**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=2)**] History**

|  |  |
| --- | --- |
| Question book-new.svg | This section **needs additional** [**citations**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources#Inline_citations) **for** [**verification**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability)**.**Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit) by adding [reliable references](http://en.wikipedia.org/wiki/Wikipedia%3AReliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3AFact) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(November 2007)* |

*Main article:* [*Open source movement*](http://en.wikipedia.org/wiki/Open_source_movement)

The [free software movement](http://en.wikipedia.org/wiki/Free_software_movement) was launched in 1983. In 1998, a group of individuals advocated that the term [free software](http://en.wikipedia.org/wiki/Free_software) should be replaced by open source software (OSS) as an expression which is less ambiguous and more comfortable for the corporate world[[5]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-4). Software developers may want to publish their software with an [open source license](http://en.wikipedia.org/wiki/Open_source_license), so that anybody may also develop the same software or understand its internal functioning. Open source software generally allows anyone to create modifications of the software, port it to new operating systems and processor architectures, share it with others or market it.

The [Open Source Definition](http://en.wikipedia.org/wiki/Open_Source_Definition), notably, presents an open source philosophy, and further defines the terms of usage, modification and redistribution of open source software. [Software licenses](http://en.wikipedia.org/wiki/Software_license) grant rights to users which would otherwise be reserved by [copyright](http://en.wikipedia.org/wiki/Copyright) law to the copyright holder. Several open source software licenses have qualified within the boundaries of the Open Source Definition. The most prominent and popular example is the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL). While open source distribution presents a way to make the source codes of a product publicly accessible, the open source licenses allow the authors to fine tune such access.

The *open source* label came out of a strategy session held in [Palo Alto](http://en.wikipedia.org/wiki/Palo_Alto%2C_California) in reaction to [Netscape's](http://en.wikipedia.org/wiki/Netscape_Communications_Corporation) January 1998 announcement of a source code release for [Navigator](http://en.wikipedia.org/wiki/Netscape_Navigator) (as [Mozilla](http://en.wikipedia.org/wiki/Mozilla)). A group of individuals at the session included Todd Anderson, [Larry Augustin](http://en.wikipedia.org/wiki/Larry_Augustin), John Hall, Sam Ockman, Christine Peterson and [Eric S. Raymond](http://en.wikipedia.org/wiki/Eric_S._Raymond). They used the opportunity before the release of Navigator's source code to clarify a potential confusion caused by the ambiguity of the word "free" in [English](http://en.wikipedia.org/wiki/English_language). The 'open source' movement is generally thought to have begun with this strategy session.[[*citation needed*](http://en.wikipedia.org/wiki/Wikipedia%3ACitation_needed)] Many people, nevertheless, claimed that the birth of the [Internet](http://en.wikipedia.org/wiki/Internet), since 1969, started the open source movement, while others do not distinguish between open source and free software movements.[[6]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-5)

The [Free Software Foundation](http://en.wikipedia.org/wiki/Free_Software_Foundation) (FSF), started in 1985, intended the word 'free' to mean *freedom to distribute* (or "free as in free speech") and not *freedom from cost* (or "free as in free beer"). Since a great deal of free software already was (and still is) free of charge, such free software became associated with zero cost, which seemed anti-commercial.

The [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative) (OSI) was formed in February 1998 by Eric S. Raymond and [Bruce Perens](http://en.wikipedia.org/wiki/Bruce_Perens). With at least 20 years of evidence from case histories of closed software development versus open development already provided by the Internet developer community, the OSI presented the 'open source' case to commercial businesses, like Netscape. The OSI hoped that the usage of the label "open source," a term suggested by Peterson of the [Foresight Institute](http://en.wikipedia.org/wiki/Foresight_Institute) at the strategy session, would eliminate ambiguity, particularly for individuals who perceive "free software" as anti-commercial. They sought to bring a higher profile to the practical benefits of freely available source code, and they wanted to bring major software businesses and other high-tech industries into open source. Perens attempted to register "open source" as a [service mark](http://en.wikipedia.org/wiki/Service_mark) for the OSI, but that attempt was impractical by [trademark](http://en.wikipedia.org/wiki/Trademark) standards. Meanwhile, due to the presentation of Raymond's paper to the upper management at Netscape—Raymond only discovered when he read the [Press Release](http://wp.netscape.com/newsref/pr/newsrelease558.html), and was called by Netscape CEO Jim Barksdale's PA later in the day—Netscape released its Navigator source code as open source, with favorable results.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=3)**] Definitions**

