Open-source software in education: a modern review of a case study

Bassel Walid, STEM High School for Boys – 6th of October

Abstract

Since the rise of technology, many of the new folds of the new generations have been attracted to programming and computer science as their majors. This gave rise to large numbers of programmers all over the world with different skillsets and programming experiences. Open-source software (OSS) comes as a natural result of the wide variety of ideas that come to the minds of programmers and the collaborative nature of the field that emphasizes the creation of bigger software from smaller pieces of free and simple pieces. Open-source software has come a long way since the first piece of major OSS (the Linux kernel and its development [1]) and its importance in education has increased, this why this paper acts as a review of the current progress of OSS in education.

I. Introduction

Since the inception of OSS (open-source software), it has been a major source of revenue from operations in many places from small startups to international multimillion-dollar conglomerates and has become a popular stable in IT departments and any successful IT toolkit.

With the Internet's arrival in the early 1990s, programmers from various locations around the world have been able to work together and distribute, share, and collaborate on software easily. Along with the various advantages of OSS, the vastly improved ability for developers to communicate has increased the outreach of OSS.

The previously mentioned popularity of OSS gave rise to many important pieces of OSS that became a major part of modern-day software such as the Linus kernel and the GNU compiler collection -the C++ compiler in which is one of the most used-, Python, VLC, Mozilla's Firefox, and Blender.

In this article, the role and position of OSS in education are discussed. First, a history of OSS is given with the story of the first piece of OSS. Second, a case study on using OSS in education is studied and reviewed. finally, then compared to the possible outcomes if the same experiment is repeated in the modern-day.

II. History and Background

Since its creation, the history of open-source software has been filled with many achievements, failures, and unique events that give it its rich, vibrant and, interesting background.

The history of OSS starts with Richard Stallman and the decline of free software back in the 1960s. back in the early days of early consumer electronics, most pieces of software were distributed in the form of physical media which contained both the source code (the human-readable code) and the machine code to execute these programs, the reason why both versions were backed was that early software needed to be modified by the user to run on different machines and systems. With the rise of computer operating systems, software development costs started to increase significantly relative to the

development costs of hardware at the time, the increase in costs led to a slew of antitrust and copyright lawsuits to go through in the technology field. After the dust had settled many if not most of the software released was released without its source code and was only released with the necessary machine code to run it. This trend started to worry programmer and future free software activist Richard Stallman. He worried that now without legal access[*] to source code further collaborative modifications of software would not be possible. In 1983 Richard Stallman started the GNU (Gnu's Not Unix) project in hopes of countering the closed-source trend. He also started the Free Software Foundation and invented "copyleft". Copyleft was made to be the opposite of copyright for software, in it the freedom of contribution, modification, and redistribution of software was protected. He implemented copyleft into GNU's General Public License which required any derivative work on software to also follow the same license, preventing any modifications of software to be turned into closed-source software.

The term "open-source" was first coined by Stallman and another prominent programmer who had the same set of beliefs (such as Linus Torvalds) in hopes of appealing to bigger companions. The Open-Source Initiative was founded to further spread the usage of the term. The coining of the term came shortly after Netscape (a prominent internet company at the time) released the source code for their web browser in hopes of programmers helping them grow it further.

When talking about the history of OSS, one cannot Linus Torvalds, the original creator of Linux (a Unix-like system) who added his creation to the OSS pool, accelerating Stallman's vision of fully open-source computer life. Linux is now one of the most popular operating systems, due to its open-source nature comes in tons of varieties that serve multiple purposes from normal desktop use to server operating systems to the main operating system for cybersecurity testing.

III. OSS in Education: Case Study by Swansea University [2]

i. Prerequisites and background

In the case study presented in the paper "*Open-Source Software in Computer Science and IT Higher Education: A Case Study",* which was published in 2011, students in Swansea University were given a completely open-source setup that included a full IT setup starting from the text editor for the students to the operating system of the server and the database software for the said server. The academic year included three main courses: Data Structures and Algorithms using Java, Rapid Java Application Development (an advanced Java class), and Design and Analysis of Algorithms.

The presented goals for the classes themselves were for the teacher to have to devices (Desktop and laptop) that were linked together securely through a server which also served both static and dynamic pages to the students. The second goal is for the teacher to have a wide range of software to fully deliver a successful class, the software kit had to be fully open-source and had to include the basics such as a web browser, text editor, and an email client and also advanced software such as an IDE (integrated development environment) and a program to create presentations and diagrams for the students.

ii. Used Software

For the case study the software used was as follows:

- Mozilla Firefox as the web browser of choice and Evolution as the mail client allowing for mail organization and calendar management.
- LaTeX, was a big part as it was chosen as the word processor and the presentation software's base (Prosper), this was due to its advanced nature and multitude of features.

