Ryan Stenmark

English 10 A

2015-10-26

Open source and closed source software development are both methods that work toward a common goal -- creating good software for users. However, the similarities end there. During the advent of personal computing, most software was closed-source, or proprietary. This means that the source code that the software is comprised of is only available through licensing, or in binary form, compiled down to a level that only a computer can understand. An example of closed-source software today is the popular operating system Windows, which comes preinstalled on almost every retail available computer. After a time, with the creation of ARPANET in 1969 and later the Internet, the idea of writing software for personal use, and releasing it for others to use, improve, and publish their improvements could become something bigger than small projects between friends. This is the heart of open-source.

Where closed-source software requires large, paid development teams to create, open-source projects are kept afloat by dedicated contributors, who in most cases receive no payment for their work. An example of open-source software today is the Linux kernel, and the many unique operating systems built around it, the most commonly used probably being the Android operating system on your phone. Open-source allows developers to take freely available software, and branch from it to make their own derivative or distribution of it that follows their design philosophy, and includes their improvements. These same developers can also contribute code directly to a project, in hopes that it is accepted and included in the next release version.

In closed-source projects, developers cannot read, edit, or redistribute the original software, or their changes, depending on the license used by the original developers. This is to prevent piracy, the unauthorized sharing of copyrighted software. Often, closed-source software is not free, and sale of the binary version of the software and the license to use it is where profit is made to continue development and cover development costs. Open-source software is always free as it must be, due to the source code being open. This complicates the business model for companies like Red Hat, the organization that oversees development of Red Hat Linux, an open source operating system used in many enterprise computer systems. While money cannot be made directly off the sale of free software, Red Hat stays in business through the promise of continuing to support and develop Red Hat Linux for their clients, as long as they have the financial ability to do so. This incentivizes their clients to provide funding.

Open source very much subscribes to the socialist-founded idea of 'what is mine, is also yours'. The entire culture of open source revolves around this idea. If you must take from an open source project, you must give back, usually in the form of a contribution of code. This contrasts from the capitalistic goals of closed sourced software development greatly. This mentality of taking while simultaneously giving back also has benefits in the form of peer review. Since any person can view or submit changes to an open source project, the chances of malicious code being intentionally packaged with a piece of software is almost nullified because it will nearly always be caught by other contributors before being pushed to a release version of the software.

To recap, closed source software development generally follows a capitalistic approach, their end goal is creating a piece of software that people will buy and use. On the opposite end of the spectrum, open source software development follows a somewhat socialist mindset, all in order to create software that will not only be used, but poked at by developers who want to make the software the best it can be for everyone. Clearly, despite having the same end goal, open-source and closed-source software development follow very different rules, and are the hosts of very different cultures.