*Main article:* [*Open Source Definition*](http://en.wikipedia.org/wiki/Open_Source_Definition)

There are numerous groups who claim ownership of the term "Open Source", but in reality the term has not been trademarked. The [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative)'s definition is widely recognized as the standard or [*de facto*](http://en.wikipedia.org/wiki/De_facto) definition.[*[citation needed](http://en.wikipedia.org/wiki/Wikipedia%3ACitation_needed%22%20%5Co%20%22Wikipedia%3ACitation%20needed)*]

The Open Source Initiative (OSI) formed in February 1998 by Raymond and Perens. With about 20 years of evidence from case histories of closed and open development already provided by the Internet, the OSI continued to present the 'open source' case to commercial businesses. They sought to bring a higher profile to the practical benefits of freely available source code, and wanted to bring major software businesses and other high-tech industries into open source. Perens adapted [Debian](http://en.wikipedia.org/wiki/Debian)'s [Free Software](http://en.wikipedia.org/wiki/Free_Software) Guidelines to make the [The Open Source Definition](http://en.wikipedia.org/wiki/Open_Source#The_Open_Source_Definition).[[7]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-6)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=4)**] The Open Source Definition**

The [Open Source Definition](http://en.wikipedia.org/wiki/Open_Source_Definition) is used by the [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative) to determine whether a [software](http://en.wikipedia.org/wiki/Computer_software) license can be considered **open source**. The definition was based on the [Debian Free Software Guidelines](http://en.wikipedia.org/wiki/Debian_Free_Software_Guidelines), written and adapted primarily by [Bruce Perens](http://en.wikipedia.org/wiki/Bruce_Perens).[[8]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-7)[[9]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-8) Perens did not base his writing on the "four freedoms" of Free Software from the [Free Software Foundation](http://en.wikipedia.org/wiki/Free_Software_Foundation), which were only widely available later.[[10]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-9)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=5)**] Perens' principles**

Under Perens' definition, open source describes a broad general type of [software license](http://en.wikipedia.org/wiki/Software_license) that makes source code available to the general public with relaxed or non-existent [copyright](http://en.wikipedia.org/wiki/Copyright) restrictions. The principles, as stated, say absolutely nothing about [trademark](http://en.wikipedia.org/wiki/Trademark) or [patent](http://en.wikipedia.org/wiki/Patent) use and require absolutely no cooperation to ensure that any common [audit](http://en.wikipedia.org/wiki/Audit) or [release](http://en.wikipedia.org/wiki/Software_release) regime applies to any derived works. It is an explicit “feature” of open source that it may put no restrictions on the use or distribution by any organization or user. It forbids this, in principle, to guarantee continued access to derived works even by the major original contributors.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=6)**] Proliferation of the term**

While the term applied originally only to the source code of software,[[11]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-10) it is now being applied to many other areas such as open source ecology,[[12]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-11) a movement to decentralize technologies so that any human can use them. However, it is often misapplied to other areas which have different and competing principles, which overlap only partially.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=7)**] Non-software use**

The principles of open source have been adapted for many other forms of [user generated content](http://en.wikipedia.org/wiki/User_generated_content) and technology, including [open source hardware](http://en.wikipedia.org/wiki/Open_source_hardware).

Supporters of the [open content](http://en.wikipedia.org/wiki/Open_content) movement advocate some restrictions of use, requirements to share changes, and [attribution](http://en.wikipedia.org/wiki/Attribution_%28copyright%29) to other authors of the work.

