International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Study of Open Source and Open Source Software

Kanchan Wankhade
I.T.& Computer Department, DPCOE, Pune University, Wagholi, Pune, Maharashtra, India.
dangekanchan05@gmail.com

Rupali Zamare
I.T.& Computer Department, DPCOE, Pune University, Wagholi, Pune, Maharashtra, India.
rupalizamare@yahoo.com

Piyusha Deshpande
I.T.& Computer Department, DPCOE, Pune University,Wagholi, Pune, Maharashtra, India.
d_piyusha@yahoo.com

Priyanka Kharote
I.T.& Computer Department, DPCOE, Pune University, Wagholi, Pune, Maharashtra, India.
priyanka.k59@gmail.com

Reema Bharati
I.T.& Computer Department, DPCOE, Pune University, Wagholi, Pune, Maharashtra, India.
reemabharati77@gmail.com

***Abstract* - This paper gives the detailed information about Open Source and Open Source Software, and difference between Proprietary and open source software. Many research groups are working towards the highly technological goal of modifying the open source software, which would dramatically improve software qualities. Open source software is very much useful in enhancing the capabilities of software engineers. Open source software is being widely used in government and private sectors as well as provides transparency to all which gives rise to democratic world. Also its a good earning source.**

***Keywords* - Attributes an open source software must have, Benefits and usage of open source software, Comparison between Proprietary software and Open source software, Examples of open source software, Future scope, History, Introduction, Licenses and schemes, Misconceptions about open source, Open Source Software, Open Source software-No one works for free , Responsibilities of open source software, Total cost of ownership.**

I. INTRODUCTION

Open source stands for software components and solutions whose source code is available, can be used, changed and distributed to other users following commonly agreed-upon rules. These rules are defined in various license models.[13]

Open source software is a part of the software ecosystem that affords software developers and users an alternative style of software development and distribution. It co-exists in that environment along with a broad spectrum of other development and distribution methods, including public domain software, freeware, shareware, proprietary commercial software and even vaporware. [3]

Open source first evolved during the 1970s. [6].Today we know about more than 140,000 open source projects. Even if these projects are developed using open source as a central method, they differ in their state of development and their maturity.

The (r)evolutionary elements of open source are:

1. Low cost and infinitely scalable distribution of the software through the Internet
2. Collaborative development process across time zones and geographies, often by only loosely connected development teams
3. Transparent access to the source code, allowing other projects to use and change the code and by providing the basis for new services offerings such as support and maintenance independently of owning the intellectual property. Based on these three elements, open source is changing the way software is developed, acquired and used.[13]

II. HISTORY

The concept of Open Source first evolved in 1970's. Richard Stallman, an American software developer developed free version of UNIX operating system. He believed that ideas and thoughts should be shared among several people. Hence he started the concept of Open Source software where source-code of software is widely shared among people. The term 'Open Source' was invented by Eric. S. Raymond in 1998 to avoid the confusion with the term 'Free Software'.

The main motto behind Open Source Software is to solve technical problems and share the outcome with rest of the world. The primary missions of Open Source Software are:

1) Allowing computer users to use, study, copy, modify, and redistribute computer programs
2) Allowing the development and use of free software and free documentation
3) Spreading awareness of ethical and political issues of freedom in the use of software
4) Developing new software and gaining knowledge from it
5) Making such software which can highly eliminate the need to use proprietary software.

III. OPEN SOURCE SOFTWARE

Open source software is becoming the most interesting and the wide sector of the entire information technology world. Open-source software (software with open source code) has become very popular in the last few years and is advancing at a speed unknown outside the world of IT.

Open-source software has become an area of business – an alternative, and therefore a competitor, to proprietary software.

This interest has also spread to the world of politics. Because open-source software is now increasingly used for commercial purposes, because it is characterized by independence from software producers, because it is opposed to the creation of monopolies and because it is characterized by a 'free-of-charge principle', open-source software has had a great impact on the political agenda, nationally and internationally.[3]

Open Source Software is software for which the underlying programming code is available to the users so

that they may read it, make changes to it, and build new versions of the software incorporating their changes. [3]

There is one more thing on which we would like to add a focus-the major difference between commercial and open software, is the way in which they are developed, the licensing schemes used in their distribution, their relationship to standards, and the total cost of ownership.

