

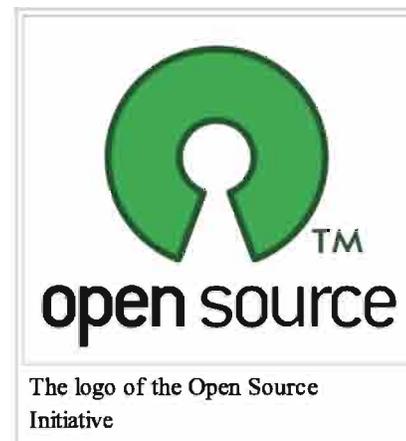
# Open-source software

From Wikipedia, the free encyclopedia

**Open-source software** (**OSS**) is computer software that is available in source code form: the source code and certain other rights normally reserved for copyright holders are provided under an open-source license that permits users to study, change, improve and at times also to distribute the software.

Open source software is very often developed in a public, collaborative manner. Open-source software is the most prominent example of open-source development and often compared to (technically defined) user-generated content or (legally defined) open content movements.<sup>[1]</sup>

A report by the Standish Group states that adoption of open-source software models has resulted in savings of about \$60 billion per year to consumers.<sup>[2][3]</sup>



## Contents

- 1 History
- 2 Definitions
  - 2.1 The Open Source Definition
  - 2.2 Perens' principles
- 3 Proliferation of the term
- 4 Non-software use
  - 4.1 Business models
- 5 Widely used open source products
- 6 Development philosophy
- 7 Licensing
- 8 Funding
- 9 Comparison with closed source
- 10 Comparison with free software
- 11 Open source vs. source-available
- 12 Pros and cons for software producers
- 13 Development tools
- 14 Projects and organizations
- 15 Certification
- 16 Criticism
- 17 See also
- 18 References
- 19 Further reading
  - 19.1 Legal and economic aspects
- 20 External links

## History

*Main article: Open source movement*

The free software movement was launched in 1983. In 1998, a group of individuals advocated that the term free software should be replaced by open source software (OSS) as an expression which is less ambiguous and more comfortable for the corporate world.<sup>[4]</sup> Software developers may want to publish their software with an open source license, so that anybody may also develop the same software or understand its internal functioning. With open source software, generally anyone is allowed to create modifications of it, port it to new operating systems and processor architectures, share it with others or, in some cases, market it. Scholars Casson and Ryan have pointed out several policy-based reasons for adoption of open source, in particular, the heightened value proposition from open source (when compared to most proprietary formats) in the following categories:

- Security.
- Affordability.
- Transparency.
- Perpetuity.
- Interoperability.
- Localization.<sup>[5]</sup>

Particularly in the context of local governments (who make software decisions), Casson and Ryan argue that "governments have an inherent responsibility and fiduciary duty to taxpayers" which includes the careful analysis of these factors when deciding to purchase proprietary software or implement an open-source option.<sup>[6]</sup>

The 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 grant rights to users which would otherwise be reserved by 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 (GPL), which "allows free distribution under the condition that further developments and applications are put under the same licence" – thus also free.<sup>[7]</sup> While open source distribution presents a way to make the source code 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 on April 7, 1998 in Palo Alto in reaction to Netscape's January 1998 announcement of a source code release for Navigator (as Mozilla). A group of individuals at the session included Tim O'Reilly, Linus Torvalds, Tom Paquin, Jamie Zawinski, Larry Wall, Brian Behlendorf, Sameer Parekh, Eric Allman, Greg Olson, Paul Vixie, John Ousterhout, Guido van Rossum, Philip Zimmermann, John Gilmore and Eric S. Raymond.<sup>[8]</sup> 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.

Many people claimed that the birth of the Internet, since 1969, started the open source movement, while others do not distinguish between open source and free software movements.<sup>[9]</sup>

The 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 (OSI) was formed in February 1998 by Eric S. Raymond and 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 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 for the OSI, but that attempt was impractical by 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.

## Definitions

The Open Source Initiative's definition is widely recognized as the standard or *de facto* definition.<sup>[citation needed]</sup>

The Open Source Initiative (OSI) was 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 the Debian Free Software Guidelines to make *The Open Source Definition*.<sup>[10]</sup>

### The Open Source Definition

The Open Source Initiative wrote a document called *The Open Source Definition* and uses it to determine whether it considers a software license open source. The definition was based on the Debian Free Software Guidelines, written and adapted primarily by Bruce Perens.<sup>[11][12]</sup> Perens did not base his writing on the "four freedoms" of Free Software from the FSF, which were only widely available later.<sup>[13]</sup>

## Perens' principles

Under Perens' definition, 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. The principles, as stated, say absolutely nothing about trademark or patent use and require absolutely no cooperation to ensure that any common audit or 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.

## Proliferation of the term

*Main article: Open source*

While the term "open source" applied originally only to the source code of software,<sup>[14]</sup> it is now being applied to many other areas such as Open source ecology,<sup>[15]</sup> 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.

## Non-software use

The principles of open source have been adapted for many forms of user generated content and technology, including open source hardware.

Supporters of the open content movement advocate some restrictions of use, requirements to share changes, and attribution to other authors of the work.

