The Open Source Community and Piracy

Bachelor Thesis Information Science

Iris Trepels
s0712574
Radboud University Nijmegen

Supervisor: Luca Consoli

27 June 2011

Abstract

This thesis attempts to find a relationship between the open source community and software piracy. The reason for believing there is such a relationship is the fact that the open source community advocates that software should be redistributed freely, and is therefore also opposed to software patents. Literature study and interviews have researched the relationship between the open source community and piracy, and if this relationship can be established through the community's dislike of software patents. It has been concluded that software patents are not connected to this relationship. Interviews have shown that the open source community's attitude towards software piracy is negative, and that this attitude may be more negative than the general opinion towards software piracy. This is concluded to be the relationship between the open source community and software piracy.

Contents

Chapter 1

Introduction

On June 28, 2010 in Bilski v. Kappos the United States Supreme Court rejected the patent application submitted by Bilski et. al. The patent was requested for a "business method", and was rejected on grounds of the process being too abstract. 35 U.S.C. 101 states that "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title." According to [19] the Supreme Court precedents provide three exceptions to this paragraph, namely laws of nature, physical phenomena, and abstract ideas. These matters are not eligible to be patented. According to [12] the subject to be patented was "a routine that provides insurance against jumps in energy prices due to weather fluctuations". [12] also states that "If the claims only consisted of a few linear equations that could be solved on the back of a napkin, one could argue that the Court is ruling only on an attempt to patent pure math. But the claims are about statistical number crunching so extensive that the author needs computational shortcuts to get up to n=3. The application is not in the language of do-while loops and function calls, but the story is clear nonetheless: Bilski is describing software." The decision of the court was based on the decision that an algorithm is an abstraction and for that reason cannot be patented.

Even though the patent application was rejected the decision was seen as a disappointment by the open source community. The reason for this is that the opinion stated that business processes are not categorically ineligible to be patented. For the open source community, which had hoped this case would set a precedent against software patents, this is a disappointing outcome. According to [7] patents are a threat to open source software. According to the open source definition [7] software (source code and compiled form) should be available to be freely redistributed.

Given the fact that the open source community is against software patents (among others) because they threaten innovation, and they believe that software should be redistributed freely, this poses the question if the open source community thinks this should also be applied to commercial software. Taking this a step further would be to assume that the open source community considers piracy of patented commercial software to be morally just. The research question this poses is “What is the relationship between the open source community and software piracy?”

To research this the hypothesis will be the reasoning in the previous paragraph. It will be attempted to connect the open source community to piracy directly and through it’s opposition to software patents. To this end four sub questions have been formulated.

1. What are the differences between open source and software piracy?
2. What effects do software patents have on software piracy?
3. Why is the open source community against software patents?
4. What does the open source community think of software piracy?

The goal of the research is to determine the relationship between the open source community and software piracy. In order to answer this question two routes will be taken. The first will be to establish a link between open source and software piracy through the open source community’s attitude on software patents. The dislike of software patents by the open source community may turn out to be the link between open source and software piracy. One of the questions that needs to be answered in order to establish this link is sub question 3: “why is the open source community against software patents?”. This may link open source to software patents. The question that will link patents to piracy is sub question 2: “what effects do software patents have on software piracy?”

Another approach that will be used to answer the main question will be to establish a direct link between open source and software piracy. First the relationship between open source and software piracy will be researched. How do these phenomena differ, and how are they alike? This will be done by answering sub question 1: “What are the differences between open source and software piracy?”. The opinion of the open source community on software piracy will be researched by answering sub question 4: “What does the open source community think of software piracy?”. More information on the method used for this research can be found in chapter 3 of this thesis.

Chapter 2

Justification

According to [4] the global piracy ratio in 2009 was 43%, in 2008 this was 41%. The value of the pirated software for 2009 was $51.4 billion. This is obviously a big problem for companies that develop commercial software because of the income they do not receive due to piracy. The open source community however, is in favor of free redistribution of software. This is one of the criteria for a piece of software that is distributed under an open source license [7]. They also develop open source versions of commercial software such as Open Office. These pieces of software are available under an open source licence, and theoretically make pirating the commercial versions of this software unnecessary. When there is a patent on software this means that the concept of the software cannot be used without consent of the patent holder. It is for this reason (among others) that the open source community wants software patents to come to an end. Another reason brought forward by the community is that patents block innovation because the concept or idea behind the software cannot be used in other products or inventions and it cannot be expanded or improved without consent of the patent holder. Innovation is important for the scientific community, and when certain methods cannot be used because they are patented this will be held back. [11] and [16] have researched (software) patents and the effects they have on innovation. [16] concludes that patents threaten technological advance by not allowing for creation and use of so called “design spaces”. This means that the pool of knowledge that is being created is not shared, thus others cannot use this knowledge for further research and innovation, and thereby hindering technological advance. [11] points out a negative relationship between the amount of software patents and R&D intensity, meaning that if more software patents were acquired in a certain industy, R&D became of lesser focus. [11] also states that when software patents became less expensive to obtain it became more attractive for companies to shift their competitive strategy from R&D to strategic patenting.