IV. Attributes A Software Must Have

There is a huge debate on benefits of Proprietary software and open source software; still some analysts suggest that the comparison should be done point by point. Even if the software is proprietary or open source it should have following attributes:

- Reliability: we can take it to mean the absence of defects which cause incorrect operation, data loss or sudden failures, perhaps what many people would mean when they use the term `bug'.
- Quality: commonly defined as the number of errors in a fixed number of lines of code.[6]
- Security: how unaffected the software is from unauthorized actions (e.g. viruses)
- Flexibility: how easily the software can be modified to meet specific needs and run on different types of devices.[6]
- Project management: how the development process is in proper order.
- Open Standards: documents created with one type of software being readable in another. This avoids 'lock-in' to a particular document format.[6]
- Switching Costs: the cost of moving from one system to another.[6]
- Total Cost of Ownership (TCO): the full costs incurred over the lifetime of the software.[6]
- User-Friendliness: how easy the software is for usage.

Advocates of open source software claims that as large team of developers is used in development process, bugs and errors can be easily detected and corrected which in turn increases the reliability and security of software. As number of developers are used, hence the software is developed in modules. Module is small block of code having some functionality. This Modularity simplifies software design and can increase the reliability as well as flexibility of software.

VI. Total Cost of Ownership [3]

The term "open source" is generally similar with the term "free software", and many open source advocates argue that[3] the most important advantage of the open source is the ability to have the software without any of the license fees. However, as the Free Software Foundation says, "'Free software' is a matter of liberty, not price.[3] The open source software can be acquired, modified, used and redistributed without any charge. But, when the "cost" of software is considered, it is very essential to consider the *total cost* of acquiring possession and use – *i.e.*, the "total cost of ownership".

In case of commercial software, the total cost is determined by the market research and development cost of that software according to needs of market *i.e.*, end user. It may not be perfectly fitting the need of market but mostly it is developed to satisfy substantial part of those needs. However, open source software is often developed without considering needs of end users and is distributed in a form that has to be modified to meet their needs. To meet the needs of a particular user, the modification cause higher expense. In addition, commercial software is commonly maintained and supported by the developer, whereas open source software commonly is not.

When considering the cost of "free" software, it is important to have in mind such things as the cost of modifying, maintaining and supporting the software, the cost of required user and technical documentation, the cost of quality assurance testing, and the costs of customization, implementation, defect correction, ongoing development, and dealing with security issues. The cost of training should be considered at two levels – training of technical personnel, and training of end users – because the available labor pool will more likely be possessed of knowledge and skills developed through experience with commercial products and may lack the expertise needed to support and use "free" substitutes without additional, specialized training. [3]

V. Open Source: No One Works For Free

Many people think that how to earn money in the field of Open Source Softwares. How one can earn money by developing software and giving it for free? These thought are lagging the contribution of companies to Open Source Software far behind.

They can think in a different way. It is better understood by taking the example of normal "Bell Curve" distribution which represents R&D investment over a Time. As the knowledge

About technology increases and technology starts succeeding in market, R&D investment increases. When taken the integral of normal distribution, it gives us the total investment.

Strong and good companies develop an invest in new sustaining technologies. e.g Microsoft's invention in Operatin Systems for PC. It started with DOS and later on Windows Series and finally Windows NT. Strong companies can easily change from technology to technology over a sustained innovation path.

In development of new technologies many people invest called as 'Contributers'. Each contibuter invest in development of software to meet their specific needs. As the project groups gets wider, many people start contributuing. However,it might happen that some people may have common needs. Hence in this case the total investment remains the same but the investment cost gets distributed amonf those people. Hence by this way individual contributer gets an enormous return on their investmnet. When organisational contributers invest then

the return on investment is huge amount as compare to their contributions.