This "culture" or 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.<sup>[16]</sup>

Advocates of the open source principles often point to Wikipedia as an example, but Wikipedia has in fact often restricted certain types of use or user, and the GFDL license it has historically used makes specific requirements of all users, which technically violates the open source principles.

## Business models

*Main article: 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, 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 of commercial open-source software firms is the dual-license Strategy, as demonstrated by Ingres, MySQL, Alfresco, and others.

## 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, the e-commerce platform osCommerce and the internet browser Mozilla Firefox. One of the most successful open source products is the GNU/Linux operating system, an open source Unix-like operating system, and its derivative Android, an operating system for mobile devices.<sup>[17][18]</sup> In some fields, open software is the norm, like in voice over IP applications with Asterisk (PBX).

## Development philosophy

In his 1997 essay *The Cathedral and the Bazaar*,<sup>[19]</sup> open source evangelist 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".<sup>[19]</sup> 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 in his book *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<sup>[20]</sup> 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 states, "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 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.

Data suggests, however, that OSS is not quite as democratic as the bazaar model suggests. An analysis of five billion bytes of free/open source code by 31,999 developers shows that 74% of the code was written by the most active 10% of authors. The average number of authors involved in a project was 5.1, with the median at 2.<sup>[21]</sup>

## Licensing

*Main article: 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; whereupon 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 / open source licenses include Apache License, BSD license, GNU General Public License, GNU Lesser General Public License, MIT License, Eclipse Public License and 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. With more than 180,000 open source projects available and its more than 1400 unique licenses, the complexity of deciding how to manage open source usage within "closed-source" commercial enterprises have dramatically increased. Some are home-grown while others are modeled after mainstream FOSS licenses such as Berkeley Software Distribution ("BSD"), Apache, MIT-style (Massachusetts Institute of Technology), or GNU General Public License ("GPL"). In view of this, open source practitioners are starting to use classification schemes in which FOSS licenses are grouped (typically based on the existence and obligations imposed by the copyleft provision; the strength of the copyleft provision).<sup>[22]</sup>

An important legal milestone for the open source / free software movement was passed in 2008, when the US federal appeals court ruled that 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.<sup>[23]</sup>

## Funding

*See also: 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, 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 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, CMU Common Lisp, or the NCSA HTTPd which evolved into Apache.

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 local bug-fix or solution to a software problem, written by a developer either at a company's request or to make his/her own job easier, can be released as an open source contribution without costing the company anything.<sup>[7]</sup> 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.

## Comparison with closed source

*Main article: Comparison of open source and closed source*

The debate over *open source* vs. *closed source* (alternatively called proprietary software) is sometimes heated.

The top four reasons (as provided by Open Source Business Conference survey<sup>[24]</sup>) individuals or organizations choose open-source software are: 1) lower cost, 2) security, 3) no vendor 'lock in', and 4) better quality.

Since innovative companies no longer rely heavily on software sales, proprietary software has become less of a necessity.<sup>[25]</sup> As such, things like open-source content management system -- or CMS—deployments are becoming more commonplace. In 2009,<sup>[26]</sup> the US White House switched its CMS system from a proprietary system to Drupal open-source CMS. Further, companies like Novell (who traditionally sold software the old-fashioned way) continually debate the benefits of switching to open-source availability, having already switched part of the product offering to open-source code.<sup>[27]</sup> In this way, open-source software provides solutions to unique or specific problems. As such, it is reported <sup>[28]</sup> that 98% of enterprise-level companies use open-source offerings in some capacity.

With this market shift, more critical systems are beginning to rely on open-source offerings,<sup>[29]</sup> allowing greater funding (such as US Department of Homeland Security grants <sup>[29]</sup>) to help "hunt for security bugs."

This is not to argue that open-source software does not have its flaws. One of the greatest barriers facing wide acceptance of open-source software relates to the lack of technical and general support.<sup>[24]</sup> Open-source companies often combat this by offering support sometimes under a different product name. Acquia provides enterprise-level support for its open-source alternative, Drupal, for instance.<sup>[30]</sup>

Many open-source advocates argue that open-source software is inherently safer because any person can view, edit, and change code.<sup>[31]</sup> But closed-source software—and some research<sup>[32]</sup> -- suggests that individuals that aren't paid to scrub code have no incentive to do the boring, monotonous work.

Research indicates <sup>[33]</sup> that the open-source software - Linux - has a lower percentage of bugs than some commercial software.

## Comparison with free software

*Main article: 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") one lets others know about what one's goals are. As Richard Stallman puts it, "Open source is a development methodology; free software is a social movement."<sup>[34]</sup>

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 *Free/open source Software* (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.<sup>[35]</sup> 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.<sup>[36]</sup>

**OSI Certified** is a trademark licensed only to people who are distributing software licensed under a license listed on the Open Source Initiative's list.<sup>[37]</sup>

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 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, software available at zero price.

The definition of open source software was written to be almost identical to the free software definition.<sup>[38]</sup> 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. "Open source" highlights that the source code is viewable to all; 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 or open source software.

