Open Source Software and Licenses

Introduction

- Open source describes a broad general type of software license that makes source code available to the general public with relaxed or non-existent copyright restrictions.
- Software that is collectively developed by a community of technologists with an interest in a particular application or tool and then distributed at no cost to the broader community of individuals who can find a use for it.

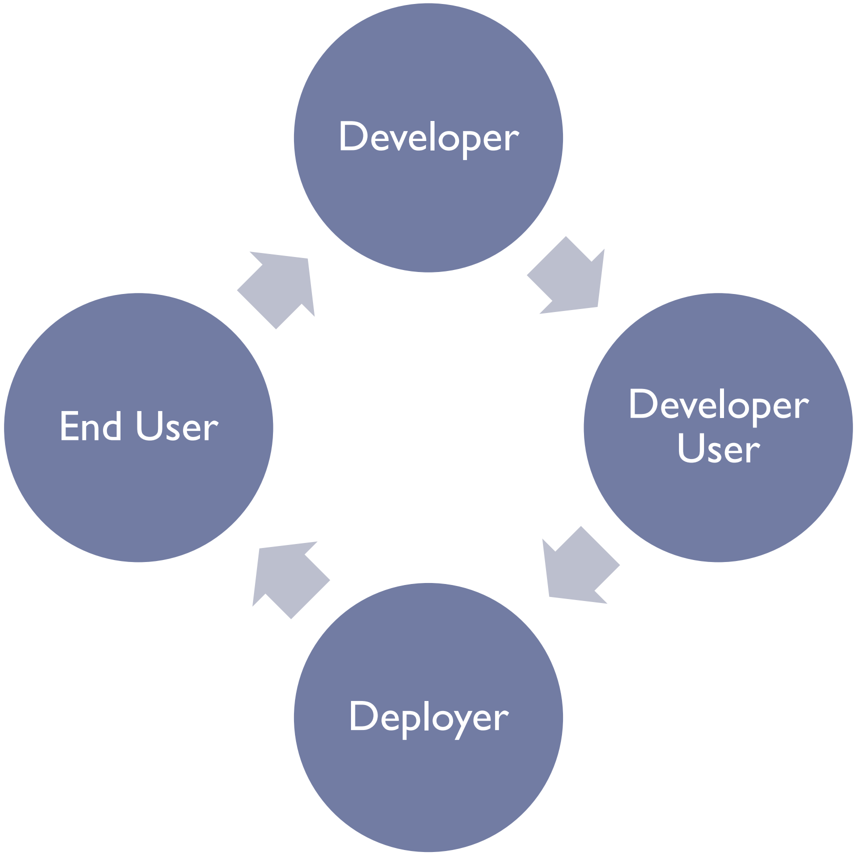
Open Source Definition

- Open source licenses meets the requirements of open source definition(OSD).
- Open source software are available in public domain and are even developed by collaborative efforts of public.
- Open Source is a certification standard issued by the Open Source Initiative (OSI) that indicates that the source code of a computer program is made available free of charge to the general public.