We can take example of Linux Operating Systems. It has been adopted by many people as their desktop operating system and even by mobile handset manufacturers. They has been a continous change in Linux as each contributer contributes to it over the time passes. As stastics shows there is a new release of Linux after every three months. Hence in Open Source, nobody is working for free. The value gained by each contributer in enormous as compare to their investment. People choose working with Open source software for the following reasons: Firstly, open source programmers may improve their performance rather than reducing it in paid work. Second, the programmer may find their pleasure in choosing open source than a routine task set by an employer. Third, in a broad way, open source contributions may lead to future job offers, shares in commercial open source-based companies and last but not the least ego gratification from peer recognition.[17]

VI. Everything About Open Source Licensing

Models

Proprietary Model	Open Source Model
Licensor distributes object code only; source code is kept a trade secret.	Licensor distributes source code.
Modifications are prohibited	Modifications are permitted
All upgrades, support and development are done by licensor	Licensee may do its own development and support or hire any third party to do it
Fees are for the software license, maintenance, and upgrades	Fees, if any, are for integration, packaging, support, and consulting
Sublicensing is prohibited, or is a very limited right	Sublicensing is prohibited, or is a very limited right

[9]

VII. Schemes

There are over 40 different open source licenses, each with their own conditions and implications. Roughly they can be classified into these three categories:

- *Free-for-all licenses:*

These licenses only require licenses to give credit to the original authors. Derivative works can be kept proprietary. Sometimes these licenses are referred to as "academic licenses". Examples:

1. Berkeley Software Distribution (BSD): This permits a license to 'close' a version (by holding the most recent modifications to the source-code) and sell it as a proprietary product.
2. MIT: It licenses as well as the license used for the Apache Web server.

- *Keep-open licenses*:

Any modifications to software under these licenses have to be made available as open source as well. Larger works incorporating such software can be kept proprietary. Examples:

1. General Public License ("GPL"):

License Rights Granted under the GPL-

- Licensee may run the Program
- Licensee may copy and distribute verbatim copies of the Program's source code
- Licensee may create "derivative works" of the Program
- Licensee may distribute such derivative works

2. Lesser General Public License ("LGPL"):

Very similar to the GPL-

- Intent is to promote use of certain Libraries in conjunction with "non-free" programs.
- Contains exception for linking "works that use the library" to proprietary programs, which mitigates some Copy left concerns.

The licensee can either charge a fee for this service or work free of charge. It is the license which permit modification and distribution of free derivatives, but which preclude the creation of proprietary derivatives. The GNU Lesser GPL (used for Linux system libraries).

3. Mozilla Public License (used for the Firefox Web browser) is Keep-open licenses.

• *Share-alike licenses:*

When software under such a license is modified or extended, the result as a whole has to be made available as open source. The term 'copy left' is sometimes used to characterize this kind of license. The most famous example is the GNU GPL (General Public License), which applies e.g. to the Linux operating system. Another example is the Open Software License (OSL). [1,3,6,9]

VIII. Responsibilities Of Open Source Software

Keeping your software updated. Open source platforms frequently release new versions, and keeping your website in the latest version is strongly encouraged to keep your site secure and take advantage of new features.

Finding a support partner.

It is important to realize that open source software is not accompanied by traditional, "800-number" support; rather, it's your responsibility to find a trusted support partner. For example, Acquia offers a range of support packages for the Drupal CMS.

Contributing back to the community.

When you use open source software, and benefit from the work others have done, we feel that you have a responsibility to contribute solutions back to that community. Contributing custom modules and themes confers an added benefit: the community will ensure that these custom pieces stay updated as the platform evolves.[15]

IX. Comparison Between Proprietary and Open Source Software

Companies develop both types of software. Open Source Software is developed by an on-going basis where people come together and share their ideas. Hence people come together, share their ideas, contribute to the development of project, fix the errors and make a project so that others can use it freely.

Both Open Source Software and proprietary software allows companies to make profit. In case of Proprietary Software, companies gain money by selling license to use the software whereas in case of Open Source Software companies gain money by providing services such as installing software, customizing software and redistributing it under the terms of license available for Open Source Software

Commercial software is distributed only in binary i.e.,executable form and the source code remains with developer keeping the rights to customize the software with developer only. This is not the case with Open Source Software. In Open Source Software, the source code is easily available to everyone. These people have authorization to modify and redistribute the software in both binary and source code form.