The FSF believes that knowledge of the concept of freedom is an essential requirement,<sup>[38][39]</sup> insists on the use of the term *free*,<sup>[38][39]</sup> and separates itself from the open source movement.<sup>[38][39]</sup>

## 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*, a term coined by Microsoft.

Michael Tiemann, president of OSI, had criticized<sup>[40]</sup> companies such as SugarCRM for promoting their software as "open source" when in fact it did not have an OSI-approved license. In SugarCRM's case, it was because the software is so-called "badgeware"<sup>[41]</sup> since it specified a "badge" that must be displayed in the user interface (SugarCRM has since switched to GPLv3<sup>[42]</sup>). Another example was Scilab prior to version 5, which called itself "the open source platform for numerical computation"<sup>[43]</sup> but had a license<sup>[44]</sup> that forbade commercial redistribution of modified versions. Because OSI does not have a registered 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.

## Pros and cons for software producers

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<sup>[citation needed]</sup>. It has also helped build developer loyalty as developers feel empowered and have a sense of ownership of the end product.<sup>[45]</sup> 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.<sup>[46]</sup> The OSS development approach has helped produce reliable, high quality software quickly and inexpensively.<sup>[47]</sup> The term "open source" was originally intended to be trademarkable; however, the term was deemed too descriptive, so no trademark exists. 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.<sup>[48]</sup> Moreover free software can be developed in accord with purely technical requirements. It does not require thinking 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.<sup>[49]</sup>

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.<sup>[50][51]</sup> In the most complex projects these rules may be as strict as reviewing even minor change by two independent developers.<sup>[52]</sup>

Not all OSS initiatives have been successful, for example, SourceXchange and Eazel.<sup>[45]</sup> 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).<sup>[16]</sup> 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.<sup>[16]</sup> 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.<sup>[53]</sup>

## Development tools

In OSS development, the participants, who are mostly volunteers, are distributed among 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 systems such as Concurrent Versions System (CVS) and later Subversion (svn) and Git 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. Commonly used bugtrackers include Bugzilla and GNATS.

Tools such as mailing lists, IRC, and instant messaging provide means of Internet communication 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, including GNU Savannah, SourceForge, and BountySource.

## Projects and organizations

One of the benefits of open source software is that there are a wide variety of codes in oss projects for program developers. Without any blocking of this wide and diverse platform, developers create a wide range of projects and organizations. Some of the "more prominent organizations" involved in OSS development include the Apache Software Foundation, creators of the Apache web server; a loose affiliation of developers headed by Linus Torvalds, creators of the Linux operating system kernel; the Eclipse Foundation, home of the Eclipse software development platform; the Debian Project, creators of the influential Debian GNU/Linux distribution; the Mozilla Foundation, home of the Firefox web browser; and OW2, 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.<sup>[54]</sup>

Several Open Source programs have become defining entries in their space, including the GIMP image editing system; Sun's Java programming language and environment; the MySQL database system; the FreeBSD Unix operating system; Sun's OpenOffice.org office productivity suite; and the Wireshark network 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 and SourceForge web sites, and using tools that are themselves Open Source, including the CVS and Subversion source control systems, and the GNU Compiler Collection.

Open Source for America is a group created to raise awareness in the U.S. Federal Government about the benefits of open source software. Their stated goals are to encourage the government's utilization of open source software, participation in open source software projects, and incorporation of open source community dynamics to increase government transparency.<sup>[55]</sup>

Mil-OSS is a group dedicated to the advancement of OSS use and creation in the military.<sup>[56]</sup>

## 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) 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.

## Criticism

*See also: Free software#Criticism*

The criticisms of the specific Open Source Initiative (OSI) principles are dealt with above as part of the definition and differentiation from other terms. The open content movement does not recognize nor endorse the OSI principles and embraces instead mutual share-alike agreements that require commercial use or the preparation of derivative works.

Of the vocal critics, Richard Stallman of the FSF, flatly opposes the term "Open Source" being applied to what they refer to as "free software". Although it is clear that legally free software does qualify as open source, Stallman considers that the category is abusive.<sup>[57]</sup> Critics also oppose the professed pragmatism of the 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.<sup>[58][59]</sup>

Increasingly<sup>[*citation needed*]</sup>, the consensus term "free and open-source software" is used by the communities at large to describe the

common ground between free software and open-source software.<sup>[*citation needed*]</sup>

## See also

- Open source
- Free software
- Free alternatives to proprietary software
- Free software community
- List of open source software packages
- Open source advocacy
- Open Source Initiative
- Open source software security
- Open source video games
- Business models for open source software
- Shared source
- Timeline of open source software
- Open source hardware

