Evan Hill

Professor O'Neill

CSCI 301

19 February 2022

Copyright and Open Source Ethics

As many people know, ethics are extremely important in the field of computing, especially regarding how ethics and the law pertain to copyright. Specifically, programmers copying code from other programmers and using that code in their own work can present a number of legal and ethical issues that pertain to copyright, which we will be examining in this paper.

Firstly, in order to provide an informed ethical opinion on these matters, we must know what the world's various computing organizations deem ethical and unethical. In examining the ACM and IEEE codes of ethics, they are similar in some ways and different in others. The most notable difference of these is that the ACM code of ethics is much longer than the IEEE code of ethics, and as such, the ACM code covers more topics and goes into more detail than the IEEE code. However, the topics and ethical stances that the IEEE code does cover align quite closely with some of the major topics presented in the ACM code, so the two codes are similar in that regard. The major tenets covered in both codes are, as quoted from the IEEE code (because they are similar to the major tenets of the ACM code): " . . . uphold[ing] the highest standards of integrity, responsible behavior, and ethical conduct in professional activities", " . . . treat[ing] all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others", and " . . . striv[ing] to ensure this code is upheld by colleagues and co-workers."

Secondly, we must examine what the United States government states about copyright law and then cross-reference it with a given code of ethics. The U.S. government informs on their webpage about the basics of U.S. copyright law that "Works are original when they are independently created by a human author and have a minimal degree of creativity" and that " . . . copyright protects expression, and never ideas, procedures, methods, systems, processes, concepts, principles, or discoveries". In applying this to programming, we can understand that U.S. copyright would protect the program code itself, but not the ideas or algorithms that make the code work. For example, if a person wrote a program that counted the number of all combinations of any three letters in the English alphabet, then under standard U.S. copyright law another person would not be able to copy any of the raw original code and use it in their own work without permission. However, if that other person wanted to look at the original code and use and expand upon the original counting algorithm idea for their own code, that would be deemed acceptable and fair use under U.S. copyright law. In looking at the ACM code of ethics, the latter outcome of this example where the second coder uses the algorithm idea for his/her own code would be deemed ethical because it is not in violation of any law and respects the copyright of the original coder. However, the ACM code of ethics also states that "Computing professionals should . . . credit the creators of ideas, inventions, work, and artifacts", which would mean that in order for the second coder to fully comply with the ACM code of ethics, he/she would have to attribute the original idea for the counting algorithm to the original programmer somewhere in the code itself and/or in the execution of the code. Lastly, my ethical opinion on this matter would align with the ACM code of ethics in this example scenario.

Additionally, there is also the issue of open-source licensing. According to the Open Source Initiative, open source software is defined primarily as software that can be redistributed for free or for sale without penalty, that includes and allows distribution of the source code, and that allows all derivative works and requires such works to be distributed under the same license as the source software. There are many different open-source licenses available that accommodate a variety of use cases. However, I posit that the MIT license is the best general-purpose license because it is easy to understand and is permissive. Specifically, "permissive" means that software under a permissive license, according to GitHub's online legal resource for open-source licenses, "gives the public permission to use, modify, and share, without any condition for downstream licensing", which allows many people to make derivative works for the benefit of society without many restrictions on the handling and distribution of both the original work and derivative works.

As a programmer, choosing whether to use standard copyright law or an open-source license can become an ethical dilemma. If a programmer wrote a program that could help contribute to the common good of society, should he/she invoke standard copyright law and attempt to monetize it greatly which would result in great profit for the creator but could also potentially cripple its usage base and thus its benefit on society, or should he/she apply an open-source license to make the program freely accessible to all to greatly improve society at the cost of monetary gain for the creator? According to the ACM code of ethics, "The public good should always be an explicit consideration when evaluating tasks associated with research, requirements analysis, design, implementation, testing, validation, deployment, maintenance, retirement, and disposal", and "Both custom and the law recognize that some exceptions to a creator's control of a work are necessary for the public good. . . . Efforts to help others by contributing time and energy to projects that help society illustrate a positive aspect of this principle. Such efforts include free and open source software and work put into the public domain". However, the ACM code of ethics also states that, “Developing new ideas, inventions, creative works, and computing artifacts creates value for society, and those who expend this effort should expect to gain value from their work”. In order to make an informed ethical judgment on what to do in this example predicament, more details about the specific uses of the software, the parties involved, and what is at stake would be required. However, one generalized solution to this problem could be using an open-source license agreement that allows the software to be sold but still allows derivative works to be made. Additionally, the software should be sold at a reasonable price so that it can be more accessible. That way, the creators will still be compensated for their work while still making the software accessible due to its reasonable selling price and open source license. I believe that this solution would not be in conflict with the ACM code of ethics or the definition of open source software given by the Open Source Initiative.

Works Cited

"IEEE Code of Ethics." www.ieee.org/about/corporate/governance/p7-8.html.

“The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” *Code of Ethics*, https://www.acm.org/code-of-ethics.

“The Legal Side of Open Source.” *Open Source Guides*, https://opensource.guide/legal/.

"The Open Source Definition." *News | Open Source Initiative*, opensource.org/osd.

Office, U.S. Copyright. “What Is Copyright?” *What Is Copyright? | U.S. Copyright Office*, https://www.copyright.gov/what-is-copyright/.