Chapter 3

Method

3.1 Introduction

To introduce this chapter an overview of the research elements of this thesis will be presented. In the following chapters the results of the literature that has been studied will be presented. The following topics have been researched:

1. Open Source
2. Piracy
    (a) Attitudes
    (b) Differences with open source
3. Patents
    (a) Effect on piracy
    (b) Arguments against patents

As stated in the introduction chapter of this thesis the opinion of the open source community on piracy will also be researched. Because there is no literature available to answer this question, interviews will be conducted to answer this question.

3.2 Open source – patents – piracy

This section will describe the approach that will be taken in answering sub questions 2 and 3. These questions are used to establish a link between the open source community and software piracy through the open source community's opinion on software patents. First is sub question 3 "Why is the open source community against software patents?" The variable for this question is the open source community's opinion on software patents. An indicator for this variable will be articles and other documentation written by the open source community opposing software patents. The five most popular opinions, ranked according to the amount of literature that is to be found corresponding to that opinion, will be listed. Another indicator will be the opinion of the interview subjects, they will be asked about their opinion on software patents.

Subquestion 2 asks "What effects do software patents have on software piracy?" Variables for this subquestion are software patents and piracy. Indicators for these variables will be articles researching the influence software patents have on piracy. [15] will mainly be used to answer this question. The data collected will be made into a coherent whole and will be used to form answers for the question posed. If the literature does not provide an unambiguous answer all views will be presented.

3.3 Open source – piracy

In this section the method that will be used to establish the link between open source and software piracy will be explained. Sub question 1 "What are the differences between open source and software piracy?" has the variables open source and piracy, and the indicators that go with these variables are the definition of open source and the definition of software piracy. Using both definitions differences between these two elements can be identified. Sub question 4 "What does the open source community think of software piracy?" is the second question. The variable for this question is the opinion of the open source community on piracy. Indicators for this variable are:

- Belief of the person that all software should be open.
- Belief of the person that there should be no software patents.
- Belief that less patents would lead to less piracy.
- Belief of the person that it is okay to pirate software.
- Belief of the person that it is okay to pirate patented software.
- Piracy behavior of the person.

3.4 Interviews

As mentioned above interviews will be held to answer sub questions 3 and 4. The indicators mentioned in the previous section will be incorporated into the interview guide that will be used as the guide through the semi-structured interviews that will be held. This interview guide can be found in appendix A. The result of these interviews will not be a precise view of what the open source community thinks of piracy, but it will give a good idea of what this opinion is. The fact that the outcome of the interviews cannot be a precise view of what the whole open source community thinks is due two factors. The actual sample size is not known yet, but will consist of 6 people or less due to time constraints. As the interviews progress it will become known when the number of required participants is reached. This will be when new explanations and ideas/data stop emerging from the interviews (data saturation). Also, it will be difficult to find a representative sample due to the fact that participants must meet the criteria set for the target group, which will be descibed further on in this chapter. Therefore participants will be sought for in the direct environment, and through "snowball" sampling techniques.

A question that can be asked is why conduct interviews instead of, for example, a survey. This leads to the question if qualitative research or quantitative research would be more suitable for this thesis. Quantitative research allows for a very precise measurement of the data. It is suited when a researcher knows precisely which information is sought, and what questions need to be asked. This will result in a set of data that can be statistically analyzed. Qualitative research can yield stories and anekdotes from the research subjects. This form of research is more suitable when the area that is being researched has not been researched (much). Therefore it can be difficult to know exactly what to look for. When doing qualitative research this will become more clear as the research progresses. The data from qualitative research cannot be statistically analyzed and is therefore analyzed by the researcher by, for example, coding. The choice of doing interviews (qualitative research) as opposed to a survey (quantitative research) is that this research question has not been researched before. Because of this it may well be that not all relevant questions are known beforehand. New insights may be introduced by the research subjects, and when doing interviews it is possible to ask further questions about these new insights. Because these interviews seek an in depth opinion the interviews will take between 1 and 1,5 hours. The interview subjects will be given as much freedom as possible to give their opinion, while staying on topic. Therefore it is very much possible that new information or opinions also come to light regarding the literature research that will be presented in the following chapters. All (relevant) information that comes forth from the interviews will be presented in chapter 7.