## References

1. ↑ Verts, William T. (2008-01-13). "Open source software" (<http://www.worldbookonline.com/wb/Article?id=ar751706>) . *World Book Online Reference Center*. <http://www.worldbookonline.com/wb/Article?id=ar751706>.
2. ↑ Rothwell, Richard (2008-08-05). "Creating wealth with free software" ([http://www.freesoftwaremagazine.com/community\\_posts/creating\\_wealth\\_free\\_software](http://www.freesoftwaremagazine.com/community_posts/creating_wealth_free_software)) . *Free Software Magazine*. [http://www.freesoftwaremagazine.com/community\\_posts/creating\\_wealth\\_free\\_software](http://www.freesoftwaremagazine.com/community_posts/creating_wealth_free_software). Retrieved 2008-09-08.
3. ↑ "Standish Newsroom - Open Source" ([http://standishgroup.com/newsroom/open\\_source.php](http://standishgroup.com/newsroom/open_source.php)) (Press release). Boston. 2008-04-16. [http://standishgroup.com/newsroom/open\\_source.php](http://standishgroup.com/newsroom/open_source.php). Retrieved 2008-09-08.
4. ↑ Raymond, Eric S. (1998-02-08). "Goodbye, "free software"; hello, "open source"" (<http://www.catb.org/~esr/open-source.html>) . <http://www.catb.org/~esr/open-source.html>. Retrieved 2008-08-13.
5. ↑ Casson and Ryan, Open Standards, Open Source Adoption in the Public Sector, and Their Relationship to Microsoft's Market Dominance ([http://papers.ssrn.com/so3/papers.cfm?abstract\\_id=1656616](http://papers.ssrn.com/so3/papers.cfm?abstract_id=1656616))
6. ↑ Id.
7. ↑ *ab* Holtgrewe, Ursula (2004). "Articulating the Speed(s) of the Internet: The Case of Open Source/Free Software.". *Time Society* **13**: 129–146. DOI:10.1177/0961463X04040750 (<http://dx.doi.org/10.1177%2F0961463X04040750>) .
8. ↑ "Open Source Pioneers Meet in Historic Summit" (<http://press.oreilly.com/pub/pr/796>) . 1998-04-14. <http://press.oreilly.com/pub/pr/796>. Retrieved 2010-08-19.
9. ↑ Muffatto, Moreno (2006). *Open Source: A Multidisciplinary Approach*. Imperial College Press. ISBN 1-86094-665-8.
10. ↑ Perens, Bruce. Open Sources: Voices from the Open Source Revolution (<http://www.oreilly.com/catalog/opensources/book/perens.html>) . O'Reilly Media. 1999.
11. ↑ The Open Source Definition by Bruce Perens (<http://www.oreilly.com/catalog/opensources/book/perens.html>) . January 1999. ISBN 1-56592-582-3. <http://www.oreilly.com/catalog/opensources/book/perens.html>.
12. ↑ "*The Open Source Definition*" (<http://opensource.org/docs/osd>) . <http://opensource.org/docs/osd>., The Open Source Definition according to the Open Source Initiative
13. ↑ "How Many Open Source Licenses Do You Need? - Slashdot" (<http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815>) . *News.slashdot.org*. 2009-02-16. <http://news.slashdot.org/comments.pl?sid=1129863&cid=26875815>. Retrieved 2012-03-25.
14. ↑ Stallman, Richard (September 24, 2007). "Why "Open Source" misses the point of Free Software" (<http://www.gnu.org/philosophy/open-source-misses-the-point.html>) . *Philosophy of the GNU Project*. Free Software Foundation. <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. Retrieved December 6, 2007. "However, not all of the users and developers of free software agreed with the goals of the free software movement. In 1998, a part of the free software community splintered off and began campaigning in the name of 'open source.' The term was originally proposed to avoid a possible misunderstanding of the term 'free software,' but it soon became associated with philosophical views quite different from those of the free software movement."
15. ↑ "*Open Source Ecology*" (<http://www.openfarmtech.org>) . <http://www.openfarmtech.org>. "...building the world's first replicable open source self-sufficient decentralized high-appropriate-tech permaculture ecovillage..."
16. ↑ *abc* Raymond, Eric S. *The Cathedral and the Bazaar*. ed 3.0. 2000.
17. ↑ Michael J. Gallivan, "Striking a Balance Between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies", *Info Systems Journal* 11 (2001): 277–304
18. ↑ Hal Plotkin, "What (and Why) you should know about open-source software" *Harvard Management Update* 12 (1998): 8-9
19. ↑ *ab* Raymond, Eric S. (2000-09-11). "The Cathedral and the Bazaar" (<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>) . <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>. Retrieved 2004-09-19.