This “culture” or [ideology](http://en.wikipedia.org/wiki/Ideology) takes the view that the principles apply more generally to facilitate concurrent input of different agendas, approaches and priorities, in contrast with more centralized models of development such as those typically used in commercial companies.[[13]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-autogenerated1-12)

Advocates of the open source principles often point to [Wikipedia](http://en.wikipedia.org/wiki/Wikipedia) as an example, but Wikipedia has in fact often restricted certain types of use or user, and the [GFDL](http://en.wikipedia.org/wiki/GFDL) license it has historically used makes specific requirements of all users, which technically violates the open source principles.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=8)**] Business models**

*Main article:* [*Business models for open source software*](http://en.wikipedia.org/wiki/Business_models_for_open_source_software)

There are a number of commonly recognized barriers to the adoption of open source software by enterprises. These barriers include the perception that open source licenses are [viral](http://en.wikipedia.org/wiki/Viral_license), lack of formal support and training, the velocity of change, and a lack of a long term roadmap. The majority of these barriers are risk-related. From the other side, not all proprietary projects disclose exact future plans, not all open source licenses are equally viral and many serious OSS projects (especially operating systems) actually make money from paid support and documentation.

A commonly employed [Business Strategy](http://en.wikipedia.org/wiki/Business_Strategy) of Commercial Open Source Software Firms is the [Dual-License](http://en.wikipedia.org/wiki/Dual-licensing) Strategy, as demonstrated by [Ingres](http://en.wikipedia.org/wiki/Ingres_%28database%29), [MySQL](http://en.wikipedia.org/wiki/MySQL), [Alfresco](http://en.wikipedia.org/wiki/Alfresco_%28software%29), and others.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=9)**] Widely used open source products**

Open source software (OSS) projects are built and maintained by a network of volunteer programmers. Prime examples of open source products are the [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), the e-commerce platform [osCommerce](http://en.wikipedia.org/wiki/OsCommerce) and the internet browser [Mozilla Firefox](http://en.wikipedia.org/wiki/Mozilla_Firefox). One of the most successful open source products is the [GNU/Linux](http://en.wikipedia.org/wiki/Linux) operating system, an open source [Unix-like](http://en.wikipedia.org/wiki/Unix-like) operating system.[[14]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-13)[[15]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-14)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=10)**] Development philosophy**

In his 1997 essay [The Cathedral and the Bazaar](http://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar)[[16]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-ray2000-15), [open source evangelist](http://en.wikipedia.org/wiki/Open_source_evangelist) [Eric S. Raymond](http://en.wikipedia.org/wiki/Eric_S._Raymond) suggests a model for developing OSS known as the *bazaar* model. Raymond likens the development of software by traditional methodologies to building a cathedral, "*carefully crafted by individual wizards or small bands of mages working in splendid isolation*".[[16]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-ray2000-15) He suggests that all software should be developed using the bazaar style, which he described as "*a great babbling bazaar of differing agendas and approaches*."

In the traditional model of development, which he called the *cathedral* model, development takes place in a centralized way. Roles are clearly defined. Roles include people dedicated to designing (the architects), people responsible for managing the project, and people responsible for implementation. Traditional software engineering follows the cathedral model. [Fred P. Brooks](http://en.wikipedia.org/wiki/Fred_Brooks) in his book [*The Mythical Man-Month*](http://en.wikipedia.org/wiki/The_Mythical_Man-Month) advocates this model. He goes further to say that in order to preserve the architectural integrity of a system, the system design should be done by as few architects as possible.

The bazaar model, however, is different. In this model, roles are not clearly defined. Gregorio Robles[[17]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-16) suggests that software developed using the bazaar model should exhibit the following patterns:

Users should be treated as co-developers

The users are treated like co-developers and so they should have access to the source code of the software. Furthermore users are encouraged to submit additions to the software, code fixes for the software, bug reports, documentation etc. Having more co-developers increases the rate at which the software evolves. [Linus's law](http://en.wikipedia.org/wiki/Linus%27s_law) states that, "*Given enough eyeballs all bugs are shallow.*" This means that if many users view the source code they will eventually find all bugs and suggest how to fix them. Note that some users have advanced programming skills, and furthermore, each user's machine provides an additional testing environment. This new testing environment offers that ability to find and fix a new bug.

Early releases

The first version of the software should be released as early as possible so as to increase one's chances of finding co-developers early.

Frequent integration

Code changes should be integrated (merged into a shared code base) as often as possible so as to avoid the overhead of fixing a large number of bugs at the end of the project life cycle. Some open source projects have nightly builds where [integration is done automatically](http://en.wikipedia.org/wiki/Continuous_integration) on a daily basis.

