

Open Source Software Alternatives in Higher Education Following Computer Science Curricula 2013

Borislav Stoyanov and Krasimir Kordov

Department of Computer Informatics, Faculty of Mathematics and Informatics, Konstantin Preslavski University of Shumen, 115 Universitetska Str., Shumen 9712, Bulgaria

Abstract: In this study we overview the open source software, describe the advantages and disadvantages of using open source software in modern Higher Education following Computer Science Curricula 2013. The study's main purposes are to clear the understanding of open source software, to present alternatives of the commercial software and demonstrate the potential benefits of integrating open source software in Higher Education.

Keywords: CS2013, higher education, open source software alternatives

INTRODUCTION

The Higher Education requires more active implementation of modern software products because of the rapid development of the new technologies. Other factor is integrating of eLearning in Higher Education in all universities in all levels of education, as stated in Laurillard (2006). That is why using always specific commercial software is not the best possible decision and we need an alternative. Open Source Software (OSS hereafter) considerably reduces the expenses and a lot of universities have taken this step and have successfully integrated teaching of open source software in their education.

Some OSS alternatives are pointed in Brocco and Frapolli (2011), Corbesero (2006), Herman and Lugo (2008), Lakhan and Jhunjhunwala (2008), Lipşa and Laramée (2011), Nandigam *et al.* (2008), Osunade (2012), Sharma and Koshy (2011), O'Hara and Kay (2003), Kamthan (2006), Wiley (2006) and Zaritski (2003).

This study aims to clear the understanding of open source software, to present alternatives of the commercial software and demonstrate the potential benefits of integrating open source software in Higher education following Computer Science Curricula 2013 (CS2013), refer to ACM/IEEE-CS Joint Task Force on Computing Curricula (2013).

MATERIALS AND METHODS

According to the files the users get, the software is:

Closed Source Software (CSS hereafter): Users receive only executable files. The source code is not shared with the public to read, modify or develop. The most of commercial software is distributed as closed

source software. Even some software is distributed as freeware and users can use it for free, users don't have any access to source code of the software.

Open source software: Users receive not only executable files but also the source code of the software. According to the license of the software users may be able to change, develop or distribute the software following license agreements.

Common mistake is to confuse open source software with free software and it is important to be familiar with types of software licenses.

Free BSD license: A very permissive license that allows most rights, as long as any redistributed products includes a copy of the license. In case of redistribution copyright holder cannot be changed (The FreeBSD Copyright, 2014).

GNU general public license (GPL V2 and V3): This is the most popular open source license. Allows for liberal usage, copying and modification. But products which redistribute GPL software must themselves be licensed with the GPL (GNU General Public License, 2014).

Apache license: Also a permissive license but with terms that specify how credit must be attributed (The Apache Software Foundation: Licenses, 2014).

MIT license: Perhaps the most permissive license, allows anyone to do anything as long as they include a copy of the license text (<http://opensource.org/licenses/MIT>).

Open source software: Open source software is computer software that is distributed along with its

source code (the program code that is used to create the software) -licensed with an open-source license in which the copyright holder provides the rights to study, change and distribute the software for free to anyone and for any purpose. However it is wrong to consider open source software as “free” that is why is necessary to carefully read the license under which the software is distributed. The license defines the rights of customers while using the software.

One of the best examples of open source software is the Linux OS (linux.org), which is the open source equivalent of UNIX. Other bright example is Android OS (android.com) (mostly used for smartphones and tablets). Firefox is also a good example of open source software. Content Management Systems (CMS) with open source take significant part of the web space: Drupal (drupal.com), WordPress (wordpress.org), Joomla (www.joomla.org), .. etc.

Advantages of open source software:

Lower costs: In times of global economic crisis this can be very significant factor for choosing software for business and education. Using open source software can reduce the expenses because of lower prices (compared to commercial software) and the fact that usually open source software is distributed free.

Security: Feedback from user allow the authors to fix the flaws found in the source code by the customers. The customers even have opportunity to fix the open software themselves and to share their experience helping to develop the open source software they are using. This also can save time waiting for the next release of the software.

Flexibility: Very often purchased commercial software does not respond completely to the needs of the customers (business or education) and users do not have a chance or right to change the software. Open source software provides the source code which can be changes to fit the exact need of a single user, company, school or university. This can improve education process or profit of the company.