20. ^ Robles, Gregorio (2004). "A Software Engineering Approach to Libre Software" (<http://www.opensourcejahrung.de/2004/pdfs/III-3-Robles.pdf>) . In Robert A. Gehring, Bernd Lutterbeck (PDF). *Open Source Jahrbuch 2004* (<http://www.opensourcejahrung.de>) . Berlin: Technical University of Berlin. <http://www.opensourcejahrung.de/2004/pdfs/III-3-Robles.pdf>. Retrieved 2005-04-20.
21. ^ Ghosh, R.A.; Robles, G. and Glott, R. (2002). "Free/Libre and Open Source Software: Survey and Study Part V.". *Maastricht: International Institute of Infonomics.*
22. ^ Andrew T. Pham, Verint Systems Inc. and Matthew B. Weinstein and Jamie L. Ryerson. "Easy as ABC: Categorizing Open Source Licenses (<http://www.ipo.org/AM/Template.cfm?Section=Patents&Template=/CM/ContentDisplay.cfm&ContentID=26682>) "; [www.IPO.org](http://www.IPO.org). June 2010.
23. ^ Shiels, Maggie (2008-08-14). "Legal milestone for open source" (<http://news.bbc.co.uk/1/hi/technology/7561943.stm>) . BBC News. <http://news.bbc.co.uk/1/hi/technology/7561943.stm>. Retrieved 2008-08-15.
24. ^ *a b* Irina Guseva (@irina\_guseva) (2009-03-26). "Bad Economy Is Good for Open Source" (<http://www.cmswire.com/cms/web-cms/bad-economy-is-good-for-open-source-004187.php>) . *Cmswire.com*. <http://www.cmswire.com/cms/web-cms/bad-economy-is-good-for-open-source-004187.php>. Retrieved 2012-03-25.
25. ^ "Open Source vs. Proprietary Software | PCWorld Business Center" ([http://www.pcworld.com/businesscenter/article/243136/open\\_source\\_vs\\_proprietary\\_software.html](http://www.pcworld.com/businesscenter/article/243136/open_source_vs_proprietary_software.html)) . *Pcworld.com*. 2011-11-03. [http://www.pcworld.com/businesscenter/article/243136/open\\_source\\_vs\\_proprietary\\_software.html](http://www.pcworld.com/businesscenter/article/243136/open_source_vs_proprietary_software.html). Retrieved 2012-03-25.
26. ^ Geoff Spick (@Goffee71) (2009-10-26). "Open Source Movement Finds Friends at the White House" (<http://www.cmswire.com/cms/web-cms/open-source-movement-finds-friends-at-the-white-house-005867.php>) . *Cmswire.com*. <http://www.cmswire.com/cms/web-cms/open-source-movement-finds-friends-at-the-white-house-005867.php>. Retrieved 2012-03-25.
27. ^ "Pandora's box for open source - CNET News" ([http://news.cnet.com/Pandoras-box-for-open-source/2009-7344\\_3-5157470.html](http://news.cnet.com/Pandoras-box-for-open-source/2009-7344_3-5157470.html)) . *News.cnet.com*. 2004-02-12. [http://news.cnet.com/Pandoras-box-for-open-source/2009-7344\\_3-5157470.html](http://news.cnet.com/Pandoras-box-for-open-source/2009-7344_3-5157470.html). Retrieved 2012-03-25.
28. ^ Murphy, David (2010-08-15). "Survey: 98 Percent of Companies Use Open-Source, 29 Percent Contribute Back | News & Opinion" (<http://www.pcmag.com/article2/0,2817,2367829,00.asp>) . *PCMag.com*. <http://www.pcmag.com/article2/0,2817,2367829,00.asp>. Retrieved 2012-03-25.
29. ^ *a b* "Homeland Security helps secure open-source code - CNET News" ([http://news.cnet.com/Homeland-Security-helps-secure-open-source-code/2100-1002\\_3-6025579.html](http://news.cnet.com/Homeland-Security-helps-secure-open-source-code/2100-1002_3-6025579.html)) . *News.cnet.com*. [http://news.cnet.com/Homeland-Security-helps-secure-open-source-code/2100-1002\\_3-6025579.html](http://news.cnet.com/Homeland-Security-helps-secure-open-source-code/2100-1002_3-6025579.html). Retrieved 2012-03-25.
30. ^ "Drupal - Content Management, Social Business, Cloud, Support" (<http://www.acquia.com>) . Acquia. <http://www.acquia.com>. Retrieved 2012-03-25.
31. ^ Seltzer, Larry (2004-05-04). "Is Open-Source Really Safer?" (<http://www.pcmag.com/article2/0,2817,1566726,00.asp>) . *PCMag.com*. <http://www.pcmag.com/article2/0,2817,1566726,00.asp>. Retrieved 2012-03-25.
