Charleston Southern University

The Ethics of Software

Should the Future be Open or Closed Source?

Taylor Singleton

CSCI 332

Dr. Lin

29 November, 2019

An increasingly divisive debate in modern media, intellectual property rights when it comes to computer software has become an issue that must be solved. On one hand, advocates for open source software argue that the public has the right to all information. Scientists and mathematicians are outraged that certain mathematical formulas and algorithms can be locked behind a copyrighted “wall”, potentially setting science back. On the other hand, companies want to make as much money as possible by holding a monopoly on their software, and by extension, algorithms used in the software. Without license fees and sales, companies and programmers have less incentive to meet deadlines and produce high quality software. In order to understand the debate between open source software and copyrighting algorithms and reach a conclusion on how to satisfy both sides, we must understand copyrights, trade secrets, and patents.

Freibrun Law defines a patent as “a twenty year exclusive monopoly on the right to make, use and sell a qualifying invention” (Freibrun). The invention in question must be determined by the patent office to be new and “nonobvious”. To be considered “nonobvious”, an invention must obviously advance technology in that specific area (Freibrun). When it comes to software, “patents can be obtained for ideas, systems, methods, algorithms, and functions embodied in a software product: editing functions, user-interface features, compiling techniques, operating system techniques, program algorithms, menu arrangements, display presentations or arrangements, and program language translation methods” (Freibrun). Patents are exclusive, meaning no one is legally allowed to use any portion of the software covered behind the patent. This effectively locks portions of mathematics and science behind twenty-year walls.

The next form of invention protection, copyrights do not protect ideas like patents do. Copyrights protect the specific way in which an idea is expressed, whereas patents cover all methods in which an idea may be used (Freibrun). If a software is copyrighted, only the owner has the rights to copy, modify, duplicate, or sell the product in any way. The final type of invention protection is trade secrets. Trade secrets are formulas, processes, or other ways of making items that can be kept secret from rivals and the public. Examples of trade secrets are Coca-Cola’s formula or Colonel Sander’s herbs and spices. In software, trade secrets can be difficult to maintain since most software can be reverse engineered (Freibrun). However, the intellectual property right can be upheld if the developer can prove that the secret was not known to the public.

Open source software and closed source software both have advantages and disadvantages. Firstly, open source software is generally free, making it more budget friendly for the average person. However, closed source software is typically serviced far longer than open source programs are (Ubilla). Closed source software is usually more user-friendly, as customer happiness is vital to their business. On the other hand, open source tends to “throw you to the wolves” when it comes to using the program. Typically, open source software does not have a beautiful user interface designed to ease use. Finally, closed source software is generally far more secure than open source software (Ubilla). Much like with usability, security is not the concern of the open source programmer. To sum up, open source software is free and easy to access, but close source software is more dependable down the road and easier to use.

In order to strike a balance between protecting the rights of developers to sell their products and protecting the public’s right to information, developers should be allowed to hold copyrights on their software. For instance, should a developer produce an app that finds items in a list using binary search, the idea of finding an item in an ordered list by evaluating the value of the item versus the value of the midpoint of the list should not be locked behind a patent, as binary search is a widespread search algorithm. It would be ridiculous to allow someone to hold a copyright on binary search. In the same way, algorithms themselves should not be patentable. The expression of these algorithms should be copyrightable, however. This way, while the algorithm would still be available to the public, using that algorithm for that specific job would be copyrighted, and the developer would be able to make money from people buying the only software using the algorithm for that specific purpose. While no ethical issue has a perfect answer, in this case, I feel that this answer is the best answer in light of finding a compromise to satisfy both sides of the dilemma.

Works Cited

Freibrun, Eric. “Intellectual Property Rights in Software – What They Are and How to Protect Them.” Freibrun Law, https://freibrun.com/intellectual-property-rights-software-protect/.

Ubilla, Alma. “Open Source vs. Closed Source Software.” StackPath, https://www.iticollege.edu/open-source-vs-closed-source-software/.