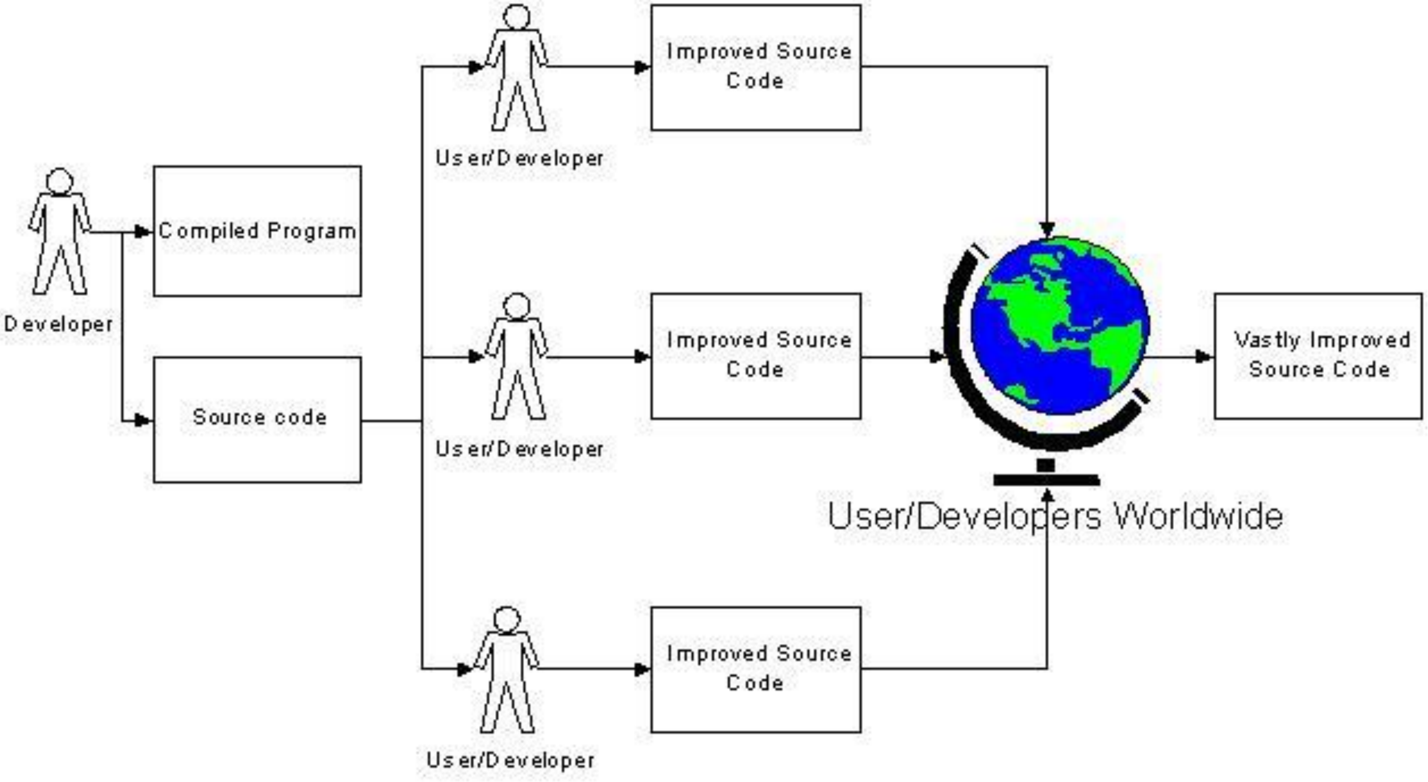
Open source is a Development Methodology

- Programmer commitment, because the programmers work on the software they need
- Rapid change, because programmers want to see results
- Unconstrained specifications, because there is no external client
- Collective responsibility for the code
- Response to change, dictated by (perhaps unexpected) use

Open source is about community



Development of Open Source



Development Philosophy

- According to "The Cathedral and Bazaar Model" the Open Source Software are being developed which has following pattern:
- *Users should be treated as co-developers*
  - Users should have access to source code
  - Users are encouraged to submit additions to s/w, code fixes for the s/w, report bugs and documentation.
  - More no. of co-developers help in building s/w faster.
  - Linus's law says :" *Given enough eyeballs all bugs are shallow.*" – Many users view same source code then they find all bugs and ways to solve them.
    - □ As some users have advanced programming knowledge.

Development Philosophy

- Early Releases
  - First version of software should be released as soon as possible to find co-developers earlier.
- Frequent Integration
  - Code changes should be integrated as frequently as possible to avoid integration at the end of the project life-cycle.
- Several versions
  - There should be at least 2 versions of software:-
    - 1.Development version with more no. of features.
    - 2. Stable version with fewer no. of features.
- Dynamic Decision-Making Structure
  - There is a need for a decision making structure, whether formal or informal, that makes strategic decisions depending on changing user requirements and other factors.

Distribution terms of Open Source Software (OSS) :

- Free Redistribution.
- Source Code.
- Derived Work.
- Integrity of the author’s source code.
- No discrimination against persons or groups.
- No discrimination against fields of endeavor.
- Distribution of licenses.
- License must not be specific to a product.
- License must not restrict other software.
- License must be technology neutral.