X. Benefits/Use Of Open Source Software

1. Private sector:

The use of Open Source software is widely increasing in private sectors. Major corporations such as IBM believe that, because of open source, it has been possible for them to make use of a worldwide community of developers to improve their products and services. Some industry people say that Open Source will lead to a more competitive software industry. Currently over 67% of web-servers run open source software called Apache6. The majority of websites and email systems run on Open Source Software. Worldwide, around 30% of infrastructural computers run GNU/Linux, an open source operating system. However, use of Open Source Software on the desktop is limited: over 96% of desktop computers still use Microsoft Windows. Open Source Software has inspired new portable device projects, such as the 'Simputer'. This is a small, inexpensive, handheld computer, intended to bring computing power to India and other emerging economies.[6]

2. Government sector:

Governments' interest in Open Source Software has been increasing day by day. The UK Office of Government Commerce released a series of case studies in October 2004 which tells how open source software has been used in the public sector. However, the response of UK parliamentary onto the questions on use of open source software in government shows that its usage is still limited. The Office of the Deputy Prime Minister is funding the 'Open Source Academy' project. This is done to overcome the barriers on the usage of open source software in local government such as lack of information, skills, confidence and lack of suitable products.[6]

Examples of usage of open source in government:

• *Powys County Council, Wales:* by replacing existing machines with GNU/Linux servers, the number of servers has been dramatically reduced. This has led to cost savings on hardware, licensing and support.

• *Ministry of Defense (MoD) Academy:* OSS was chosen on the basis of functionality rather than to reduce costs. However, its use has led to lower licensing costs, lower consultancy rates for developers and faster development times. The software used was security accredited by the MoD.

• *Beaumont Hospital, Dublin, Ireland:* the hospital has projected savings of €8 million as a result of using Open Source Software. These were mainly due to an elimination of software licensing costs for an x-ray system and the ability to reuse hardware using GNU/Linux.[6]

3. Open Source and Transparency

Open Source also provides transparency as the ability to see 'source code' is provided by it. Because of this transparency open source has laid to democratic cultute. Examples are:

• *Tax and benefits*: under the Open Government Code and the Freedom of Information Act, the general public may have the right to know how a particular tax or benefit has been calculated. Open source may help achieve this, as having access to the source-code allows calculations to be read and checked;

• *E-voting:* with the transition to e-voting, political parties or the public might wish to inspect any software used in the process to counter electoral fraud or vote-rigging. Some say that Open Source Software is one possible way of doing this because the source-code is freely available to anyone who wishes to inspect it.

• *Public participation in Parliament:* some innovative projects have been developed using Open Source Software. Examples include the websites, 'They Work For You', which presents Hansard debates and Written Answers and 'The Public Whip', which gives details of MP voting records. These sites search the contents of Hansard and present it in an easy to read format for the public.[6]

Why Some Companies Use Open Source Software:

• Cost savings
• Stability
• No forced upgrades
• Access and broad rights to source code
• Access to skilled community of developers
• Ability to define and expedite new development.[6]

Why Some Companies Don't Use Open Source Software:

• "Free like a puppy"
• Migration costs
• Uncertainty about open development model
• Lack of IP warranties and indemnities
• Lack of performance warranties
• Copy left provisions of GPL, LGPL and similar Licenses[6]

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

XI. Common Misconceptions Regarding Open Source

There are a lot of myths out there regarding the use of open source technologies. Here are some common misconceptions we've encountered:

Open source isn't reliable. One common misconception surrounding open source software is that it is unreliable because it is developed by "amateurs." This could not be further from the truth. Open source technologies are often built by incredibly talented developers, and are used by some of the largest companies in the world.

There is no money to be made from open source software. Some people assume that because a product is built using open source technology, that product can not be sold for a profit. This is incorrect. The GNU General Public License, which governs many open source solutions, explicitly states that open source products can be sold for a profit.

