Proprietary Software and Closed Source

- Proprietary software is computer software which is the legal property of one party. The terms of use for other parties is defined by contracts or licensing agreements. These terms may include various privileges to share, alter, dissemble, and use the software and its code
- Closed source is a term for software whose license does not allow for the release or distribution of the software's source code. Generally, it means only the binaries of a computer program are distributed and the license provides no access to the program's source code. The source code of such programs is usually regarded as a **trade secret** of the company. Access to source code by third parties commonly requires the party to sign a **non-disclosure agreement**

Proprietary Software and Closed Source

- Under the closed source model, source code must be hidden from the public and competitors who might otherwise reproduce, study or modify the code, either to resell the product, learn from the product or for other reasons. Software companies that follow the closed source model see it as a **way to protect** their products from software piracy or misuse, from reverse engineering and duplication, and to maintain **competitive advantage** and **vendor lock-in**
- Closed source software usually is developed and maintained by a team who produces their product in a **compiled executable** state, which is what the market is allowed access to

Proprietary Software and Closed Source

- The primary business model for closed-source software involve the use of constraints on what can be done with the software and the restriction of access to the original source code. The end result is that an **end-user is not actually purchasing software, but purchasing the right to use the software**. To this end, the source code to closed-source software is considered a trade secret by its manufacturers

Free Software and Open Source

- In the 1970s and early 1980s, the software industry began using technical measures (such as only distributing binary copies of computer programs) to prevent computer users from being able to study and modify software
- In 1980 **copyright law** was extended to computer programs
- The free software movement was conceived in 1983 by Richard Stallman to satisfy the need for and to give the benefit of "software freedom" to computer users
- R. Stallman announced the plan for the GNU operating system in September 1983
- The GNU Manifesto was written by R. Stallman and published in March 1985

Free Software and Open Source

- The Free Software Foundation (FSF) is a non-profit corporation founded by R. Stallman on 4 October 1985 to support the free software movement, a copyleft-based movement which aims to promote the **universal freedom to distribute and modify computer software without restriction**.
- The first formal definition of free software was published by FSF in February 1986
- Alternatives for "free software" were sought for marketing purposes and because of a perceived "moralizing and confrontational" attitude that had been associated with the term. In addition, the "available at no cost" ambiguity of the word "free" was seen as discouraging business adoption. In a 1998 strategy session in California, "open-source software" was selected by Todd Anderson, Larry Augustin, Jon Hall, Sam Ockman, Christine Peterson, and Eric S. Raymond (a group of people very familiar with free software)

Free Software and Open Source

- Open Source Initiative (OSI) was soon-after founded by Eric Raymond (helped Netscape to go with free software) and Bruce Perens (leader of the Debian project) to promote the term as part of "**a marketing program for free software**"
- The first formal definition of open source software was published by OSI in 1998
- B. Perens of OSI, attempted to register "open source" as a service mark for OSI in the United States of America, but that attempt failed to meet the relevant trademark standards of specificity. OSI claims a trademark on "OSI Certified", and applied for trademark registration, but did not complete the paperwork
- The term "free software" is associated with FSF's definition, and the term "open-source software" is associated with OSI's definition. FSF's and OSI's definitions are worded quite differently but the set of software that they cover is almost identical

Free Software and Open Source

- For the Open Source movement, the issue of whether software should be open source is a **practical question**, **not an ethical one**. As one person put it, "open source is a development methodology; free software is a social movement"
- The formation of the Open Source Initiative, and the choice of the term "open source" was explicitly chosen to:
  "…dump the moralizing and confrontational attitude that had been associated with 'free software' in the past and sell the idea strictly on the same pragmatic, business-case grounds that had motivated Netscape…"
- R. Stallman criticized this motivation, saying that pragmatic focus of the initiative distracts users from the central moral issues and the freedoms offered by free software

Eric Steven Raymond (The Cathedral and the Bazaar)

- The development of what was later named Mozilla has been only a qualified success. It achieved **Netscape's original goal**, which was to **deny Microsoft a monopoly lock on the browser market**. It has also achieved some dramatic successes (notably the release of the **next-generation Gecko rendering engine**).
- However, **it has not yet garnered the massive development effort from outside Netscape that the Mozilla founders had originally hoped for**. The problem here seems to be that for a long time the Mozilla distribution actually broke one of the basic rules of the bazaar model; **it didn't ship with something potential contributors could easily run and see working**.
- Most negatively (from the point of view of the outside world) the Mozilla group **didn't ship a production-quality browser** for two and a half years after the project launch