Distribution terms of Open Source Software (OSS) :

- Free Redistribution :
  - License shall not restrict any party from selling or giving the software as a component of an aggregate software distribution containing programs from several different sources.
  - License shall not require royalty or other fee for such sale.
- Source Code:
  - Program must include source code and must allow distribution in source code as well as compiled form.
  - Where some form of a product is not distributed with the source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge.
- High modularization
  - The general structure of the software should be modular allowing for parallel development on independent components.

Distribution terms of Open Source Software (OSS) :

- Source Code :
  - The source code must be the preferred form in which the programmer would modify the program.
  - Code that has been made difficult to understand for humans is not allowed.
  - Intermediate forms such as the output of a preprocessor or translator are not allowed.
- Derived Work :
  - The license must allow modifications and derived works, and must allow them to be distributed under the same terms as license of original software.

Distribution terms of Open Source Software (OSS) :

- Integrity of the author’s code :
  - The license may restrict source-code from being distributed in modified form only if the license allows the distribution of “patch files” with the source code for the purpose of modification of program at build time.
  - The license must explicitly permit distribution of software built from modified source code.
  - The license may require derived works to carry different names and version number from original source code.

Distribution terms of Open Source Software (OSS) :

- No discrimination against persons or group :
  - The license must not discriminate against any person or group of persons.
- No discrimination against fields of enterprise :
  - The license must not restrict anyone from making use of the program in a specific field of enterprise.
  - For example, it may not restrict the program from being used in a business, or being used for a genetics research.

Distribution terms of Open Source Software (OSS) :

- Distribution of license:
  - The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- License must not be specific to a product:
  - The rights attached to the program must not depend on the program's being part of a particular software distribution.
  - If program is extracted from that distribution and used or distributed within the terms of the program's license all parties to whom the program is being redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

Distribution terms of Open Source Software (OSS) :

- License must not restrict other software:
  - The license must not place restrictions on other software that is distributed along with the licensed software.
  - Ex. the license must not insist that all other programs distributed on the same medium must be an open-source software.
- License must be technology-neutral :
  - No provision of the license may be predicated on any individual technology or style of interface.

Open source

- Open source refers to software that is published under licenses that defines how source code can be made available to everyone to inspect, change, download, and explore as they wish.
- Software for which:
  - the source code is available to the end-user
  - the source code can be modified by the end-user
  - there are no restrictions on redistribution or use
  - the licensing conditions are usually intended to *facilitate* continued re-use and wide availability of the software, in both commercial and non-commercial contexts
  - the cost of acquisition to the end-user is often minimal.

Open Source vs. Closed Source

- **Open source software** can be defined as software distributed under a licensing agreement which allows the source code (computer code) to be shared, viewed and modified by other users and organizations.
- **Closed source software** can be defined as proprietary software distributed under a licensing agreement to authorized users with private modification, copying and republishing restrictions.

Open Source vs. Closed Source

Term	Open Source	Closed Source
Cost	•Open source Software may be free. If you have in-house capability for implementation, training then it is the best option. Otherwise you need to pay for it. •Open source software requires a certain level of technical expertise in order to manage content. •Open source software providers are increasingly charging for add-ons, additional services and integration.	•The cost of proprietary software will vary from a few thousand to a few hundred thousand dollars, depending on the complexity of the system required. •This cost is made up of a **base fee for software**, **integration** and **services** and **annual licensing/support fees**.

Open Source vs. Closed Source

Term	Open Source	Closed Source
Services & Support	•Service is one of the key issues regarding open source software. Open source software relies on its online community network to deliver support via forums and blogs. •These communities cannot guarantee the high level of responsive service and support proprietary software can offer.	•Proprietary software providers offer ongoing support to users, a key selling point for users without technical expertise. •Service is one of the main reasons users choose proprietary software, many proprietary software providers compete on service , increasing the bargaining power of buyers and thereby increasing customer service levels among providers.