[**] (compiled machine code of a lot of programming can still be converted back into source code using decompilers, but it is illegal under copyright laws that prevent reverse-engineering of software)

- Prosper, a LaTeX based presentation package with important features such as incremental display, overlays, transition effects. Xfig was used as a drawing tool for creating vector graphics.
- For the IDE Netbeans IDE was used to edit and debug Java, JSP, and HTML code.

iii. Results

In the original paper reviewing the case study, three aspects were observed when the results were collected.

First was cost, open-source software by nature comes free of charge. So, to calculate the cost difference the OSS setup was compared to an equal educational functionality setup using normal proprietary software (shown in table 1)

Function	Application	Cost (USD)
Operating System	MS Windows XP Professional	300
Web Server	IIS (Windows Server 2003, Std.)	1000
Firewall	(Windows Server 2003, Std.)	0
Encrypted Communic.	(Windows Server 2003, Std.)	0
Database Server	MS SQL Server	1500
Source Control	(Visual Studio .NET 2003 Enterprise)	0
Web Browser	MS Internet Explorer	0
Email Client	Outlook (MS Office 2003)	0
Word Processor	Word (MS Office 2003)	150
Presentation Program	Powerpoint (MS Office 2003)	0
Drawing Tool	MS Visio Standard 2003	200
IDE	Visual Studio .NET 2003 Enterprise	1800
	Total	4950

TABLE 1: THE COST OF OSS SETUP

The results in cost were that using OSS reduced the individual cost of employing significantly but had the chance to raise administrative costs depending on whether further training to use the OSS was required or not.

Second came student appeal, the paper found that there were two main reasons why OSS might appeal to a university student. First, was that cost reduction did not just affect the professor and the school but also the student, the reason for which is because these same students might want to use the software they use in university at home or even in personal projects or small startups, with the cost reduction of OSS this significantly improves the promotion of entrepreneurship. Second, exposing computer science students to software that they can themselves change and modify according to their needs and wills is an immense boost to their hands-on experience in software development.

Third came the ease of use, in this aspect specifically the upper hand went to proprietary software due to it being more mature and more widespread to use.

IV. Results with Modern Improvements in Mind

In this part of the article, the three previously mentioned result aspects of the original paper are looked upon from a modern outlook.

First, the cost has not changed that much since the current plan for big technology companies to compete with OSS is to either give high educational discounts (where OSS still wins with its free nature) or to offer more stripped-down versions of their software lineups for free (ex. *free* Office online vs *paid* Office 365) where OSS still gets the upper hand due to the developers not looking for a profit through upfront cost and so adding the full feature set.

Second, the appeal for computer science has only gone up, this comes as a result of the ever-increasing nature of the software development and OSS industries where the more experience a student has with software development the higher the chance, they get employed post-graduation [3].

Finally, ease of use, which out of the three studied aspects is the one that faced the most positive change. The positive change of usability of OSS comes as a result of the general maturity of the OSS industry and the increased experience and numbers of OSS developers which led to higher quality and functional software that is easier to use. It also does not stop there as there have been many pieces of research that only serve to improve usability in OSS [4].

V. Conclusion

In conclusion, Open-source Software has improved significantly since the early days of its adoption, this resulted in it improving a lot on its downsides while keeping its main advantage (freedom and freeness of use) safe and secure. It also has come a long way since the writing of the original paper as shown in the modern look section of the article. The improvements in the field of OSS and the increasing importance of participating in it only serve to increase the importance of integrating it into modern higher-level education to help students prepare for the competitive work market and future endeavors.

VI. References

[1]O. Koren, "A study of the Linux kernel evolution," ACM SIGOPS Operating Systems Review, vol. 40, no. 2, pp. 110–112, 2006.

[2]D. R and R. S., "Open Source Software in Computer Science and IT Higher Education: A Case Study," International Journal of Advanced Computer Science and Applications, vol. 2, no. 1, Jan. 2011.

[3]G. Passaretta and M. Triventi, "Work experience during higher education and post-graduation occupational outcomes: A comparative study on four European countries," International Journal of Comparative Sociology, vol. 56, no. 3-4, pp. 232–253, 2015.

[4]K. A. Dawood, K. Y. Sharif, A. A. Zaidan, A. A. Abd Ghani, H. B. Zulzalil, and B. B. Zaidan, "Mapping and Analysis of Open Source Software (OSS) Usability for Sustainable OSS Product," IEEE Access, vol. 7, pp. 65913–65933, 2019.