32. ^ Kevin Poulsen (2004-01-30). "DARPA-funded Linux security hub withers" (<http://www.securityfocus.com/news/7947>) . *Securityfocus.com*. <http://www.securityfocus.com/news/7947>. Retrieved 2012-03-25.
33. ^ As a case in point, the study Linux: Fewer Bugs Than Rivals (<http://www.wired.com/software/coolapps/news/2004/12/66022>) reports that linux source code has 0.17 bugs per 1000 lines of code while non-Open-Source commercial software generally scores 20-30 bugs per 1000 lines
34. ^ "Why "Free Software" is better than "Open Source"" (<http://www.gnu.org/philosophy/free-software-for-freedom.html>) . <http://www.gnu.org/philosophy/free-software-for-freedom.html>.
35. ^ Nelson, Russell (2007-03-26). "Certification Mark" ([http://web.archive.org/web/20080206050627/http://www.opensource.org/docs/certification\\_mark.html](http://web.archive.org/web/20080206050627/http://www.opensource.org/docs/certification_mark.html)) . Open Source Initiative. Archived from the original ([http://opensource.org/docs/certification\\_mark.html](http://opensource.org/docs/certification_mark.html)) on 2008-02-06. [http://web.archive.org/web/20080206050627/http://www.opensource.org/docs/certification\\_mark.html](http://web.archive.org/web/20080206050627/http://www.opensource.org/docs/certification_mark.html). Retrieved 2007-07-22.
36. ^ Raymond, Eric S. (1998-11-22). "OSI Launch Announcement" (<http://www.opensource.org/pressreleases/osi-launch.php>) . Open Source Initiative. <http://www.opensource.org/pressreleases/osi-launch.php>. Retrieved 2007-07-22.
37. ^ Nelson, Russell (2006-09-19). "Open Source Licenses by Category" (<http://opensource.org/licenses>) . Open Source Initiative. <http://opensource.org/licenses>. Retrieved 2007-07-22.
38. ^ *a b c d* Stallman, Richard (2007-06-16). "Why "Open Source" misses the point of Free Software" (<http://www.gnu.org/philosophy/open-source-misses-the-point.html>) . *Philosophy of the GNU Project*. GNU Project. <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. Retrieved 2007-07-23.
39. ^ *a b c* Stallman, Richard (2007-06-19). "Why "Free Software" is better than "Open Source"" (<http://www.gnu.org/philosophy/free-software-for-freedom.html>) . *Philosophy of the GNU Project*. GNU Project. <http://www.gnu.org/philosophy/free-software-for-freedom.html>. Retrieved 2007-07-23.
40. ^ Tiemann, Michael (2007-06-21). "Will The Real Open Source CRM Please Stand Up?" (<http://www.opensource.org/node/163>) . Open Source Initiative. <http://www.opensource.org/node/163>. Retrieved 2008-01-04.
41. ^ Berlind, David (2006-11-21). "Are SugarCRM, Socialtext, Zimbra, Scalix and others abusing the term "open source?"" (<http://blogs.zdnet.com/BTL/index.php?p=3430>) . ZDNet. <http://blogs.zdnet.com/BTL/index.php?p=3430>. Retrieved 2008-01-04.
42. ^ Vance, Ashlee (2007-07-25). "SugarCRM trades badgeware for GPL 3" ([http://www.regdeveloper.co.uk/2007/07/25/sugarcrm\\_gp3/](http://www.regdeveloper.co.uk/2007/07/25/sugarcrm_gp3/)) . The Register. [http://www.regdeveloper.co.uk/2007/07/25/sugarcrm\\_gp3/](http://www.regdeveloper.co.uk/2007/07/25/sugarcrm_gp3/). Retrieved 2008-09-08.
43. ^ "The open source platform for numerical computation" (<http://www.scilab.org>) . INRIA. <http://www.scilab.org>. Retrieved 2008-01-04.
44. ^ "SCILAB License" (<http://www.scilab.org/legal/license.html>) . INRIA. <http://www.scilab.org/legal/license.html>. Retrieved 2008-01-04.
45. ^ *a b* Sharma, Srinarayan; Vijayan Sugumaran and Balaji Rajagopalan (2002). "A framework for creating hybrid-open source software communities" (<http://www.cin.ufpe.br/~in953/lectures/papers/ISJAFrameworkForCreatingHybrid-OpenSourceSoftwareCommunities.pdf>) (PDF). *Info Systems Journal* 12: 7–25. DOI:10.1046/j.1365-2575.2002.00116.x (<http://dx.doi.org/10.1046%2Fj.1365-2575.2002.00116.x>) .