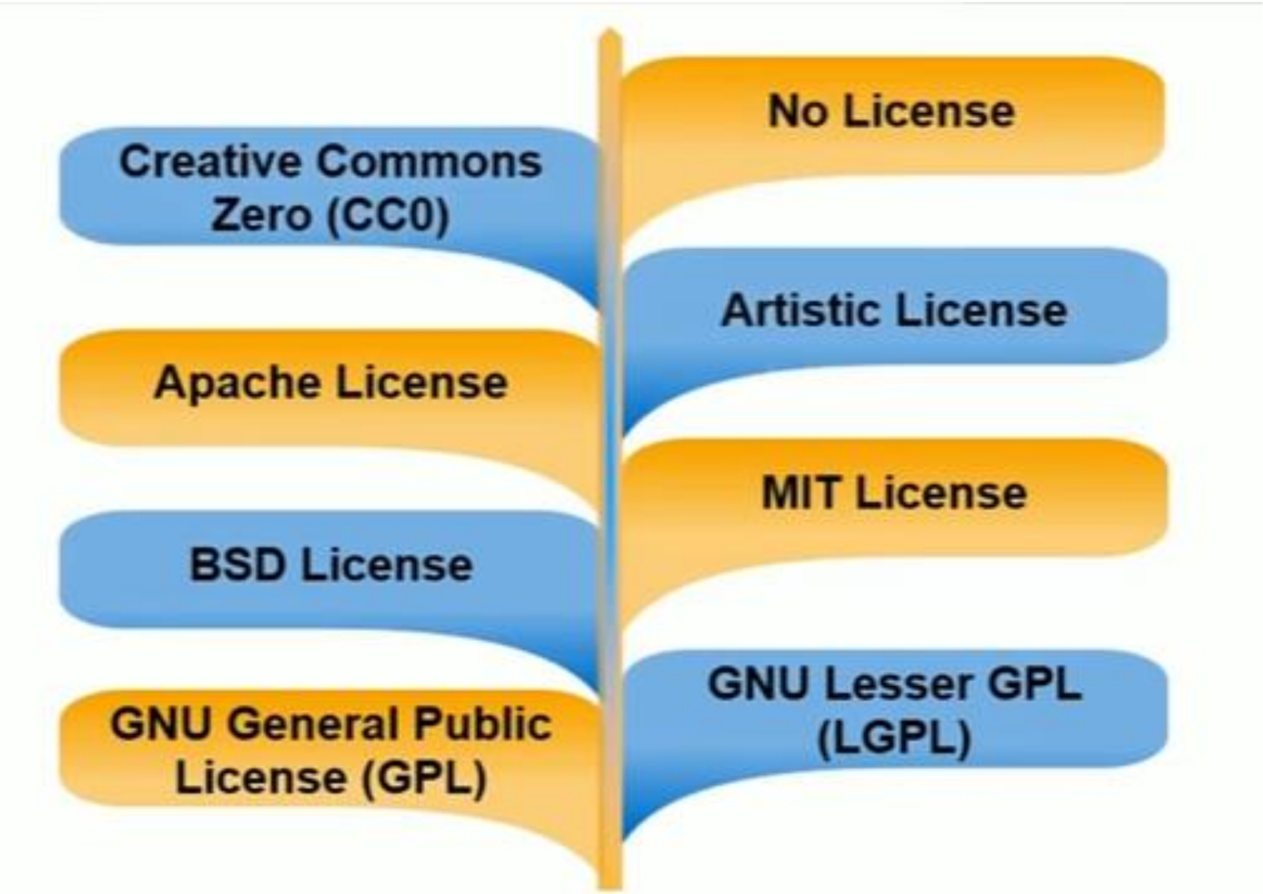
Open Source vs. Closed Source

	Open Source Term	Closed Source
Innovation	Open source software enables innovation by providing users with the freedom and flexibility to adapt the software to suit, without restriction.	Proprietary software providers do not allow users to view or alter the source code. While this may be viewed as a disadvantage to some, it ensures the security and reliability of the software.
Usability	It is developer-centric. Requires system administration expertise. No availability of the proper documentation of the system is available.	Proprietary software generally employs expert usability testing, and as the software is normally aimed at a more targeted audience

Open Source vs. Closed Source

Term	Open Source	Closed Source
Security	•Open source software is not developed in a controlled environment. •While development they often have a concentrated dev. team, individuals all over the world who may not work on the software for the duration of its developing lifetime. This lack of continuity and common direction can lead to barriers to effective communication surrounding the software.	•Proprietary software is viewed as more secure because it is developed in a controlled environment by a concentrated team with a common direction.