3.4.1 Target group

For this research the target group will be the open source community, more precisely, open source developers. The developers will be targeted because they actually contribute to the community so they share the “open source mentality”. [13] poses some interesting constraints when selecting individuals that will be taken into account for this research. They selected their sample from “individuals listed as official developers on F/OSS projects hosted on the SourceForge.net F/OSS community Web site” [13]. F/OSS standing for Free/Open Source Software. The two elements [13] considered when selecting individuals were (1) project development status (Planning, Pre-Alpha, Alpha, Beta, Production/Stable, Mature) and (2) number of developers on a project. Projects that had the Planning or Pre-Alpha status were not considered because they typically do not contain source code. Projects that had only one developer were not considered to ensure that the projects required coordination with other members [13]. While [13] conducted online surveys and this research will be conducting interviews these considerations are still valid in selecting interview subjects. First of all because this is research into the open source community, therefore some interaction with the community is required from the individuals. Second it is important to be sure that the individuals actually contribute to the community so it is important to have individuals from projects that are well into the development chain. Therefore these two requirements will be used when selecting participants for the interviews.

The regulations regarding software patents in Europe are a bit different from the regulations in the United States. According to article 52 of the European Patent Convention a computer program “as such” is not patentable. Criteria for patentability are “they have to be new, represent a non-obvious technical solution to a technical problem, and be susceptible of industrial application.” [20] So while you cannot patent a computer program, you can patent a solution to a technical problem which is implemented as a computer program. However, software that solves a business problem is not eligable for patenting because it is not a solution to a technical problem. However [8] states that patentability in the United States and Europe is nearly uniform and that Europe is moving in the direction of the United States on this issue. Given is that in both the United States and Europe there is loud protest from the open source community, it can be concluded that both sets of regulations are unsatisfactory regarding software patents. Therefore it will be assumed that this can also be considered an issue for European (more specifically: Dutch) open source software developers. This point is made as the case that is referred to in the introduction chapter took place in the United States of America, while the interviews that will be conducted for this research will take place in the Netherlands.

3.5 Conclusions

This section will explain how the four sub questions will lead to an answer to the main question. The goal of the sub questions is to make a connection between the open source community and software piracy. Sub questions 1 and 4 will try to make this link directly, and sub questions 2 and 3 will try to make this link through the open source community's opinion on software patents.

The open source community is openly against software patents because it is in favor of freely redistributing software. The question is if this means that the open source community thinks that because software should be redistributed freely that it is okay to pirate software. To this end sub questions 2 and 3 will attempt to establish a link between the open source community and software piracy through patents. The hypothesis is that if there are less software patents there would be less piracy of software. Sub question 2 will try to confirm or reject that hypothesis. With the answer to that question we can say if this could have an effect on the view of the open source community towards software piracy. Sub question 3 will give a list of reasons why the open source community is against software patents. If any of these reasons have to do with software piracy a relationship between the open source community can be established. If the answers to neither of these questions lead to a concrete link between open source and software piracy then it will be concluded that there is no relationship between open source and software piracy through software patents. If question 2 confirms that more software patents cause more software piracy it cannot be directly concluded that this ties the open source community to software piracy. The data from sub question 4 will still be needed to confirm this.

With the answer to sub question 1 it will be established (from a theoretical standpoint) if the hypothesis that there is a likeness between open source and software piracy is valid. If there are fundamental differences between the ideology of open source and the definition of software piracy then it can be concluded that open source and software piracy are not theoretically linked to each other. It will be interesting to see if this matches the results of sub question 4. Sub question four will establish the opinion of the open source community on software piracy. Interviews will be used to gain an idea of the attitude of open source developers towards software piracy.