Several versions

There should be at least two versions of the software. There should be a buggier version with more features and a more stable version with fewer features. The buggy version (also called the development version) is for users who want the immediate use of the latest features, and are willing to accept the risk of using code that is not yet thoroughly tested. The users can then act as co-developers, reporting bugs and providing bug fixes.

High modularization

The general structure of the software should be modular allowing for parallel development on independent components.

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. Cf. [Extreme programming](http://en.wikipedia.org/wiki/Extreme_programming).

Most well known open-source software products follow the bazaar model as suggested by Eric Raymond. These include projects such as the [Linux kernel](http://en.wikipedia.org/wiki/Linux_kernel), [Firefox](http://en.wikipedia.org/wiki/Firefox), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), the [GNU Compiler Collection](http://en.wikipedia.org/wiki/GNU_Compiler_Collection), and [Perl](http://en.wikipedia.org/wiki/Perl).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=11)**] Licensing**

|  |  |
| --- | --- |
| Question book-new.svg | This section **does not** [**cite**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources) **any** [**references or sources**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability).Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit) by adding citations to [reliable sources](http://en.wikipedia.org/wiki/Wikipedia%3AReliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3ACitation_needed) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(November 2007)* |

*Main article:* [*Open source license*](http://en.wikipedia.org/wiki/Open_source_license)

A license defines the rights and obligations that a licensor grants to a licensee. Open Source licenses grant licensees the right to copy, modify and redistribute source code (or content). These licenses may also impose obligations (e.g., modifications to the code that are distributed must be made available in source code form, an author attribution must be placed in a program/ documentation using that Open Source, etc.).

Authors initially derive a right to grant a license to their work based on the legal theory that upon creation of a work the author owns the copyright in that work. What the author/licensor is granting when they grant a license to copy, modify and redistribute their work is the right to use the author’s copyrights. The author still retains ownership of those copyrights, the licensee simply is allowed to use those rights, as granted in the license, so long as they maintain the obligations of the license. The author does have the option to sell/assign, versus license, their exclusive right to the copyrights to their work; where upon the new owner/assignee controls the copyrights. The ownership of the copyright (the “rights”) is separate and distinct from the ownership of the work (the “thing”) - a person can own a copy of a piece of code (or a copy of a book) without the rights to copy, modify or redistribute copies of it.

When an author contributes code to an Open Source project (e.g., Apache.org) they do so under an explicit license (e.g., the Apache Contributor License Agreement) or an implicit license (e.g., the Open Source license under which the project is already licensing code). Some Open Source projects do not take contributed code under a license, but actually require (joint) assignment of the author’s copyright in order to accept code contributions into the project (e.g., OpenOffice.org and its Joint Copyright Assignment agreement).

Placing code (or content) in the public domain is a way of waiving an author’s (or owner’s) copyrights in that work. No license is granted, and none is needed, to copy, modify or redistribute a work in the public domain.

Examples of [free software license](http://en.wikipedia.org/wiki/Free_software_license) / open source licenses include [Apache License](http://en.wikipedia.org/wiki/Apache_License), [BSD license](http://en.wikipedia.org/wiki/BSD_license), [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), [GNU Lesser General Public License](http://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License), [MIT License](http://en.wikipedia.org/wiki/MIT_License), [Eclipse Public License](http://en.wikipedia.org/wiki/Eclipse_Public_License) and [Mozilla Public License](http://en.wikipedia.org/wiki/Mozilla_Public_License).

The proliferation of open source licenses is one of the few negative aspects of the open source movement because it is often difficult to understand the legal implications of the differences between licenses.

An important legal milestone for the open source / [free software movement](http://en.wikipedia.org/wiki/Free_software_movement) was passed in 2008, when the US federal appeals court ruled that [free software licences](http://en.wikipedia.org/wiki/Free_software_licences) definitely do set legally binding conditions on the use of copyrighted work, and they are therefore enforceable under existing copyright law. As a result, if end-users do violate the licensing conditions, their license disappears, meaning they are infringing copyright[[18]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-17).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=12)**] Funding**

*See also:* [*Commercial open source applications*](http://en.wikipedia.org/wiki/Commercial_open_source_applications)

Unlike proprietary off-the-shelf software, which comes with restrictive copyright licenses, open-source software can be given away for no charge. This means that its creators cannot require each user to pay a license fee to fund development. Instead, a number of alternative models for funding its development have emerged.

