, that is known as the open source software development developers. The author conducted an online survey among the OSS developers to obtain information on how the required engineering activities were carried out and what their perceptions were. With regard to 57% of the engineering practices obtained from the closed-source software development literature (CSID).The author also asks open OS developers to share their perceptions about the use of the five activities that generate the unofficial requirements in OS. The subsequent analysis revealed that OSS developers use formal requirements generation activities far more than CSID requirements in their development activities. The author used the results of the survey to analyze the effects of practice and research.

The author [18] has tested the various learning curves of individual software developers in the development of open source software (OSS). The author has also collected the data set from repositories. Here the 200 developers were working on the 20 open source software projects. They build and appreciate models of regression to evaluate the individual learning progress of developers. Author rating results indicate that developer coding and indirect debugging experiments do not reduce bug ratios while the experience of fixing errors can lead to a low impurity ratio of learning progress.

Learning with experience [19] enhanced the productivity. There are number of models and approaches of learning curve have still developing and have been successfully applied to traditional industries. A new model for quantitative learning of the software life cycle is proposed and compared to existing learning models. A formal business model for testing programs is modified to include learning effects based on the developed learning model. Initially, the testing process is applied to several industrial software testing projects to validate the improved prediction capabilities of the model.

Software engineering [20] trainers face many challenges. Among these challenges is the draft session. Trainers are required to train their students on professional skills to be ready for business in the real world, which requires students to work on real projects. However, due to the poor quality of student work, not all professional organizations cooperate to provide an opportunity for software engineering students to work on a real project. Therefore, most of course software engineering projects are class projects, in which the trainers represent customers. In this document, the author proposed solving this problem based on his experience in engaging software engineering students in real world projects.

The author [21] has implemented an Open Sources software search engine with respect to dolphin which is a proposed approach. This search engine continues services online. The experimental results and studies shows the proposed method provides the good results.

In paper [22] the author has studied about the dependencies of various packages. They also analyzed that to find the solutions how the dependency graphs of npm, CRAN and RubyGems which are the three large packaging ecosystems develop with the time.

The author [23] have worked on the ad-hoc network with the Commercial Off the Shelf that is COTS nodes like mobile phones and tablets with the help of open source software (OSS) . The author has modified the TrevE-Mod WiFi Tether application over the ad-hoc network.
The article [24] proposed a approach to get the difference between the structural using call graph. This result will visualize. The author has applied the method and it seems to be that this approach speedup the application process. The OSS has been used which also reduces the update error.

V. Conclusion

The rapid growth of computer science has increased the responsibility of Software system. To improve the efficiency and performance of software there is a need of a mechanism by which any one can update or modified the specific software. This facility can be achieved by the open source software. To provide this facility in academics the performance of software has been improved. This paper shows the various models in order to develop the OSS. Here the application of OSS has also discussed and how this approach can be used in other application area.

REFERENCES

[1]. The MITRE Corporation "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense" version 1.2.04, 2003

[2]. Vinay Tiwari "Some Observations On Open Source Software Development On Software Engineering Perspectives", International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010

[3]. Liu,C "Enriching software engineering courses with service-learning projects and the open source approach" Proceedings of the 27th international Conference on Software Engineering, ICSE '05.ACM Press,pp.613-14.

[4]. Faber,B.D.(2002),"Educational models and open source: resisting the proprietary university."Proceedings of the 20th Annual international Conference on Computer Documentation , SIGDOC '02', ACM press, pp 31-38

[5]. Carrington, D and Kim S. (2003), "Teaching Software Engineering Design with Open Source Software." 33rd IEEE Frontiers in Education Conference, Nov. 5-8, Boulder, CO.

[6]. German, M.D. ”Experience teaching a graduate course in Open Source Software Engineering “Proceedings of the first International Conference on Open Source Systems.Genova,pp.326-328

[7]. Sowe S. K. Stamelos, I., Deligiannis , “A Framework for Teaching Software Testing using F/OSS Methodology”. International Federation for Information Processing, Open Source Systems, Vol.203, (Boston: Springer), pp.261-266.

[8]. Sowe, Sulayman K.& Ioannis Stamelos,” Involving Software Engineering Students in Open Source Software Projects: Experiences from a Pilot Study”, Journal of Information Systems Education(JISE),2008 .

[9]. Kamna Solanki, Sandeep Dalal and Vishal Bharti, “Software Engineering Education and Research in India: A Survey” International Journal of Engineering Studies, 2009,pp.181-192

[10]. Timothy C. Lethbridge. The Relevance of Software Education: A Survey and some Recommendations. Annals of Software Engineering,6:91-110,1998

[11]. R. K. Pandey: FOSS Projects: A Bridge Between The Industry And The Academia, Journal Of Information, Knowledge And Research In Humanities And Social Sciences,ISSN:0975-6701,Nov10 to Oct 11

[12]. Rajendra K.Raj and Fereydoun Kazemian: Using Open Source Software In Computer Science Courses,36th ASEE /IEEE Frontiers in Education Conference, 2006

[13]. Ju Long “Open Source Development Experiences On The Students Resumes :Do They Count?-Insights From The Employers Perspectives” , Journal of Information Technology Education,2009

[14]. Jorgensen, N., (2001), Putting it all in the trunk: Incremental software development in the Free BSD open source project. Information Systems Journal, 11(4), 321-336

[15]. Rinette Roets, Marylou Minnaar, and Kerry Wright, (2007) Open source: Towards Successful Systems Development Projects in Developing Countries, Proceedings of the 9th International Conference on Social implications of computers in developing countries, Sao Paulo, Brazil, May 2007

[16]. Fakhar Lodhi: Involving Students In Open Source Software Development As An Alternative To The Capstone Project,ICCIT 2012

[17]. J. Kuriakose and J. Parsons, "How do open source software (OSS) developers practice and perceive requirements engineering? An empirical study,” IEEE *Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, Ottawa, ON, 2015, pp. 49-56.

[18]. Y. Kim and L. Jiang, "The Knowledge Accumulation and Transfer in Open-Source Software (OSS) Development," *48th Hawaii International Conference on System Sciences*, Kauai, HI, 2015, pp. 3811-3820.

[19]. G. Abu, J. W. Cangussu and J. Turi, "A quantitative Learning Model for Software Test Process," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005, pp. 78b-78b.

[20]. Zakarya Alzamil “Towards an Effective Software Engineering Course Project”, Proceedings of the 27th international conference on software Engineering,ICSE ’05.ACM Press, pp.631-632

[21]. Yun Zhan, G. Yin, T. Wang, Cheng Yang, Zhixing Li and H. Wang, "Dolphin: A search engine for OSS based on crowd discussions across communities," *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, 2016, pp. 599-605.

[22]. A. Decan, T. Mens and M. Claes, "An empirical comparison of dependency issues in OSS packaging ecosystems," *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Klagenfurt, 2017, pp. 2-12.

[23]. A. A. Malik, A. Mahboob and T. M. Khan, "Implementing MANET for Trustworthy Collaboration Using OSS and Android Based COTS Devices," *2016 International Conference on Collaboration Technologies and Systems (CTS)*, Orlando, FL, 2016, pp. 485-492.

[24]. M. Fukuyori, R. Umekawa, H. Fujino, J. Yamaguchi and K. Ariyama, "An Approach Based on Structural Differences to Expedite Applying OSS Updates to Products," *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Turin, 2017, pp. 445-450.