Few Widely used Open Source Licenses



No License

Creative Commons Zero (CC0)

Artistic License

Apache License

MIT License

BSD License

GNU Lesser GPL (LGPL)

GNU General Public License (GPL)

Open Source Licenses Organizations

Open Source Initiative (OSI)

Free Software Foundation (FSF)

Open Source Initiative

GNU General Public License (GPL)

- Grants wide range of rights to developers
  - Copy the software
  - Distribute the software in your preferred way
  - Charge a fee to distribute a software
  - Make whatever modifications to the software we want
  - Copyleft (retains with modifications)

Permissions	Conditions	Limitations
Commercial Use	Disclose Source	Liability
Distribution	License and Copyright Notice	Warranty
Modification	Same License	Sub License
Patent Use	State Changes	
Private Use		

GNU Library or "Lesser" General Public License (LGPL)

- Used for Software Libraries
- Requires derived work licensed under the same license
- Works that only links to it do not fall under this restriction
- Permits sublicesing

Permissions	Conditions	Limitations
Commercial Use	Disclose Source	Liability
Distribution	License and Copyright Notice	Warranty
Modification	Same License (library)	
Patent Use	State Changes	
Private Use		

BSD

- Permissive License
- Can be modified and given to a proprietary firm
- It is GPL compatible license compatible with OSI and FSF
- BSD “New” ,“Revised” ,“Simplified” and “Free” BSD license

Permissions	Conditions	Limitations
Commercial Use	License and Copyright Notice	Liability
Distribution		Warranty
Modification		
Private Use		

MIT License

- More Permissive and Short License

Permissions	Conditions	Limitations
Commercial Use	License and Copyright Notice	Liability
Distribution		Warranty
Modification		
Private Use		
Sub Licensing		

Apache license

- Used by apache software
- Version 2 grants number of rights to the user
- Permissive license
- Applies to both copyrights and patents

Artistic License

- Usually used by Programmers
- Heavily favored by Perl Community
- Requires that modified version of the software do not prevent running the standard version
- Approved by OSI, but not approved by FSF

Permissions	Conditions	Limitations
Commercial Use	License and Copyright Notice	Liability
Distribution	State Changes	Warranty
Modification		Trademark Use
Patent Use		
Private Use		

Creative Commons Zero (CC0)

- Public domain dedication
- Waives copyright interest in any work that is created
- Widest reach

Permissions	Conditions	Limitations
Commercial Use		Trademarks
Distribution		Warranty
Modification		Liability
Private Use		Patent use

Mozilla License

- Maintained by Mozilla foundation
- Attempts to comprise between permissive BSD and license and reciprocal GPL license

Permissions	Conditions	Limitations
Commercial Use	Disclosure of Notice	Trademarks
Distribution	Copyrights and License Agreement	Warranty
Modification		
Private Use		
Sub licensing		

Software and Licenses

Open Source Software	License
jQuery, Rails	MIT License
Apache Tomcat, SVN, NuGet	Apache License
Linux, GIT, WordPress, QGIS	GNU GPL

Open Source Software

- LibreOffice
- LabView
- LibreCAD
- SciLab
- LaTex
- CEDAR
- Logisim
- Logic Gate Simulator
- VLC Media Player
- Firefox

Free software

- Free software is the matter of user's freedom to run, copy, distribute, study, change & improve the software.
- It provides the program's users the four essential freedoms:-
  - Freedom to run the program for any purpose.
  - Freedom to study how the program works and change it so it does your computing as you wish.
  - Freedom to redistribute copies.
  - Freedom to distribute copies of your modified versions to others.