Software can be developed as a consulting project for one or more customers. The customers pay to direct the developers' efforts: to have bugs prioritized and fixed or features added. Companies or independent consultants can also charge for training, installation, [technical support](http://en.wikipedia.org/wiki/Technical_support), or customization of the software.

Another approach to funding is to provide the software freely, but sell licenses to proprietary add-ons such as data libraries. For instance, an open-source [CAD](http://en.wikipedia.org/wiki/Computer_aided_design) program may require parts libraries which are sold on a subscription or flat-fee basis. Open-source software can also promote the sale of specialized hardware that it interoperates with, as in the case of the Asterisk telephony software, developed by a manufacturer of PC telephony hardware.

Many open-source software projects have begun as research projects within universities, as personal projects of students or professors, or as tools to aid scientific research. The influence of universities and research institutions on open source shows in the number of projects named after their host institutions, such as [BSD Unix](http://en.wikipedia.org/wiki/BSD_Unix), [CMU Common Lisp](http://en.wikipedia.org/wiki/CMU_Common_Lisp), or the [NCSA HTTPd](http://en.wikipedia.org/wiki/NCSA_HTTPd) which evolved into [Apache](http://en.wikipedia.org/wiki/Apache_Web_server).

Companies may employ developers to work on open-source projects that are useful to the company's infrastructure: in this case, it is developed not as a product to be sold but as a sort of shared public utility. A larger project such as the Linux kernel may have contributors from dozens of companies which use and depend upon it, as well as hobbyist and research developers.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=13)**] Open source vs. closed source**

|  |  |
| --- | --- |
| Question book-new.svg | This section **does not** [**cite**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources) **any** [**references or sources**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability).Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit) by adding citations to [reliable sources](http://en.wikipedia.org/wiki/Wikipedia%3AReliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3ACitation_needed) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(November 2007)* |

*Main article:* [*Comparison of open source and closed source*](http://en.wikipedia.org/wiki/Comparison_of_open_source_and_closed_source)

The debate over *open source* vs. [*closed source*](http://en.wikipedia.org/wiki/Closed_source) (alternatively called [proprietary software](http://en.wikipedia.org/wiki/Proprietary_software)) is sometimes heated.

One source of conflict is related to economics: Making money through traditional methods, such as sale of the use of individual copies and patent royalty payment (generally called *licensing*), is more difficult and in many ways against the very concept of open source software.

Some closed-source advocates see open source software as damaging to the market of commercial software. This is one of the many reasons, as mentioned above, that the term *free software* was replaced with *open source* — because many company executives could not believe in a product that did not participate economically in a free-market or mixed-market economy.

The counter to this argument is the use of open source software to fuel the market for a separate product or service. For example:

* Providing support and installation services; similar to IT Security groups, Linux Distributions, and Systems companies.
* Using the software as a stepping stone to sell a higher-end product or service; e.g., [OpenOffice.org](http://en.wikipedia.org/wiki/OpenOffice.org) vs. [StarOffice](http://en.wikipedia.org/wiki/StarOffice).
* Cost avoidance / cost sharing: many developers need a product, so it makes sense to share development costs ([X Window System](http://en.wikipedia.org/wiki/X_Window_System) and the Apache web server)

Since Open Source software is open, all of the defects and security flaws are easily found. Closed-source advocates argue that this makes it easier for a malicious person to discover security flaws. Further, that there is no incentive for an open-source product to be patched. Open-source advocates argue that this makes it easier also for a patch to be found and that the closed-source argument is [security through obscurity](http://en.wikipedia.org/wiki/Security_through_obscurity), which this form of security will eventually fail, often without anyone knowing of the failure. Further, that just because there is not an immediate financial incentive to patch a product, does not mean there is not any incentive to patch a product. Further, if the patch is that significant to the user, having the source code, the user can technically patch the problem themselves. These arguments are hard to prove. However, research indicates [[19]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-18) that open-source software does have a higher flaw discovery, quicker flaw discovery, and quicker turn around on patches.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=14)**] Open source software vs. free software**

*Main article:* [*Alternative terms for free software*](http://en.wikipedia.org/wiki/Alternative_terms_for_free_software)

The main difference is that by choosing one term over the other (i.e. either "open source" or "free software") you let others know about what your goals are.