Chapter 4

Open Source

4.1 How it came about

The GNU project was started in 1984 by Richard Stallman. He believes that the knowledge behind a computer program, source code, should be free. In contrast to commercial software vendors who see bits of software as trade secrets that should be protected, Stallman sees it as scientific knowledge that should be shared and distributed. To Stallman source code is necessary for innovating in the computer science world and it should therefore be distributed freely. However he thinks it should not become of the public domain like other scientific research because businesses would then use the software for financial gain. That is why he created the GNU General Public License (GPL). This license forbids people from charging others for the software, and states that derived and modified works must also be licensed under GPL. Software companies rejected free software. In 1997 leaders of the free software community assembled to discuss ways to promote the idea of free software. This was necessary due to the fact that the anti-business message of free software was keeping people from embracing free software. A new term to describe the software they were promoting came about: open source. Guidelines to describe open source software were created and brought together in The Open Source Definition. This definition allows open source and proprietary software to be combined. Derived or modified software do not have to be licensed under an open source license according to the definition. [7]

4.2 Ideology

In [7] Bruce Perens calls the Open Source Definition a “bill of rights” for computer users. He states that users of software should not accept the restrictive licences of other software that prevent them from for example installing the software on a second computer. Software can become unusable to a computer user when he buys a new computer, installs a new operating system, has to pay for upgrades, or if the company that maintains the software ceases to exist. Under these licences it is also impossible to add to, or improve software to suit the user's specific needs, or fix the software when it has bugs. This lead to the following rights for people using open source software:

- The right to make copies of the program, and distribute those copies.
- The right to have access to the software's source code, a necessary preliminary before you can change it.
- The right to make improvements to the program. [7]

4.3 Licences

The name "open source" software tends to lead people to think that the software is of the public domain. However this is not the case. The software is covered by copyright and an open source licence. The Open Source Definition, Version 1.0 [7] lists the criteria that a licence must comply with to be considered an open source licence. Examples of these licences are the GNU General Public Licence, and the Berkeley Software Distribution (BSD) licence. Some of the criteria that an open source licence must comply with are that the software must be redistributed freely, the distribution must include the source code, and the licence must allow modifications and derived works of the program.

4.4 Business method

Open source is often seen as giving the software away. However one of the goals of the open source community is to show businesses that open source can be a successful business strategy, and that businesses using an open source strategy can be/are profitable. Examples of companies using open source as a strategy are Netscape, IBM, SUN Microsystems and Apache.

Netscape is one of the companies that uses open source as a strategy. They made their Navigator web browser freely available on the internet. At that time many wondered how they could make money by giving the browser away. When Netscape released the source code of one of their browsers it started receiving valuable input from free-software developers. Now many see this as the strategy that caused Netscape to grow the way it did [10]. Red Hat Software, which distributes Linux (an open source operating system), has received money from Intel and Netscape. This is due to the fact that Intel and Netscape have interests in the business of Red Hat. Netscape produces software that runs on Linux, and Intel manufactures processors on which Linux can run [10].

It is stated by [7] that choosing the right business model is essential for businesses looking to make a profit with open source software. The following business models are given by [10]:

- “Support Sellers: Revenue comes from media distribution, branding, training, consulting, custom development, and post-sales support.
- Loss Leader: A no-charge open-source product is used as a loss leader for traditional commercial software.
- Widget Frosting: Companies in business primarily to sell hardware use the open-source model for enabling software such as driver and interface code.
- Accessorizing: A company distributes books, computer hardware, and other physical items associated with and supportive of open-source software.
- Service Enabler: Open-source software is created and distributed primarily to support access to revenue-generating online services.
- Brand Licensing: One company charges other companies for the right to use its brand names and trademarks in creating derivative products.
- Sell It, Free It: A company’s software products start out their product life cycle as traditional commercial products and then are continually converted to open-source products when appropriate.
- Software Franchising: This combines several of the preceding models (in particular Brand Licensing and Support Sellers).A company authorizes others to use its brand names and trademarks in creating associated organizations doing custom software development in particular geographic areas or vertical markets. The company might also supply franchises with training and related services in exchange for franchise fees of some sort. ”

The world of software is always changing, and maintenance, and new versions of software are always necessary. The advantage of open source software is that it does not have to be done by one (group of) person(s). If software is open source there will often be software developers that are willing to fix/report bugs, or create patches or additions for the software. [7]

4.5 Motivations of open source developers

Much research has been done into the motivations of software developers to participate in open source software development. A difference can be made between intrinsic and extrinsic motivators. Intrinsic motivation comes from inside a person, such the enjoyment of performing a task, or a feeling of creativity when performing a task. Extrinsic motivation relies on external factors that motivate people. This can be monetary rewards, promotion, or self marketing. The results of the studies done on this topic are not unambiguous. [13] has found that enjoyment based intrinsic motivation is the most important motivational factor. This refers to a feeling of creativity when working on the project. [13] also states that user need, intellectual stimulation and the improvement of programming skills are important motivators for open source developers. In contrast [9] states that intrinsic motivators do play a role, but extrinsic motivators are key motivators for open source developers. Building human capital, self marketing and the need for the software play the biggest role according to [9].