“Long life” of the software: Open source is not dependent on the company or author that originally created it. Even if the company closes, the code continues to exist and to be developed by its users. Also, it uses open standards accessible to everyone. Thus, it does not have the problem of incompatible formats that exist in proprietary software.

Experience: Lower costs (usually no costs at all) of testing and working with different software for same needs provides educational experience for customers, employees, teachers or students. Learning how to

change the source code of open source software can provide useful experience for future IT specialists.

Profit: Modification and developing the open source software can provide the profit for programmers or companies working on improving open source software if the license agreement allows.

Disadvantages of open source software:

Need of experience: Working with new software can be difficult without some guidance explanations by experienced person familiar with the new software. Choosing the best open source for business or education can take some time until selecting the best software fitting the needs of business or education.

Hardware incompatibility: Sometimes open source software is incompatible with latest generation hardware. The solution is using third party drivers or waiting for new release of the software.

Programming knowledge: In order to make modification in source code the customer has to have knowledge on programming language used for software creation. If they are not programmers they have to rely on feedback with authors or developers of the open source software.

Possible hacking: All users have access to source code files and this may give opportunity of exploiting the software by unethical users. That is why is really important to keep your software updated to prevent hacking.

Advantages and disadvantages of open source software in higher education/conclusion: Using open source software in higher education can reduce the expenses of universities for software. Students have the opportunity to gain experience with new software which is very useful not only to the future IT specialists but to all who needs IT technologies for their future jobs. The tech supports of the universities are able to modify the open source software for specific education or students are able to do modifications themselves gaining experience. However the lecturers have to be familiar with the open source software used for education which can be a challenge and take a lot of lecturer’s time.

RESULTS AND DISCUSSION

Konstantin Preslavski University of Shumen case: Good example of successful implementation of open source software in higher education is CS (Computer Science) education in KP University of Shumen. For

Table 1: Open source software used for education

Knowledge area	Course	Open source software
Programming languages	Programming C++ (part I)	DevC++ (http://www.bloodshed.net/devcpp.html), Code::Blocks (www.codeblocks.org/)
Programming languages	Programming C++ (part II)	DevC++, Code::Blocks
Programming languages	Functional and logic programming	Racket (racket-lang.org)
Operating systems	Operating systems	Ubuntu (www.ubuntu.com), RedHat (www.redhat.com), Android
Networking and communications	Computer network and communications	GNS3 (www.gns3.net)
Software engineering	Data modeling	Argo UML (http://argouml.tigris.org/)
Software engineering	Project management	OpenProj (sourceforge.net/projects/openproj/), ProjectLibre (www.projectlibre.org/)
Discrete structures	Discrete mathematic	Scilab (www.scilab.org/)
Graphics and visualization	Computer graphics	OpenGL (www.opengl.org)
Platform-based development	Web technology	Apache, MySQL (www.mysql.com)
Platform-based development	Web based information systems	Apache, MySQL

Table 2: Recommended open source software for every knowledge area

No.	Knowledge area	Recommended open source software
1.	AL-algorithms and complexity	DevC++, Code::Blocks
2.	AR-architecture and organization	Hardinfo (http://sourceforge.net/projects/hardinfo.berlios/), Hardware Lister (http://ezix.org/project/wiki/HardwareLiSter), Sysinfo (http://sourceforge.net/projects/gsysinfo/)
3.	CN-computational science	Linear Program Solver (http://sourceforge.net/projects/lipside/)
4.	DS-discrete structures	Scilab
5.	GV-graphics and visual computing	Archimedes (https://forge.ocamlcore.org/projects/archimedes/), LibreCAD (librecad.org/), FreeCAD (freecadweb.org/)
6.	HC-human-computer interaction	Ruby (https://www.ruby-lang.org), Aptana (aptana.com)
7.	IAS-information assurance and security	TrueCrypt (truecrypt.sourceforge.net/), CrypTool (www.cryptool.org/), OpenVPN (openvpn.net/)
8.	IM-information management	Firebird (www.firebirdsql.org/), MySQL, eXist-db (exist-db.org/)
9.	IS-intelligent systems	Racket
10.	NC-networking and communications	GNS3, OpenVPN
11.	OS-operating systems	Ubuntu, RedHat, Android
12.	PBD-platform-based development	Apache, MySQL
13.	PD-parallel and distributed computing	DevC++, Open MPI (www.open-mpi.org/)
14.	PL-programming languages	DevC++, Code::Blocks, Racket
15.	SDF-software development fundamentals	DevC++, Code::Blocks
16.	SE-software engineering	OpenProj, ProjecLibre, Argo UML
17.	SF-systems fundamentals	Flat assembler (flatassembler.net/)
18.	SP-social and professional issues	TeXnicCenter (www.texniccenter.org/), TeXCAD (texcad.sourceforge.net/), OpenOffice (https://www.openoffice.org/)

the major courses the following open source software is used for education (Table 1).