Open source isn't secure. This is another common myth. Because the source code is freely

available, some people jump to the conclusion that it cannot be secure, because any would-be hacker can examine it. This is also false. Often, open source software is more secure, because its community of developers finds the security holes before a hacker does.

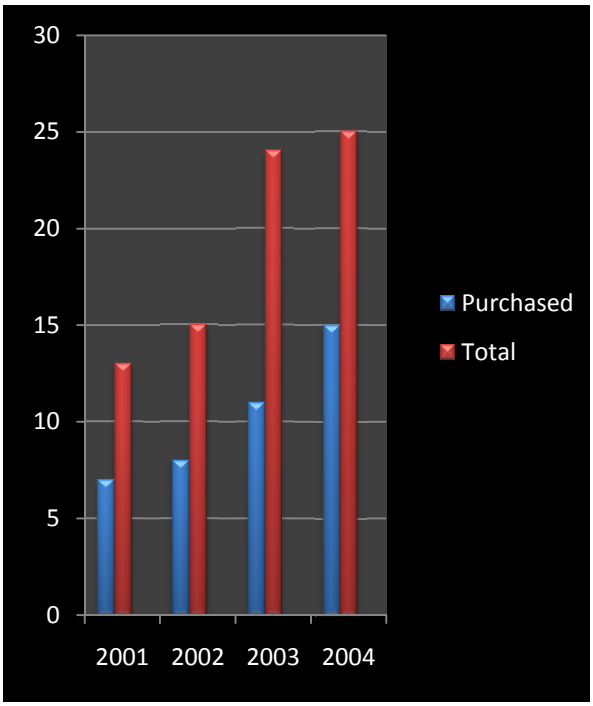
People will take my work and benefit from it. This one isn't so much a myth as it may be the wrong state of mind. In the open source world, contributing back to the community is encouraged; in fact, the success of the platform relies upon it. It can be strange to newcomers, but the world of open source is based on collaboration and mutual success. Members of this community recognize that it is great ideas that foster success, not lines of code.[15]

XII. Examples of Open Source Software

We have number of open source software available now a day. Today we know about more than 140,000 open source projects.

1. *Linux* (operating system kernel – substitutes for proprietary UNIX)
2. *Apache Web Server* (web server for UNIX systems)
3. *MySQL* (Structured Query Language – competes with Oracle)
4. *Cloudscape, Eclipse* (IBM contributions)
5. *OpenOffice* (open source implementation of Sun's StarOffice)
6. *Tomcat* (a web server)
7. *NetBeans* interactive development environment
8. *GNU RCS* (revision control system)
9. *Atlas* (Automatically Tuned Linear Algebra Software)
10. *CppAd c++* (automatic differentiation library)
11. *GNU* (Operating System)
12. *Firefox* (a Web browser)
13. *Android* (operating system)

Estimated No Of Copies Of Linux In Use(Millions) [9]



XIII. Open Source Learning: Lens On The Future

A few institutions seem poised to build on the successes in integrating technology through the open-source courseware and knowledge ware movement. These innovative institutions understand that the next big step involves a strategic convergence on three fronts :

(1) Technologies;

(2) Ideas and the cultural values that underpin them; and

(3) Resources.

Yet even though past successes have been built on the continuous development of physical infrastructure, information technology infrastructure, and social infrastructure, today's accomplishments reflect a landscape in which these three domains are converging, at the same time providing glimpses of how modern disruptive technologies might change the standard.

Only by strategically involving all-students, faculty, and staff- in this blending process will higher education develop meaningful descriptions of a learning infrastructure that is seamless, linking together people, resources, and discovery for a modern age. Successful instructional technology initiatives-forays into the new learning spaces that are at the heart of an institutions knowledge-dissemination mission-depend on such convergence in the face of disruption. They depend on an open-source environment, which will enlarge the aperture of any lens focused on the future of higher education,

shedding light on highly collaborative processes designed for creating excellent tools and materials that are reliable, affordable, and accessible for Information Age activities.[4]

Why should open source software be used in schools?