Free Software and Open Source

- R. Stallman describes the free software and the Open Source Initiative as separate political camps within the same free software community and says: *"We disagree on the basic principles, but agree more or less on the practical recommendations. So we can and do work together on many specific projects."*
- For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution
- "Software Libre" was first used publicly in 2000, by the European Commission. The word "libre", borrowed from the Spanish and French languages, means having liberty. This avoids the freedom/cost ambiguity of the word "free"
- "FLOSS" was used in 2001 by Rishab Aiyer Ghosh as an acronym for free/libre/open-source software. Later that year, the European Commission (EC) used the phrase when they funded a study on the topic

Free Software and Open Source

- Unlike "libre software", which aimed to solve the ambiguity problem, "FLOSS" aimed to avoid taking sides in the debate over whether it was better to say "free software" or to say "open source software"
- The "Software Freedom Law Center" was founded in 2005 to protect and advance FLOSS
- R. Stallman endorses the term FLOSS to refer to "open source" and "free software" without necessarily choosing between the two camps, however, he asks people to consider supporting the "free software" camp
- In early 2002, the Department of Defense used the term "FOSS" in what would later be their 2003 report Use of Free and Open Source Software (FOSS) in the U.S. Department of Defense
- R. Stallman has criticized the term FOSS as Free/Open seems to imply free-of-charge, while Free/Libre makes it clearer the reference is to freedom

Free Software and Open Source

- All of the terms are used interchangeably, the choice of which to use is mostly **political** (wanting to support a certain group) or **practical** (thinking that one term is the clearest)

Free Software Definition

- The freedom to run the program, for any purpose (freedom 0)

  Placing restrictions on the use of Free Software, such as time ("30 days trial period", "license expires January 1st, 2004") purpose ("permission granted for research and non-commercial use", "may not be used for benchmarking") or geographic area ("must not be used in country X") makes a program non-free

Free Software Definition

- The freedom to study how the program works, and adapt it to your needs (freedom 1)

  Placing legal or practical restrictions on the comprehension or modification of a program, such as mandatory purchase of special licenses, signing of a Non-Disclosure-Agreement (NDA) or - for programming languages that have multiple forms or representation - making the preferred human way of comprehending and editing a program ("source code") inaccessible also makes it proprietary (non-free). Without the freedom to modify a program, people will remain at the mercy of a single vendor

Free Software Definition

- The freedom to redistribute copies so you can help your neighbor (freedom 2)

  Software can be copied/distributed at virtually no cost. If you are not allowed to give a program to a person in need, that makes a program non-free. This can be done for a charge, if you so choose

- The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits (freedom 3)

  Not everyone is an equally good programmer in all fields. Some people don't know how to program at all. This freedom allows those who do not have the time or skills to solve a problem to indirectly access the freedom to modify. This can be done for a charge

Free Software Definition

- These freedoms are rights, not obligations
- Free Software does not exclude commercial use
- If a program fails to allow commercial use and commercial distribution, it is not Free Software
- A growing number of companies base their business model completely or at least partially on Free Software
- Free Software makes it legal to provide help and assistance, it does not make it mandatory

Open Source Definition

The group adopted the Open Source Definition for open-source software, based on the Debian Free Software Guidelines, which in turn was based on The Free Software Definition.

1. Free Redistribution

   The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale

Open Source Definition

2. Source Code

   The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed

3. Derived Works

   The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software

Open Source Definition

4. Integrity of The Author's Source Code

   The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software

5. No Discrimination Against Persons or Groups

   The license must not discriminate against any person or group of persons

6. No Discrimination Against Fields of Endeavor

   The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research

Open Source Definition

7. Distribution of License

   The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties

8. License Must Not Be Specific to a Product

   The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution

Open Source Definition

9. License Must Not Restrict Other Software

   The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software

10. License Must Be Technology-Neutral

    No provision of the license may be predicated on any individual technology or style of interface