Chapter 5

Piracy

5.1 Definition

According to [17] the definition for software piracy is “the illegal copying of computer software”. This definition states that software can be illegally copied, which is a point of discussion according to [5]. According to [6] the definition of software piracy is “with piracy, you copy programs from one disk to another or from a disk to a computer”. This however is not a complete definition as it does not consider copying from computer to computer. Also it does not consider the fact that some software licences allow the software to be freely redistributed. The following definition of software piracy from the Business Software Alliance [3] seems to be the most complete definition: “The illegal use and/or distribution of software protected under intellectual property laws. Software piracy may take many forms:

- End-user piracy occurs when an individual or organization reproduces and/or uses unlicensed copies of software for its operations.
- Client-server overuse occurs when the number of users connected to or accessing one server exceed the total number defined in the license agreement.
- Counterfeiting is the illegal duplication of software with the intent of directly imitating the copyrighted product.
- Hard-disk loading occurs when a computer hardware reseller loads unauthorized copies of software onto the machines it sells.
- Online software theft occurs when individuals download unauthorized copies of software from the Internet.
- License misuse occurs when software is distributed in channels outside those allowed by the license, or used in ways restricted by the license.”

It takes into account the many different ways that software can be pirated, and the fact that licences determine the ways that the software should be dealt with.

The global piracy ratio of 2009 was 43%, meaning that 43% of all software is pirated software; this was 41% in 2008. The worth of the pirated software in 2009 was $51.4 billion.

[6] states that if there were no piracy some changes would take place. The sale of software would rise, and the prices of popular software such as Microsoft Office would drop. This would also mean that people living in 3rd world countries who are not able to afford software licenses would not be able to use software anymore. According to [6] software vendors should take this into account and adjust prices per country. It is also stated that vendors would start selling software in countries that are now avoided due to software piracy.

5.2 Attitudes

When researching the opinion of the open source community on piracy it is useful to know what the general attitude toward piracy is for comparison. [1] has researched factors that influence the attitude towards piracy. According to [1] the most important factors are that people think they can save money by pirating, which is related to the belief that digital material is overpriced. Another factor is that people do not fear getting caught when pirating digital material, and they do not see piracy as an important issue. Furthermore [1] has shown that the most important factor is the opinion of others, which according to the study is often positive. It is also shown that people generally feel happy and excited while pirating digital material. Another study done by [14] researched factors that motivate piracy. One element of this research was the intent to pirate software. This intent was shown to be influenced by the perceived consequences and social factors. It is shown that friends are the only social factor that influence intentions regarding piracy. Examples of perceived consequences were "possessing more software, low risk of being penalized, saving time in acquiring software, deteriorating a persons sense of ethics, saving money, and no support for pirated software." [14] What is interesting about this study is that it showed that intentions to pirate showed no significant relationship with the actual piracy behavior. What did significantly affect the actual piracy behavior were conditions that facilitate piracy.

5.3 Relationship with open source

There are some factors that can be compared when trying to determine a relationship between piracy and open source. Such factors are licences, motivation, business methods, and distribution.

Piracy is illegal because (in this case) software is distributed without a valid licence. Proprietary software usually contains a licence (EULA) that must be agreed to by the user before the user can use the software. The implications of such a licence are usually that the copy of the software remains property of the publisher of the software. With such licences the software may usually be installed a limited number of times. Open source software must also be distributed with a licence. This licence is different from the licences of proprietary software because it states that the software may be redistributed freely. The difference between the illegal distribution of software and the distribution of open source software is the validity of the licence. If the distributed software does not have a valid licence this is illegal (piracy), if the distributed software has a valid licence (such as an open source licence) this is not illegal.

Another difference is motivation. Open source software is freely redistributed because of the ideology that is described in the previous chapter. The belief is that computer users should not accept the licences of proprietary software because they are too restrictive. [7] The motivations of people who pirate software are quite different from the motivations listed in [7]. These motivations have more to do with the expense or time it takes to legally acquire a piece of software, or the lack of risks involved in pirating software.