<http://www.cin.ufpe.br/~in953/lectures/papers/ISJAFrameworkForCreatingHybrid-OpenSourceSoftwareCommunities.pdf>.

46. ^ Landry, John; Rajiv Gupta (September 2000). "Profiting from Open Source". *Harvard Business Review*. DOI:10.1225/F00503 (<http://dx.doi.org/10.1225%2FF00503>) .
47. ^ Reynolds, Carl; Jeremy Wyatt (February 2011). "Open Source, Open Standards, and Health Care Information Systems" (<http://www.jmir.org/2011/1/e24/>) . *JMIR* 13. DOI:10.2196/jmir.1521 (<http://dx.doi.org/10.2196%2Fjmir.1521>) . <http://www.jmir.org/2011/1/e24/>. Retrieved 2011-03-17.
48. ^ Plotkin, Hal (December 1998). "What (and Why) you should know about open-source software". *Harvard Management Update*: 8–9. DOI:10.1225/U9812D (<http://dx.doi.org/10.1225%2FU9812D>) .
49. ^ Payne, Christian (February 2002). "On the Security of Open Source Software". *Info Systems Journal* 12 (1): 61–78. DOI:10.1046/j.1365-2575.2002.00118.x (<http://dx.doi.org/10.1046%2Fj.1365-2575.2002.00118.x>) .
50. ^ "GNU Classpath Hacker's Guide: GNU Classpath Hacker's Guide" (<http://www.gnu.org/software/classpath/docs/hacking.html>) . *Gnu.org*. 2003-08-11. <http://www.gnu.org/software/classpath/docs/hacking.html>. Retrieved 2012-03-25.
51. ^ Meffert, Klaus; Neil Rotstan (2007). "Brief summary of coding style and practice used in JGAP" (<http://jgap.sourceforge.net/doc/codestyle.html>) . Java Genetic Algorithms Package. <http://jgap.sourceforge.net/doc/codestyle.html>. Retrieved 2008-09-08.
52. ^ Tripp, Andy (2007-07-16). "Classpath hackers frustrated with slow OpenJDK process" (<http://www.javalobby.org/java/forums/t98834.html>) . Javalobby. <http://www.javalobby.org/java/forums/t98834.html>.
53. ^ Gallivan, Michael J. (2001). "Striking a Balance Between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies". *Info Systems Journal* 11 (4): 277–304. DOI:10.1111/j.1365-2575.2001.00108.x (<http://dx.doi.org/10.1111%2Fj.1365-2575.2001.00108.x>) .
54. ^ François Letellier (2008), Open Source Software: the Role of Nonprofits in Federating Business and Innovation Ecosystems (<http://flet.netcipia.net/xwiki/bin/download/Main/publications%2Dfr/GEM2008%2DFLetellier%2DSubmittedPaper.pdf>) , AFME 2008.
55. ^ Hellekson, Gunnar. "Home" (<http://opensourceforamerica.org/>) . Open Source for America. <http://opensourceforamerica.org/>. Retrieved 2012-03-25.
56. ^ from EntandoSrl (Entando) . "Mil-OSS" (<http://mil-oss.org/>) . *Mil-OSS*. <http://mil-oss.org/>. Retrieved 2012-03-25.
57. ^ Stallman, Richard (June 16, 2007). "Why "Open Source" misses the point of Free Software" (<http://www.gnu.org/philosophy/open-source-misses-the-point.html>) . *Philosophy of the GNU Project*. Free Software Foundation. <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. Retrieved July 23, 2007. "As the advocates of open source draw new users into our community, we free software activists have to work even more to bring the issue of freedom to those new users' attention. We have to say, 'It's free software and it gives you freedom!'—more and louder than ever. Every time you say 'free software' rather than 'open source,' you help our campaign."
58. ^ Stallman, Richard (June 19, 2007). "Why "Free Software" is better than "Open Source"" (<http://www.gnu.org/philosophy/free-software-for-freedom.html>) . *Philosophy of the GNU Project*. Free Software Foundation. <http://www.gnu.org/philosophy/free-software-for-freedom.html>. Retrieved July 23, 2007. "Sooner or later these users will be invited to switch back to proprietary software for some practical advantage. Countless companies seek to offer such temptation, and why would users decline? Only if they have learned to value the freedom free software gives them, for its own sake. It is up to us to spread this idea—and in order to do that, we have to talk about freedom. A certain amount of the 'keep quiet' approach to business can be useful for the community, but we must have plenty of freedom talk too."
59. ^ Stallman, Richard (June 16, 2007). "Why "Open Source" misses the point of Free Software" (<http://www.gnu.org/philosophy/open-source-misses-the-point.html>) . *Philosophy of the GNU Project*. Free Software Foundation. <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. Retrieved July 23, 2007. "Under the pressure of the movie and record companies, software for individuals to use is increasingly designed specifically to restrict them. This malicious feature is known as DRM, or Digital Restrictions Management (see [DefectiveByDesign.org](http://defectivebydesign.org/) (<http://defectivebydesign.org/>)), and it is the antithesis in spirit of the freedom that free software aims to provide. [...] Yet some open source supporters have proposed 'open source DRM' software. Their idea is that by publishing the source code of programs designed to restrict your access to encrypted media, and allowing others to change it, they will produce more powerful and reliable software for restricting users like you. Then it will be delivered to you in devices that do not allow you to change it. This software might be 'open source,' and use the open source development model; but it won't be free software, since it won't respect the freedom of the users that actually run it. If the open source development model succeeds in making this software more powerful and reliable for restricting you, that will make it even worse."

## Further reading

- Androutsellis-Theotokis, Stephanos; Spinellis, Diomidis; Kechagia, Maria; Gousios, Georgios (2010). "Open source software: A survey from 10,000 feet" (<http://www.dmst.aueb.gr/dds/pubs/jrnl/2010-TOMS-OSS-Survey/html/ASKG10.pdf>) (PDF). *Foundations and Trends in Technology, Information and Operations Management* 4 (3-4): 187–347. DOI:10.1561/02000000026 (<http://dx.doi.org/10.1561%2F02000000026>) . ISBN 978-1-60198-484-5. <http://www.dmst.aueb.gr/dds/pubs/jrnl/2010-TOMS-OSS-Survey/html/ASKG10.pdf>
- Fadi P. Deek and James A. M. McHugh (2008). *Open Source: Technology and Policy*. Cambridge University Press. ISBN 978-0-511-36775-5.
- Chris DiBona and Sam Ockman and Mark Stone, ed. (1999). *Open Sources: Voices from the Open Source Revolution*. O'Reilly. ISBN 1-56592-582-3.
- Joshua Gay, ed. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman* (<http://www.gnu.org/philosophy/fsfs/rms-essays.pdf>) . GNU Press, Free Software Foundation. ISBN 1-882114-98-1.