["As one person put it, 'Open source is a development methodology; free software is a social movement.'"](http://www.gnu.org/philosophy/free-software-for-freedom.html)

Critics have said that the term “open source” fosters an ambiguity of a different kind such that it confuses the mere availability of the source with the freedom to use, modify, and redistribute it. Developers have used the [alternative terms](http://en.wikipedia.org/wiki/Alternative_terms_for_free_software) *Free/open source Software* ([FOSS](http://en.wikipedia.org/wiki/FOSS)), or *Free/Libre/open source Software* (FLOSS), consequently, to describe open source software which is also free software.

The term “Open Source” was originally intended to be trademarkable; however, the term was deemed too descriptive, so no trademark exists[[20]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-19). The OSI would prefer that people treat Open Source as if it were a trademark, and use it only to describe software licensed under an OSI approved license.[[21]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-20).

**OSI Certified** is a trademark licensed only to people who are distributing software licensed under a license listed on the [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative)'s list[[22]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-21).

Open source software and free software are different terms for software which comes with certain rights, or freedoms, for the user. They describe two approaches and [philosophies](http://en.wikipedia.org/wiki/Philosophy) towards free software. *Open source* and *free software* (or *software libre*) both describe software which is free from onerous licensing restrictions. It may be used, copied, studied, modified and redistributed without restriction. Free software is not the same as [freeware](http://en.wikipedia.org/wiki/Freeware), software available at zero price.

The definition of open source software was written to be almost identical to the [free software definition](http://en.wikipedia.org/wiki/Free_software_definition)[[23]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070616-22). There are very few cases of software that is free software but is not open source software, and vice versa. The difference in the terms is where they place the emphasis. “Free software” is defined in terms of giving the user freedom. This reflects the goal of the [free software movement](http://en.wikipedia.org/wiki/Free_software_movement). “Open source” highlights that the source code is viewable to all and proponents of the term usually emphasize the quality of the software and how this is caused by the development models which are possible and popular among free and open source software projects.

Free software licenses are not written exclusively by the FSF. The FSF and the OSI both list licenses which meet their respective definitions of free software.

The FSF believes that knowledge of the concept of freedom is an essential requirement[[23]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070616-22)[[24]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070619-23), insists on the use of the term *free*[[23]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070616-22)[[24]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070619-23), and separates itself from the open source movement[[23]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070616-22)[[24]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-Stallman20070619-23).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=15)**] Open source vs. source-available**

Although the OSI definition of "open source software" is widely accepted, a small number of people and organizations use the term to refer to 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*](http://en.wikipedia.org/wiki/Shared_source), a term coined by Microsoft.

[Michael Tiemann](http://en.wikipedia.org/wiki/Michael_Tiemann), president of OSI, had criticized[[25]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-24) companies such as [SugarCRM](http://en.wikipedia.org/wiki/SugarCRM) for promoting their software as "open source" when in fact it did not have an OSI-approved license. In [SugarCRM](http://en.wikipedia.org/wiki/SugarCRM)'s case, it was because the software is so-called "[badgeware](http://en.wikipedia.org/wiki/Badgeware%22%20%5Co%20%22Badgeware)"[[26]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-25) since it specified a "badge" that must be displayed in the user interface ([SugarCRM](http://en.wikipedia.org/wiki/SugarCRM%22%20%5Co%20%22SugarCRM) has since switched to [GPLv3](http://en.wikipedia.org/wiki/GPLv3)[[27]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-26)). Another example is [Scilab](http://en.wikipedia.org/wiki/Scilab), which calls itself "the open source platform for numerical computation"[[28]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-27) but has a license[[29]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-28) that forbids commercial redistribution of modified versions. Because OSI does not have a registered [trademark](http://en.wikipedia.org/wiki/Trademark) for the term "open source", its legal ability to prevent such usage of the term is limited, but Tiemann advocates using public opinion from OSI, customers, and community members to pressure such organizations to change their license or to use a different term.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=16)**] Pros and cons**

Software experts and researchers on open source software have identified 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. It has also helped build developer loyalty as developers feel empowered and have a sense of ownership of the end product[[30]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-Sharma2002-29). Moreover less costs of marketing and logistical services are needed for OSS. It also helps companies to keep abreast of all technology developments. It is a good tool to promote a company's image, including its commercial products[[31]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-30). The OSS development approach has helped produce reliable, high quality software quickly and inexpensively. Besides, it 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[[32]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-31). Moreover 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[[33]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-32).