Open source software gives a lot of freedom to the user. The software may even be made into proprietary software or used in proprietary software. It is also very much possible to make money off open source software. There are many methods for doing this, these can be found in the previous chapter. While there are many businesses that make profits using open source software, other businesses complain that they are losing money due to piracy. So while open source provides many opportunities for making money publishing software, piracy only harms companies that publish software.

The similarity between open source and piracy lies in the distribution (disregarding licences). With piracy, software is made available for (free) download as long as someone is distributing it. The same is the case with open source software. This is the theoretical link that ties open source software to piracy.

Chapter 6

Patents

6.1 Patents and copyright

There may be some confusion on the difference between patents and copyright, and how these apply to software. First the definition of these two terms will be given. These definitions are given to illustrate why copyright is not regarded as sufficient for companies wanting to patent their products, and thus why patents are being requested for software products. Both patents and copyright are intellectual property rights, and thus legally protect someone's intellectual or intangible property. A patent grants the property right of an invention to the inventor. This means that the inventor has the right to exclude others from making, using or selling the invention for the time the patent is valid. In the United States this is 20 years [18]. A copyright is a form of protection for authors of original works. According to [18] this includes "literary, dramatic, musical, artistic, and certain other intellectual works, both published and unpublished". The party holding the copyright has the exclusive right to reproduce, distribute, perform, display and produce derivative works of the copyrighted work [18]. The essential difference between the two is that patents protect the idea, and copyrights protect the form of expression. As an example from the software domain Microsoft Word will be used. If the software were patented that means the idea of the word processor could not be used by anyone without the permission of the patent holder (presumably Microsoft). Because there is no patent on Microsoft Word, but there is copyright the software cannot be copied or distributed by anyone except the copyright holder. The idea however can be used because there is no patent on the software. This is why software such as OpenOffice.org Writer can be legally made. This is because it is not a copy of the Microsoft Word (thus not infringing copyright), but it uses the same idea as Microsoft Word (this is allowed because it is not patented).

6.2 Effect of patents on piracy

The study by [15] seems to be the best source of information for determining the relationship between the number of patents and the software piracy rate. This study discusses the software piracy rate and the rate of protection for intellectual property in a large number of countries. It is also attempted to tie these figures to economic, institutional and cultural factors.

The article states that intellectual property rights (among which are patents) are stronger in countries that are classified as high-income countries. [15] It is also stated that the software piracy rate is lower in high-income countries (compared to low-income countries). This would suggest a negative correlation between intellectual property rights and the software piracy rate. However, this relationship is not proven, there may also be other factors that contribute to this relationship. What is also stated in [15] is that countries that have a collectivist culture (versus individualist) have a strong belief that intellectual property is public property. Therefore there is a lower protection for intellectual property in such countries. It is also shown that the software piracy rate is significantly higher in collectivist countries. The other side of this is that individualist countries have a higher protection of intellectual property and a lower software piracy rate. The study by [15] has also shown that countries that strongly protect traditional property and contracts have significantly lower software piracy rates. Seperatly from these figures the study has also compared the measure of patent protection with average piracy rates of 57 countries. The correlation between these two figures was -0.74. [15] This implies a strong negative correlation, meaning that countries with a strong measure of patent protection have low software piracy rates, and vice versa.

Another study has been done that shows a relationship between copyright software protection and software piracy rates in European countries. [2] This study shows a strong negative correlation between copyright software protection and software piracy. The study by [15] shows that when there is a strong measure of patent protection there is less software piracy. The study by [2] does not cover patents, but does deal with copyright protection of software and in a sense confirms that a high protection of intellectual property leads to a lower software piracy rate. Therefore it is concluded that when there are more patents this will lead to less software piracy, and vice versa. Therefore it can be said that there is a negative relationship between patents and piracy.

6.3 Arguments against patents

When searching for arguments against software patents many different arguments can be found. The following five arguments are the ones found most on blogs and websites for and by members of the open source community.