Students **should, at least**, be given the opportunity to see how their new tools work. They **should** be given the opportunity to examine the inner workings of software. They **should** be given the opportunity to extend the functions of their tools, where they see or imagine possibilities. They **should not** be held back by locking the toolbox of the Information Age and told they **must not** peer inside, **must not** try to discover how it works, **must not** share their tools with others, **must not** use their tools without paying proper tribute to the software overlords, under penalty and punishment of law.[21]

CONCLUSION

We have done the case study open source and open source software and come to a conclusion that we can increase efficiency of software's. Government commerce report has stated that open source software is a great alternative to proprietary software which costs much larger than open source software.

Most of desktop users can use open source software on regular basis. Open source software and proprietary software's are good in their own ways but the combinations of both can be a more creative and innovative approach.

REFERENCES

[1] Arnoud Engelfriet www.idc.com/nordic/downloads/.../9%20-Arnoud%20Engelfriet.pdf
[2] Open source_paper_english and Open sources in e-govt www.tekno.dk/pdf/projekter/p03_opensource_paper_english.pdf
[3] Open source software primer www.wipo.int/sme/en/.../pdf/opensource_software_primer.pdf
[4] Lens on future http://net.educause.edu/ir/library/pdf/erm0253.pdf
[5] Paper http://eu.conecta.it/paper.pdf
[6] Pdf2 www.parliament.uk/documents/post/postpn242.pdf
[7] Pdf3 mitpress.mit.edu/books/chapters/0262562278.pdf
[8] Two-sided www.stern.nyu.edu/.../Economides_Katsamakas_Two-sided.pdf
[9] What should you know about open source www.lw.com/upload/pubContent/_pdf/pub1062_1.pdf
[10] EQM0824 net.educause.edu/ir/library/pdf/EQM0824.pdf
[11] openSource depts.washington.edu/rfpk/pdfs/openSource.pdf
[12] Open source in developing countries www.itinwent.org/.../opensource_in_developing_countries_eng.pdf
[13] Optaros-catalog-2007 lwn.net/images/pdf/optaros-catalog-2007.pdf
[14] OSLicense www.people.hbs.edu/jlerner/OSLicense.pdf
[15] Using open source or_your_website_1 www.commonplaces.com/sites/.../using_open_source_fo... - United States
[16] Paper: http://www.bls.gov/ore/pdf/ec070120.pdf
[17] lerner- tirole http://people.ischool.berkeley.edu/~hal/Courses/StratTech09/Lectures/Platforms/lerner-tirole05.pdf
[18] Thesis http://www.karinvandenberg.nl/Thesis.pdf
[19] comparison between open source & propietary s/w: http://mansci.journal.informs.org/content/52/7/1057.abstract?ijkey=064040fe7e9670e308de159732b306c08a7f523d&keytype2=tf_ipsecsha
[20] benefits: http://open-source.gbdirect.co.uk/migration/benefit.html
[21] Why should open source s/w be used in schools? http://edge-op.org/grouch/schools.html
[22] The rising impact of open source: http://news.cnet.com/The-rising-impact-of-open-source/2010-1071_3-281244.html

AUTHOR'S PROFILE

Mrs. Kanchan V. Wankhade
She is presently Lecturer in Information Technology Department, in Dhole Patil College of Engineering Wagholi, Pune. She has completed Bachelors Degree in Information Technology and PGD-C-DAC.
E-mail id- dangekanchan05@gmail.com

Rupali Zamare
She is presently Lecturer in Information Technology Department, in Dhole Patil College of Engineering Wagholi, Pune. She has completed Bachelors Degree in Information Technology and MBA in systems. Currently she is appearing for M.E. in Computer.
E-mail id- rupalizamare@yahoo.com

Piyusha Deshpande
She is currently pursuing her T.E.(I.T.) from Dhole Patil Collegge of Engg., Wagholi, Pune.
E-mail id- d_piyusha@yahoo.com

Priyanka Kharote
She is currently pursuing her T.E.(Computer) from Dhole Patil Collegge of Engg., Wagholi, Pune.
E-mail id - priyanka.k59@gmail.com

Reema Bharati
She is currently pursuing her T.E. (I.T.) from Dhole Patil Collegge of Engg., Wagholi, Pune.
E-mail id - reemabharati77@gmail.com