ETHICAL DECISION MAKING AND COPYING SOFTWARE: THE IMPACT OF THE IDEOLOGY OF COMPUTER SOCIALISTS

Matt McGowan
Foster College of Business Administration
Bradley University
mmcgowan@bradley.edu

Paul Stephens
Foster College of Business Administration
Bradley University
prs@bradley.edu

ABSTRACT

Ethical decision-making regarding copying software is only partially understood. This article proposes that a relationship exists between an individual's computer proficiency and his or her ethical beliefs about copying software. We identify two computer subcultures that influence computer users' ethical beliefs. Ideas from these subcultures trickle out to society at large and are misunderstood and misapplied. Information systems students adopt a simplistic view of the subculture ideology. Academic institutions contribute to the problem through a lack of guidance, use of open source systems without explanation and adoption of loose academic licenses. The paper discusses the nature of moral reasoning regarding intellectual property rights and computer software. We discuss a method that can be used to test our proposition.

Keywords: Computer Ethics, Open Source, Free Software, Intellectual Property

INTRODUCTION

Computer piracy continues to plague the United States. The Business Software Alliance (BSA) estimates that over 1.8 billion dollars of lost software sales resulted from software piracy in 2001. [2] The study asserts that there was a negative impact on the U. S. economy and a loss of jobs due to the piracy. Researchers have had limited success in explaining the ethical decision-making that leads to software piracy. [3], [13] We develop a proposition that people's ethical beliefs about software piracy are influenced by their exposure to information technology. In the first part of the paper, we discuss the nature of computer software and intellectual property (IP). Secondly, we identify two subcultures of computer users that believe computer software developers should not have ownership rights to the software they develop, and we present the moral arguments for and against intellectual property rights. In the third part of the paper we develop a proposition based on our literature review. We conclude by presenting a framework for testing our proposition.

COMPUTER SOFTWARE AND INTELLECTUAL PROPERTY

The United States legal system offers three forms of protection for software: (1) copyright, (2) trade secrecy, and (3) patent. [9] These laws provide a mechanism for creators of software to protect their intellectual property. Without this protection, there is a significant risk to developers that they may not recover the costs incurred in creating the software. The ability to make a profit from their creations provides an incentive for software developers to create or improve software products. If inventors cannot sustain their profession, then society loses as well because talented individuals are driven out of the market by financial losses.

Some people have argued that the protection offered by the legal system is too broad, and may actually impede software development. [9] Johnson cites the case of Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc. Whelan developed software for Jaslow's dental office. Several years later, Jaslow decided there would be a market for such software. He studied Whelan's program and proceeded to develop a personal computer-based program in another language. The court ruled that Jaslow violated copyright laws because his program used the same "structure, sequence, and organization" as Whelan's program. Lotus Corporation similarly successfully sued Borland because its spreadsheet software had the same "look and feel" as the Lotus spreadsheet software, even though the code to generate the menu systems was vastly different. Johnson mentions another case that the court ruled did not violate copyright laws because the similarities of the two programs could be explained by the fact that the companies' each designed software to meet the information needs of the same market.

The legal rulings regarding software protection may seem to confuse some people. Logsdon, Thompson, and Reid speculated that there is a relationship between a person's level of moral judgment and one's likelihood of illegally copying software, but found no relationship. [13] They speculated that no relationship was found because people perceive copying software as an issue of low moral intensity, or of low importance. The issue of intellectual property rights for software may not be well understood by people. When physical property is stolen, the owner of that property no longer has use of that property. When software is copied, the original remains intact. Physical property is visible. People can assess the value of the materials and craftsmanship for physical goods. The effort to create good software may not be well understood, even among people who have a high level of moral judgment.

A DISTINCT SUBCULTURE – COMPUTER SOCIALISTS

There is a computer subculture that represents ideologies that conflict with the ethics of intellectual property rights. The subculture, which we call the computer socialists, consists of two subgroups: open source advocates and free software ideologs.

Open Source Advocates

Open source advocates are a recognized internet-based community of programmers that exhibit specific cultural norms. [7] Groups like the Open Source Initiative (OSI - www.opensource.org), the Open Source Research Community (opensource.mit.edu) and Research on Innovation (ROI - researchoninnovation.org/home.htm) promote the open source philosophy that all software

should be developed in an environment where programmers can read, redistribute and modify the source code. The OSI website states:

> The Open Source Initiative does not have a position on whether ideas can be owned, whether patents are good or bad, or any of the related controversies. We think the economic self-interest arguments for open source are strong enough that nobody needs to go on any moral crusades about it. [14]

Obviously, the OSI does have a position on intellectual property protection. Fortunately, advocates of open source don't actively suggest that individuals should ignore intellectual property rights. On the other hand, they aren't taking up any space at their websites promoting a message that people should respect IP rights. The open source advocates feel that it the current system has become untenable, that it impedes the rapid development of quality software.