The argument that was found most is the argument that software patents are damaging to small and medium enterprises as well as open source/free software initiatives. This is due to the fact that these enterprises do not have large patent portfolios to defend themselves against litigation by large companies, nor do they have the funds to defend themselves in these lawsuits, which often take a lot of time and are very expensive. [21]

The next argument made is that many patents granted cover trivial "inventions" that any person with sufficient knowledge of the relevant domain at that time could have thought up themselves. To deter these obvious patents more knowledgeable patent examiners would be needed, however people with comprehensive knowledge of computer science will most likely not be working at a patent office. [21]

Another popular argument is the existence of so called "patent trolls", which are companies that do not create software, but they do buy up software patents with the purpose of making (a lot of) money off sueing companies that infringe in their patents. [21]

Terms of patents, which is 20 years from the filing date in both Europe and the United states [8] are another popular issue. The rapid rate of innovation in software means that a piece of software for which a patent application is filed will be either outdated or widely used before the patent is issued. Therefore a creator of software cannot have any certainty that the idea he is using will become patented. [21] Aside from this the average lifespan of software is about 2 years. Therefore the term of 20 years is much too long for software patents.

The final reason in this list is that it is simply very difficult to avoid infringing on patents. [21] One reason for this is that many patents are so trivial that others can easily think of them and incorporate them in their own software unknowingly. Another reason is that standardised methods are often patented and difficult to work around. Also it is difficult, if not impossible, (and expensive) to check every patent to make sure that your software doesn't infringe on it.

Chapter 7

Interview results

In total six interviews were conducted to achieve the results that will be shown in this chapter. The interviews took place between April 20, 2011 and 11 May, 2011. The lengths of the interviews were between 1 and 1,25 hours. These interviews were semi strucured and the interview guide can be found in appendix A. Transcripts have been written for each interview but will not be included in this thesis due to the fact that they contain too many pages. Note that the interviews were conducted in Dutch and the transcripts have been written in Dutch. The results and quotes taken from these transcripts have been translated into English by the author. The transcripts can be obtained by sending a request to the author by e-mail (`Iris.Trepels@student.ru.nl`).

The first section of this chapter will present the results of the interviews that correspond with subquestion 3, why is the open source community against software patents? The second section of this chapter will present the results of the interviews that correspond to subquestion 4, what does the open source community think of software piracy? The anwers to the indicator questions that were created to answer this question will also be presented in this last section.

The results in this chapter were taken from the transcripts. For the selection of relevant information from these transcripts an open source tool for qualitative data analysis called Weft QDA was used. This tool made it possible to select important parts from the transcripts and label these. When finished there was a set of labels, each containing pieces of text from the interview that are relevant for that particular label. A label referencing to, for example, a subquestion or indicator. The texts belonging to the labels were then processed into the next sections of this chapter.

7.1 Why is the open source community against software patents?

The title of this section states the question for which the results will be presented in this section. During the interviews the interviewee was asked what he thought of software patents. As these interviews were semi structured many thoughts were voiced on the subject of software patents. These thoughts have been arranged in the categories “opinion” and “alternatives”.

7.1.1 Opinion

Most of the people that were interviewed were of the opinion that there should be no software patents, at least in the form that they exist today. Interviewee #2 says that *“I think that many software patents in practice are just bullshit and should have never been granted in the first place.”* However this person also states that if someone has a truly good idea that solves a very complex problem such as the “travelling salesman problem” that that idea should be patentable and that the inventor should be able to make a lot of money from this idea and that people using this idea can also make a lot of money off this idea. He says that regardless of the time put into an idea, if it is truly a good idea then it is up to the inventor what is done with it, such as requesting a patent and deciding who gets licences to use that idea. This opinion is unique in for these interviews. The other interviewees were all firmly against software patents. Reasons given for this are the following.