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[[34]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-33)[[35]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-34). In the most complex projects these rules may be as strict as reviewing even minor change by two independent developers[[36]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-35).

Not all OSS initiatives have been successful, for example, SourceXchange and Eazel[[30]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-Sharma2002-29). 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 (collected data concerning productivity and quality)[[13]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-autogenerated1-12). 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[[13]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-autogenerated1-12). 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[[37]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-36).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=17)**] Development tools**

|  |  |
| --- | --- |
| Question book-new.svg | This section **does not** [**cite**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources) **any** [**references or sources**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability).Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit) by adding citations to [reliable sources](http://en.wikipedia.org/wiki/Wikipedia%3AReliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3ACitation_needed) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(January 2008)* |

In OSS development the participants, who are mostly volunteers, are distributed amongst different geographic regions so there is need for tools to aid participants to collaborate in source code development. Often these tools are also available as OSS.

[Revision control](http://en.wikipedia.org/wiki/Revision_control) systems such as [Concurrent Versions System](http://en.wikipedia.org/wiki/Concurrent_Versions_System) (CVS) and later [Subversion](http://en.wikipedia.org/wiki/Subversion_%28software%29) (svn) are examples of tools that help centrally manage the source code files and the changes to those files for a software project.

Utilities that automate testing, compiling and bug reporting help preserve stability and support of software projects that have numerous developers but no managers, quality controller or technical support. Building systems that report compilation errors among different platforms include [Tinderbox](http://en.wikipedia.org/wiki/Tinderbox_%28software%29). Commonly used [bugtrackers](http://en.wikipedia.org/wiki/Bugtracker) include [Bugzilla](http://en.wikipedia.org/wiki/Bugzilla) and [GNATS](http://en.wikipedia.org/wiki/GNATS).

Tools such as [mailing lists](http://en.wikipedia.org/wiki/Mailing_lists), [IRC](http://en.wikipedia.org/wiki/Internet_Relay_Chat), and [instant messaging](http://en.wikipedia.org/wiki/Instant_messaging) provide means of Internet communications between developers. The Web is also a core feature of all of the above systems. Some sites centralize all the features of these tools as a [software development management system](http://en.wikipedia.org/w/index.php?title=Software_development_management_system&action=edit&redlink=1), including [GNU Savannah](http://en.wikipedia.org/wiki/GNU_Savannah), [SourceForge](http://en.wikipedia.org/wiki/SourceForge), and [BountySource](http://en.wikipedia.org/wiki/BountySource).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=18)**] Projects and organizations**

Some of the more prominent organizations involved in OSS development include the [Apache Software Foundation](http://en.wikipedia.org/wiki/Apache_Software_Foundation), creators of the [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server) web server; a loose affiliation of developers headed by [Linus Torvalds](http://en.wikipedia.org/wiki/Linus_Torvalds), creators of the [Linux](http://en.wikipedia.org/wiki/Linux_kernel) operating system kernel; the [Eclipse Foundation](http://en.wikipedia.org/wiki/Eclipse_Foundation), home of the [Eclipse](http://en.wikipedia.org/wiki/Eclipse_%28software%29) software development platform; the [Debian Project](http://en.wikipedia.org/wiki/Debian), creators of the influential Debian GNU/Linux distribution; the [Mozilla Foundation](http://en.wikipedia.org/wiki/Mozilla_Foundation), home of the [Firefox](http://en.wikipedia.org/wiki/Mozilla_Firefox) web browser; and [ObjectWeb](http://en.wikipedia.org/wiki/ObjectWeb), European-born community developing open-source middleware. New organizations tend to have a more sophisticated governance model and their membership is often formed by legal entity members[[38]](http://en.wikipedia.org/wiki/Open-source_software%22%20%5Cl%20%22cite_note-37).

