Intellectual Property Rights

By Linda Westfall

https://www.softwareexcellenceacademy.com

I am not a lawyer, nor have I received any legal training. All legal issues and risks that arise within an organization on any software program, project, product, or contract should be referred to a lawyer or other legal professional. The descriptions below should be used for general information only.

Intellectual property

Intellectual property is a legal area that includes inventions and ideas of the human mind, such as books, music, artwork, and software. Intellectual property rights give the creators of these works exclusive rights to their creations. These rights deal with legal issues involving copyrighting, patenting, and licensing software products and trademarks. Intellectual property rights and the associated laws and regulations vary from country to country. A software practitioner should refer any questions or issues concerning intellectual property rights to the appropriate legal authority.

Patents

According to Futrell et al. (2002), "*Patents* protect ideas and are exclusive rights for novel inventions for a limited period of time." A patent owner has the right to exclude others from manufacturing, selling, or using products based on the patented idea or underlying concept for a specific time after the patent is issued. Kappos (2012), Under Secretary of Commerce for Intellectual property and Director of the United States Patent and Trademark Office, stated that "Patents are issued for process and apparatus, which are determined to be novel and non-

obvious. Because many breathtaking software-implemented innovations power our modern world, at levels of efficiency and performance unthinkable even just a few years ago, patent protection is every bit as well-deserved for software-implemented innovation as for the innovations that enabled man to fly … But it is equally important that patent protection be properly tailored in scope, so that programmers can write code and engineers can design devices without fear of unfounded accusations of infringement."

Copyrights

Copyrights protect original written works, such as software, from being copied without permission. Owning a copyright on a software product means that the owner(s) are protected under the law from other people copying, distributing, or making adaptations to their software products without their permission. Unlike a patent, a copyright does not protect the underlying idea or concept but only protects the tangible expression of that idea or concept. In the United States, fair use is a defense to copyright infringement. "Fair use is a legal doctrine that promotes freedom of expression by permitting the unlicensed use of copyright-protected works in certain circumstances. Section 107 of the Copyright Act provides the statutory framework for determining whether something is a fair use and identifies certain types of uses—such as criticism, comment, news reporting, teaching, scholarship, and research—as examples of activities that may qualify as fair use." (From copyright.gov/fair-use).

Licenses

Most software in the United States is sold under a licensing agreement rather than just depending on copyrights for protection. A typical *software license* grants the license holder the right to either use or redistribute one or more copies of copyrighted software without breaking copyright law. *Proprietary software licenses* grant limited rights to the user to use one or more copies of the software while the ownership of that software remains with the software development organization. Typically, proprietary software licenses have a well-defined list of restrictions on what the users can do with the software. The user is required to accept the terms of the proprietary software license in order to use the software.

In contrast to *proprietary software licenses*, the ownership of the specific copy of the software is transferred to the user with an *open source license*. However, the copyright ownership remains with the original software developer. The user may use the software without accepting the open source license, or the user can optionally accept the license, in which case the user is typically granted additional privileges. *Copyleft* is a form of licensing that allows individuals the right to disseminate copies of, or even modified versions of, a software product (or other types of works). These licenses typically stipulate that any adapted or derived versions of the original software must also be made available through the same or similar licenses. Examples of copyleft licenses are the GNU General Public License and Share-alike. Organizations developing software should take great care when using open source and other publically available software to verify that they remain in compliance with the associated licensing agreements, especially if they intend for their software products to remain proprietary.

Copyrights and licenses are also important when commercial-off-the-shelf (COTS) or third-party developed software is being integrated with and/or built into another software product. In this case, additional licensing requirements and/or royalty fees may be involved. Care must also be taken when reusing software components with integrated COTS or third party software to

verify that any licensing agreements or royalties are carried forward with those reused components.

Trademarks

Trademarks are devices used to "brand" products or services and distinguish them from other similar products or services in the marketplace. Trademarks can include words, names, slogans, logos, symbols, and even sounds or colors. A *service mark* is similar to a trademark except that it identifies and distinguishes a service rather than a product. In the United States (U.S.), the symbols "TM" for trademark and "SM" for service mark can be used "as soon as one intends to claim to the public as to the exclusive right to use a mark" (Vienneau 2008). However, there is a formal process for registering trademarks and service marks with the U.S. federal government if additional protection is desired. An example of a software trademark is the Capability Maturity Model Integration (CMMI®), where the ® symbol identifies it as a U.S. registered trademark.

References

Futrell, Robert T., Donald F. Shafer, and Linda Isabell Shafer. 2002. *Quality Software Project Management*. Upper Saddle River, NJ: Prentice-Hall PTR.

Kappos, David. 2012. "An Examination of Software Patents." Keynote Address to the Center for American Progress, November 20, 2012. (Available at http://www.uspto.gov/about-us/news-updates/examination-software-patents.)

Vienneau, Robert L., and Milton Johns. 2008. "Introduction to U.S. Intellectual Property (IP) Law," *SoftwareTech News, The Data & Acquisition Center for Software (DACS)* 11, no. 2 (August). (Available at www.softwaretechnews.com.)

Westfall, Linda, 2017, The Certified Software Quality Engineering Handbook, ASQ Quality, Press, Milwaukee, WI, 2017.

Author

Linda Westfall is the president of The Westfall Team, which provides software engineering, quality, and project/risk management training and consulting services. Linda has more than 45 years of experience in the software industry. Linda is the author of The Certified Software Quality Engineer Handbook 2nd Edition. Linda's hobbies include building fireworks, and she is a Pyrotechnics Guild International Grand Master.