Open source advocates believe strongly that there are both quality and economic reasons for developing software their way. Eric Raymond, one of the most vocal champions of open source and president of the OSI board of directors, has written extensively on how open source software is developed. [17] The Open Source Research Community provides an outlet for formal research in open source methods. There are currently over 200 members listed in the directory of researchers, many who have published their work in the community. Members of the community and others have conducted research to test the arguments put forth by open source advocates in an attempt to explain why a wide variety of useful and reliable software can emerge from the open source model. [5], [8], [10], [12] Open source advocates offer some compelling arguments as to how the current intellectual property environment impacts the development of information technology. [4], [15], [16]

Free Software Ideologs

Free software ideologs believe that individuals or corporations should not own or control computer software. Groups like the Free Software Foundation (FSF), the Foundation for a Free Information Infrastructure (FFII) (www.ffii.org/index.en.html), and the League for Programming Freedom (LPF) (lpf.ai.mit.edu) represent the ideals of the software ideologs. In this spirit, members of the subculture create intellectual goods and share them freely with the general population. The free exchange of ideas and technology improves the technology and creates de facto standards because of widespread adoption by the general populace. Most would consider this a positive aspect of the free software philosophy. Additionally, the free software ideologs find fault with the current U.S. system of granting patents for software. They argue that granting patents for particular programming techniques has stifled program development. The ideologs argument is that this type of restrictive patenting is absurd for two reasons: (1) many of these techniques are obvious to the point that the original writers of the code wouldn't bother to publish such techniques, yet patents are given for such techniques and (2) it is impossible for the patent office to determine if such techniques have already been produced by others. A program could contain millions of lines of code and use various techniques. It is not reasonable for any person or institution to determine if a routine made up of six lines of code has already been

produced in another program. Additionally, patent examiners are unprepared to determine if a technique is pre-existing. LPF states: "Patent examiners are often ill-prepared to evaluate software patent applications to determine if they represent techniques that are widely known or obvious--both of which are grounds for rejection." [11]

There is a significant difference between arguing that current intellectual property law should be changed and advocating the plunder of proprietary software that is legally protected. Some of the free software ideologs simply dismiss the idea of intellectual property rights in regard to software. This is most clearly demonstrated by the Free Software Foundation (FSF). The FSF states, "… the term ``intellectual property'' is an invitation to simplistic thinking. …any opinion about ``intellectual property'' is almost surely foolish." [6] The FSF would like everyone to think about issues such as software piracy and copyright protection in a different way. For example the FSF views piracy as follows,

> Publishers often refer to prohibited copying as ``piracy.'' In this way, they imply that illegal copying is ethically equivalent to attacking ships on the high seas, kidnapping and murdering the people on them. If you don't believe that illegal copying is just like kidnapping and murder, you might prefer not to use the word ``piracy'' to describe it. Neutral terms such as ``prohibited copying'' or ``unauthorized copying'' are available for use instead. Some of us might even prefer to use a positive term such as ``sharing information with your neighbor." [6]

Clearly, some of the free software ideologs not only have contempt for intellectual property protection but support actively breaking the laws protecting the owners of software. Followers of this philosophy will have few ethical qualms about software pirating and the development of techniques that allow them to access technological products and services for free that are intended to be sold to the public.

MORAL REASONING AND INTELLECTUAL PROPERTY

The philosophical basis for intellectual property comes primarily from the rights and consequentialist frameworks. [9] The rights framework suggests that an individual's labor in developing software gives the person property rights to that creation. Johnson suggests this right is a social right because if someone were to take a copy of the software, the developer still has a copy for his or her use. The primary argument against this position is that one cannot take ownership of mental operations. [9] For example, one could not own the process of multiplication.

The consequentialist framework for intellectual property considers the good and bad consequences of permitting (or not) ownership of intellectual property. One of the primary arguments in favor of ownership is that an individual would not have incentive to invest the resources to develop computer software if one could not profit from it. Opponents of this position argue that there would still be incentives to develop software, for example to sell hardware. [9] Open source advocates suggest that the incentive for developing software is for the authors of the software to receive recognition for their work. The computer socialists argue that ownership stifles development because marginal ideas are protected and users are prevented

from improving these ideas. The system of laws in the United States gives creators legal rights to software. The legal system does not prevent open source development efforts. Ignoring the law, or more specifically, ignoring the rights of software developers does more harm than good to society.

IDEOLOGICAL TRICKLEDOWN

In this paper we argue that the rhetoric that drives the computer socialists trickles out to society at large but in a way that it is both misunderstood and misapplied. In fact, some members of the subculture are simply wrong to promote the breach of IP protection. On the other hand, some members of the subculture take a measured approach where they promote the ideas of free and open software, avoiding any discussion of IP.

The motivation behind this research comes from observations we have made about our management information systems (MIS) students. Some students have a cavalier attitude about intellectual property protection. The copying and distribution of protected software seems to be common among the students. When questioned about their actions a typical response might be "It's just for my use" accompanied by a wink and a smile.