Recommended open source software in higher education: According to the world standards defined by Computing Curricula Association for Computing Machinery and IEEE-Computer Society computer science in higher education there are 18 knowledge areas corresponding to topical areas of study in computing. In Table 2 we show recommended open source software for every knowledge area.

CONCLUSION

Open source software increases its popularity every day. The interest of IT companies, Government organization and small business brings more and more investments every year. That is why the universities cannot ignore the importance open source software and have to consider the future benefits of using and teaching open source software in higher education. The

best universities in the world have already taken this important step giving the best possible example for better higher education.

ACKNOWLEDGMENT

This study is supported by the Project BG051PO 001-3.3.06-0003 "Building and steady development of PhD students, post-PhD and young scientists in the areas of the natural, technical and mathematical sciences". The Project is realized by the financial support of the Operative Program "Development of the human resources" of the European social fund of the European Union.

REFERENCES

ACM/IEEE-CS Joint Task Force on Computing Curricula, 2013. Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press, DOI: 10.1145/2534860. Retrieved from: <http://dx.doi.org/10.1145/2534860>.

- Brocco, A. and F. Frapolli, 2011. Open source in higher education: case study computer science at the University of Fribourg, pp: 1-35.
- Corbesero, S.G., 2006. Rapid and inexpensive lab deployment using open source software. *J. Comput. Sci. Coll.*, 22(2): 228-234.
- GNU General Public License, 2014. Retrieved from: <http://www.gnu.org/licenses/gpl.html> (Accessed on: March, 2014).
- Herman, R. and G. Lugo, 2008. Open source resources for teaching and research in mathematics. *Proceeding of the 20th International Conference on Technology in Collegiate Mathematics*, pp: 107-113.
- Kamthan, P., 2006. Open source software in software engineering education: No free lunch. *Proceeding of the Canadian University Software Engineering Conference (CUSEC, 2006)*, pp: 80-84.
- Lakhan, S.E. and K. Jhunjhunwala, 2008. Open source software in education. *Educ. Quart.*, 2: 32-40.
- Laurillard, D., 2006. E-learning in Higher Education. In: Ashwin, P. (Ed.), *Changing Higher Education: The Development of Learning and Teaching*. Routledge Falmer, London, pp: 71-84.
- Lipşa, D.R. and R.S. Laramee, 2011. Open source software in computer science and IT higher education: A case study. *Int. J. Adv. Comp. Sci. Appl.*, 2(1): 10-17.
- Nandigam, J., V.N. Gudivada and A. Hamou-Lhadj, 2008. Learning software engineering principles using open source software. *Proceeding of the 38th Annual Frontiers in Education Conference (FIE, 2008)*, pp: S3H-18-S3H-23.
- O'Hara, K.J. and J.S., Kay, 2003. Open source software and computer science education. *J. Comput. Sci. Coll.*, 18(3): 1-7.
- Osunade, O., 2012. Nigerian computer science education and open source software. *Proceeding of the EIE's 2nd International Conference on Computer, Energy, Networking, Robotics and Telecommunication*, pp: 98-102.
- Sharma, R. and S. Koshy, 2011. Open source software in education. *Int. J. Manag. Bus. Stud.*, 1(2): 25-27.
- The Apache Software Foundation: Licenses, 2014. Retrieved from: <http://www.apache.org/licenses/> (Accessed on: March, 2014).
- The FreeBSD Copyright, 2014. Retrieved from: <http://www.freebsd.org/copyright/freebsd-license.html> (Accessed on: March, 2014).
- Wiley, D., 2006. Open source, openness and higher education. *Innov. J. Online Educ.*, 3(1): 1-6.
- Zaritski, R.M., 2003. Using open source software for scientific simulations, data visualization and publishing. *J. Comput. Sci. Coll.*, 19(2): 218-222.