Open Source vs. Free Source

- Open source means that the source code is available to all potential users, and they are free to use, modify, and re-distribute the source code.
  - Legally, the "free" of open source refers exclusively to the source code, and it is possible to have support, services, documentation, and even binary versions which are not monetarily free.
- Free Source means software which can be downloaded, used, and copied without restrictions.
- Main difference between OSS and FSS is that there is no community and no development infrastructure around "free software" as there is around open source software.

Open Source vs. Source-available

- Source available is software where the source is available for viewing, but which may not legally be modified or redistributed.
- Such software is more often referred to as *source-available*, or as shared source.

Pros of OSS

- Open source software have several advantages and disadvantages.
  - The main advantage for business is that open source is a good way for business to achieve greater penetration of the market.
  - Companies that offer open source software are able to establish an industry standard and, thus, gain competitive advantage.
  - Less costs of marketing and logistical services are needed for OSS.
  - It also helps companies to keep up to date of all technology developments.
  - It is a good tool to promote a company's image, including its commercial products

Pros of OSS

- Helped to produce reliable, high quality software quickly and inexpensively.
- Offers the potential for a more flexible technology and quicker innovation.
- It is said to be more reliable since it typically has thousands of independent programmers testing and fixing bugs of the software.
- It is flexible because modular systems allow programmers to build custom interfaces, or add new abilities to it and it is innovative since open source programs are the product of collaboration among a large number of different programmers.
- The mix of divergent perspectives, corporate objectives, and personal goals speeds up innovation.
- Free software can be developed in accord with purely technical requirements. It does not require to think about commercial pressure that often degrades the quality of the software.
- Commercial pressures make traditional software developers pay more attention to customers' requirements than to security requirements, since such features are somewhat invisible to the customer.

Cons of OSS

- It is sometimes said that the open source development process may not be well defined and the stages in the development process, such as system testing and documentation may be ignored.
- However this is only true for small (mostly single programmer) projects.
- Larger, successful projects do define and enforce at least some rules as they need them to make the teamwork possible.
- In the most complex projects these rules may be as strict as reviewing even minor change by two independent developers.
- Not all OSS initiatives have been successful, for example, SourceXchange

Cons of OSS

- Poor support
- Software experts and researchers who are not convinced by open source's ability to produce quality systems identify the unclear process, the late defect discovery and the lack of any empirical evidence as the most important problems.
- It is also difficult to design a commercially sound business model around the open source paradigm.
- Consequently, only technical requirements may be satisfied and not the ones of the market.
- In terms of security, open source may allow hackers to know about the weaknesses or loopholes of the software more easily than closed-source software.
- It is depended of control mechanisms in order to create effective performance of autonomous agents who participate in virtual organizations

Open source technology importance

- Initial cost.
- Open source software is generally available for free and it is not necessary to purchase additional licenses for every computer that the program is to be installed on or for every person who is going to use the software.
- Open source software not only has a lower acquisition cost than proprietary software, it often has lower implementation and support costs as well.
- Since open source software is typically freely available to download, librarians and systems administrators can install complete production-ready versions of software and evaluate competing packages.
- This can be done not only without any license fees, but also without having to stick to a vendor's trial period, evaluate a limited version of the software, or deal with the vendor's sales personnel.
- If the person or organization likes an overall open source package but would like a few added features they can add these features themselves.
- Even if a person or organization does not have in-house expertise they can benefit from source code availability because another library may be able to provide them the fix or they can hire a consultant to make the changes that they desire.

Open Source Software Technology Importance

- Open source software allows for different vendors to compete for support contracts based on quality of service and on price.
- Access to the source code also allows for self-support when practical and desired.
- The amount of vendor lock-in is dramatically reduced with open source software.
- The large initial costs often associated with proprietary software makes it difficult to reevaluate the choice of software when it does not live up to expectations.
- Proprietary software can lead to a single point of failure.
- If a vendor goes out of business or decides not to support a program anymore there is often nothing an user can do.
- Organizations using the software could provide self support or other vendors can come in and fill the void left by the previous vendor if the program were available as open source software.