We have observed that as people become more involved with information technology (IT) they are more likely to be exposed to the ideas of the computer socialists. As we have discussed, there are sectors of the subculture that take a pragmatic approach to their views and others that sanction breaking intellectual property law. These two approaches don't reconcile well with each other when one is peripherally exposed to the underlying philosophies of the subculture. Students observe the practice of copying software without understanding the motivation for the behavior. As a consequence, those who are new to IT (students) are sent a rather simplistic message about the ideology. The message is that it is acceptable to copy proprietary software not because they believe that proprietary software and software patents are unethical. Students observe the behaviors of their peers that they respect and copy those behaviors. [1] So, behaviors that many consider socially unacceptable are becoming more acceptable to the students of IT. IT educators have failed to expose their students to the views of computer socialists and proponents of intellectual property and have let the simplistic message dominate. Other factors contribute to the problem. Institutions use free/open source software such as Linux and Apache without informing students about the differences between this type of software and proprietary software. Another contributing factor is the academic license. These licenses have liberal copying provisions which conflict with the student's prior experience with the same software in different environments.

Proposition 1: Those individuals who are more proficient with computer technology are more likely to copy proprietary software.

EXTENSIONS OF THIS RESEARCH

The next step to follow this research is to test the propositions that we have put forth in this paper. In this section we discuss how we might turn our proposition into a testable hypotheses.

The purpose of the future research is to discover whether our anecdotal observations are supported by data. We believe that students who have developed higher levels of computer proficiency will exhibit lower levels of reasoning when applied to certain moral questions surrounding computer technology. We will use computer self-efficacy to measure computer proficiency. Specifically, we believe that we will find that those questions involving intellectual property will be impacted by the subculture ideologies that have developed into more simplistic beliefs held by students with higher levels of computer self-efficacy.

Researchers have identified these moral issues and many frameworks exist in which we can test the level of an individual's moral development. Additionally, research also provides us with the tools to measure individual computer self-efficacy. In this research, we will use those tools to measure individual computer self-efficacy and moral development. We will then compare the moral reasoning of individuals with high computer self-efficacy with those individuals with low computer self-efficacy.

Finally, we will address the significance of our findings and how the research can be extended to test whether factors such as age, gender and profession disaggregate into groups holding differing levels of moral reasoning with regard to the use of computer technology.

REFERENCES

1. Al-Jabri, I., and Abdul-Gader, A. (1997) "Software Copyright Infrigements: an Exploratory Study of the Effects of Individual and Peer Beliefs" International Journal of Management Science, (25:3), pp. 335-344.
2. Business Software Alliance. (2002). *2001 U. S. Software State Piracy Study.* Retrieved March 22, 2003, from http://www.bsa.org.
3. Chong, James Y. L., and Yap, Chee-Sing. (1998). "Testing an ethical decision-making theory: The case of softlifting," Journal of Management Information Systems (15:1), pp. 213-237.
4. Faldetta, G. (2002). "The Concept of Freedom in Resources: The Open Source Model," Journal of Business Ethics, (39:1/2), pp. 179 – 188.
5. Fitzgerald, B. and Feller, J. (2002). "A further investigation of open source software: community, co-ordination, code quality and security issues," Information Systems Journal, (12), pp. 3-5.
6. Free Software Foundation. (2003). *Confusing Words and Phrases that are Worth Avoiding.* Retrieved March 14, 2003, from http://www.gnu.org/philosophy/words-to-avoid.html.
7. Hars, A. and Ou, S. (2002). "Working for Free? Motivations for Participating in Open Source Projects," International Journal of Electronic Commerce, (6:3), pp. 25-39.
8. Hipple, E. (2001). "Innovation by User Communities: Learning from Open-Source Software," MIT Sloan Management Review, (42:4), pp. 552 – 568.
9. Johnson, Deborah G. (1994). *Computer Ethics, Second Edition*, Upper Saddle River, NJ: Prentice-Hall, Inc.
10. Johnson, J. (2002). "Open Source Software: Private Provision of a Public Good," Journal of Economics and Management Strategy, (11:4), pp. 637-662.

11. League for Programming Freedom. (1994, April 29). *Against Software Patents.* Retrieved March 21, 2003, from http://lpf.ai.mit.edu/Patents/against-software-patents.html.
12. Lerner, J. and Tirole, J. (2002). "Some Simple Economics of Open Source," The Journal of Industrial Economics, (L:2), pp. 197 – 234.
13. Logsdon, Jeanne M., Thompson, Judith Kenner, and Reid, Richard A. (1994). "Software piracy: Is it related to level of moral judgment?" Journal of Business Ethics (13), pp. 849-857.
14. Open Source Initiative. (2003). *FAQ:Advocacy.* Retrieved March 14, 2003, from http://www.opensource.org/advocacy/faq.php.
15. Osorio, C. (2002). *A Contribution to the Understanding of Illegal Copying of Software: Empirical and analytical evidence against conventional wisdom.* Retrieved March 14, 2003 from http://opensource.mit.edu/online_papers.php.
16. Pfaffenberger, B. (1999). "Open Source Software and Software Patents: A Constitutional Perspective," Knowledge, Technology and Policy, (12:3), pp. 94 –112.
17. Raymond, E. (1999). "The Cathedral and the Bazaar," Knowledge, Technology and Policy, (12:3), pp. 23-49.