Several Open Source programs have become defining entries in their space, including the [GIMP](http://en.wikipedia.org/wiki/GIMP) image editing system; [Sun's](http://en.wikipedia.org/wiki/Sun_Microsystems) [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) programming language and environment; the [MySQL](http://en.wikipedia.org/wiki/MySQL) database system; the [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD) Unix operating system; [Sun's](http://en.wikipedia.org/wiki/Sun_Microsystems) 2 [OpenOffice.org](http://en.wikipedia.org/wiki/OpenOffice.org) office productivity suite; and the [Wireshark](http://en.wikipedia.org/wiki/Wireshark) network [packet sniffer](http://en.wikipedia.org/wiki/Packet_sniffer) and protocol analyser.

Open Source development is often performed "live and in public", using services provided for free on the Internet, such as the [Launchpad](http://en.wikipedia.org/wiki/Launchpad_%28website%29) and [SourceForge](http://en.wikipedia.org/wiki/SourceForge.net) web sites, and using tools that are themselves Open Source, including the [CVS](http://en.wikipedia.org/wiki/Concurrent_Versions_System) and [Subversion](http://en.wikipedia.org/wiki/Subversion_%28software%29) source control systems, and the [GNU Compiler Collection](http://en.wikipedia.org/wiki/GNU_Compiler_Collection).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=19)**] Certification**

Certification can help to build higher user confidence. Certification could be applied to the simplest component that can be used by developers to build the simplest module to a whole software system. There have been numerous institutions involving in this area of the open source software including The International Institute of Software Technology / United Nations University (<http://www.iist.unu.edu>). UNU/IIST is a non-profit research and education institution of The United Nations. It is currently involved in a project known as "The Global Desktop Project". This project aims to build a desktop interface that every end-user is able to understand and interact with, thus crossing the language and cultural barriers. It is drawing huge attention from parties involved in areas ranging from application development to localization. Furthermore, this project will improve developing nations' access to information systems. UNU/IIST aims to achieve this without any compromise in the quality of the software. It believes a global standard can be maintained by introducing certifications and is currently organizing conferences in order to explore frontiers in the field (<http://opencert.iist.unu.edu>).

Alternatively, assurance models (such as [DO178B](http://en.wikipedia.org/wiki/DO178B)) have already solved the "certification" approach for software. This approach is tailorable and can be applied to OSS, but only if the requisite planning and execution, design, test and traceability artifacts are generated.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit&section=20)**] Criticism**

|  |  |
| --- | --- |
| Question book-new.svg | This section **needs additional** [**citations**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources#Inline_citations) **for** [**verification**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability)**.**Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Open-source_software&action=edit) by adding [reliable references](http://en.wikipedia.org/wiki/Wikipedia%3AReliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3AFact) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(July 2007)* |

The criticisms of the specific [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative) (OSI) principles are dealt with above as part of the definition and differentiation from other terms. The [open content](http://en.wikipedia.org/wiki/Open_content) movement does not recognize nor endorse the OSI principles and embraces instead mutual [share-alike](http://en.wikipedia.org/wiki/Share-alike) agreements that require commercial use or the preparation of [derivative works](http://en.wikipedia.org/wiki/Derivative_work).

Of the vocal critics, Richard Stallman of the [Free Software Foundation](http://en.wikipedia.org/wiki/Free_Software_Foundation) (FSF), flatly opposes the term “Open Source” being applied to what they refer to as “free software”. Although it's clear that legally free software does qualify as open source, he considers that the category is abusive.[[39]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-38) They also oppose the professed pragmatism of the [Open Source Initiative](http://en.wikipedia.org/wiki/Open_Source_Initiative), as they fear that the free software ideals of freedom and community are threatened by compromising on the FSF's idealistic standards for software freedom.[[40]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-39)[[41]](http://en.wikipedia.org/wiki/Open-source_software#cite_note-40)

Increasingly the consensus term [Free and Open Source Software](http://en.wikipedia.org/wiki/Free_and_Open_Source_Software) is used by the communities at large to describe the common ground between [Free Software](http://en.wikipedia.org/wiki/Free_Software) and Open Source.