<http://www.gnu.org/philosophy/fsfs/rms-essays.pdf>

- Free Open Source Software Book Author "Understanding FOSS ([http://books.google.co.in/books?id=2JwU71H5kikC&printsec=frontcover&dq=k.s.sampathkumar&source=bl&ots=wFNsEJ7tDL&sig=VJLpX2YBhdIzs6b69qGahpWPBo&h=en&ei=AC-1TcuRFctrQe1wI3IDQ&sa=X&oi=book\\_result&ct=result&resnum=5&ved=0CCkQ6AEwBA#v=onepage&q&f=false](http://books.google.co.in/books?id=2JwU71H5kikC&printsec=frontcover&dq=k.s.sampathkumar&source=bl&ots=wFNsEJ7tDL&sig=VJLpX2YBhdIzs6b69qGahpWPBo&h=en&ei=AC-1TcuRFctrQe1wI3IDQ&sa=X&oi=book_result&ct=result&resnum=5&ved=0CCkQ6AEwBA#v=onepage&q&f=false)) " **K.S.Sampathkumar Coimbatore India**

## Legal and economic aspects

- Benkler, Yochai (2002), "Coase's Penguin, or, Linux and The Nature of the Firm." *Yale Law Journal* 112.3 (Dec 2002): p367(78) (<http://www.benkler.org/CoasesPenguin.PDF>) (in Adobe pdf format)
- v. Engelhardt, Sebastian (2008) (PDF). "*The Economic Properties of Software*", *Jena Economic Research Papers, Volume 2 (2008), Number 2008-045* (<http://ideas.repec.org/p/jrp/jrpwpr/2008-045.html>) . <http://ideas.repec.org/p/jrp/jrpwpr/2008-045.html>
- Lerner, J. & Tirole, J. (2002): 'Some simple economics on open source', *Journal Of Industrial Economics* 50(2), p 197–234
- Välimäki, Mikko (2005) (PDF). *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry* ([http://pub.turre.com/openbook\\_valimaki.pdf](http://pub.turre.com/openbook_valimaki.pdf)) . Turre Publishing. [http://pub.turre.com/openbook\\_valimaki.pdf](http://pub.turre.com/openbook_valimaki.pdf)
- Polley, Barry (2007-12-11) (PDF). *Open Source Discussion Paper – version 1.0* ([http://nzoss.org.nz/system/files/moj\\_oss\\_strategy\\_1.0.pdf](http://nzoss.org.nz/system/files/moj_oss_strategy_1.0.pdf)) . New Zealand Ministry of Justice. [http://nzoss.org.nz/system/files/moj\\_oss\\_strategy\\_1.0.pdf](http://nzoss.org.nz/system/files/moj_oss_strategy_1.0.pdf) Retrieved 2007-12-12.
- Rossi, M. A. (2006): Decoding the free/open source software puzzle: A survey of theoretical and empirical contributions, in J. Bitzer P. Schröder, eds, 'The Economics of Open Source Software Development', p 15–55.

## External links

- Computerworld article: Does the open-source development model work for business users? ([http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=development&articleId=9066615&taxonomyId=11&intsrc=kc\\_top](http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=development&articleId=9066615&taxonomyId=11&intsrc=kc_top))
- The Open Source Initiative's definition of open source (<http://opensource.org/docs/definition.php>)
- Open Sources: Voices from the Open Source Revolution (<http://www.oreilly.com/catalog/opensources/book/toc.html>) — an online book containing essays from prominent members of the open source community
- Free / Open Source Research Community ([http://opensource.mit.edu/online\\_papers.php](http://opensource.mit.edu/online_papers.php)) — Many online research papers about Open Source
- FreeOpenSourceSoftware.org (<http://freeopensourcesoftware.org/>) - Wiki on Open Source history, organizations, licenses, people, software.
- KDE developer Aaron Siego's presentation "How OSS Improves Society" (<http://video.google.com/videoplay?docid=1755108258049541143&q=duration%3Along>) at the 2nd Trans-Pacific Open Source Software Conference in Honolulu, Hawaii, January 2006.
- Whence The Source: Untangling the Open Source/Free Software Debate* ([http://www.oreillynet.com/pub/a/oreilly/opensource/news/scoville\\_0399.html](http://www.oreillynet.com/pub/a/oreilly/opensource/news/scoville_0399.html)) , essay on the differences between Free Software and Open Source, by Thomas Scoville
- Berry, D M (2004). The Contestation of Code: A Preliminary Investigation into the Discourse of the Free Software and Open Software Movement, *Critical Discourse Studies*, Volume 1(1). (<http://opensource.mit.edu/papers/berry1.pdf>)
- EU study on adopting FLOSS ([http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact\\_en.pdf](http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact_en.pdf))
- Open Source Observatory and Repository (<http://www.osor.eu>)
- Calls for open source government (<http://news.bbc.co.uk/1/hi/technology/7841486.stm>)
- <http://osdir.com> "All things open"

Retrieved from "[http://en.wikipedia.org/w/index.php?title=Open-source\\_software&oldid=496613745](http://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=496613745)"

Categories: Free software | Software licenses

- 
- This page was last modified on 8 June 2012 at 15:43.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of use for details.
  - Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.