The reason that seems to be the most frequently named is that most software patents are trivial. Interviewee #6 says *“There is a software patent on buying something on the internet, there is a software patent on hyperlink, there is a software patent on buttons that you can click on, rectangular buttons in a user interface. No one takes that seriously, otherwise you wouldn’t be able to make software.”* Interviewee 5 concurrs and says that most software patents are such silly things and so obvious that often they have been in use for a long time. Many examples of trivial software patents are named, such as Microsoft’s patent on page up and page down, and their patent on the right mouse button. A much stated cause for this flaw in the patent system is that patent examiners do not have the expertise to determine if the “invention” is in fact a real invention or if it is trivial. Another objection is the broad scope of many patent applications and patents. Unlike in chemistry or medicine there is not one, or a list of elements without which the idea cannot be executed. A software patent is a description of an idea without an implementation. This idea can be vague to the extent that many different pieces of software (unknowingly) infringe on that patent, and because of the broad spectrum it is very difficult to work around a patent. The broad spectrum does not consider independent invention and the concept that someone invents something because he needs it. As interviewee #3 states *"it doesn't patent solutions but it patents problem areas."* This way it is (nearly) impossible to know for certain that you haven't infringed on any software patents, or differently put, you know for certain that you have infringed on software patents, like interviewee #6 says *"I wilfully ignore it, I don't look at patent lists because I know it will be the case with half of my code."* Concepts that are widely known, standardised, and used in code may (very likely) be patented. Large companies can hire a team of lawyers to check patents, after which a risk assessment can be made, but there is no guarantee that there are no patents left uninfringed. For small and medium enterprises, open source communities and people working on their own it is not doable to focus on this. Most of the people that were interviewed state that this is not a big problem for open source communities which often have an underlying foundation, and not one specific person to sue. An alternative would be to sue all the people working on the project but this would not be worth the effort because there is not much to be had from these people. Interviewee #5 states that it is possible that large companies attack individual programmers that do something that the businesses don't like. Although this does not happen frequently it is a risk for individual programmers. It is also a risk for small and medium enterprises which can be easily pressured into submission by large companies with many lawyers and large patent portfolios. Like interviewee #5 says *"... and small companies get their necks twisted off right away, because once you get some attention from one of the big boys as a small company you will be sued, and there is nothing you can do about it."* It is not only large software companies that have software patent portfolios, there is also the "patent troll" phenomenon. These companies exist solely to rake in as many patents as they can in order to sue as many companies as they can, of course for the purpose of making lots of money. And of course this completely defies the purpose patents were originally created for which was to stimulatie openness of inventions and to encourage people to innovate. Patents are not used to consult when you need a solution to a problem, nowadays (in software) patents are only looked up after you have solved a problem to see if you are not infringing any patents. Then you have to pay lots of money for licences for those patents and still risk being sued for infringing patents you haven't found. *"So A it has no economic value, and B it is used as anticompetitive measure, which of course is bad for innovation and small companies"* as stated by interviewee #5. The final argument is that the life of a patent is too long for IT industry standards. Interviewee #2 states *"but I if have invented that genious idea, then I would find it unreasonable that I would still have the sole rights to it in 20 years, because the IT industry moves so incredibly fast that what was a brilliant idea 5 years ago is now used in every program."*

7.1.2 Alternatives

The interviewees have also been asked if they think there is a good alternative to software patents. Most of them could not think of one that they would favor over abolishing software patents altogether. Interviewee #1 states *"those iPhones are reproduced on a large scale in China, but the quality of the product is not the same as the original, if only it is because it runs on different software, and I think it is already protected in that way."* Copyright is not seen as a suitable alternative for software patents, or as enough protection for software.

7.2 What does the open source community think of software piracy?

This section covers subquestion 4, this question is as stated in the title of this section. The results of this will be the indicators given in the "method" chapter, and also in the names of the subsections of this section.

7.2.1 Belief of the person that all software should be open

Interviewee #1 thinks that software should be open, not from the perspective that it should be free, but from the perspective that you should be able to see what it does and how it works. Companies have intellectual property so if their product is licenced properly someone else cannot use the exact same product anyway. Open source software is better for innovation because according to interviewee #1 the internet wouldn't have been as popular as it is now if it had been closed. To illustrate this he states that Microsoft did not believe in the internet and decided to make it's own version, the Microsoft Network, which was closed and has completely flopped. Interviewee #2 has a different opinion that has more to do with choice. He thinks it should be up to the programmer what he does with his software. He also thinks it is good that some software is closed and that companies have made that their earning model, because from that closed software because it is often the case that open source alternatives are made. Interviewee #3 views it as an ethical decision. You should be able to know what the software that runs on your computer does. Also software is like a recipe, and that kind of information or knowledge should be shared from a moral perspective. He says *"the funny thing about ethics, which is now also very relevant in the Netherlands, you cannot force your own ethics upon others ... it is wrong to make proprietary software and it is dumb to use it"* however he does not think that all software should be open source. However he says that much time is put into making open alternatives for proprietary software, and the goal is to create an entirely open computer environment. When interviewee #4 was asked if he thougt all software should be open source software he said *"No, it doesn't have to be. I don't think you can force that. It would be better. It would be good if everyone would share their ideas as is done in science, but I don't think you can force that."* Interviewee #5 shares this opinion and adds that it would be good for competition and an open market. It would make it possible for people who want to create a word processor to be able to also open Microsoft Word documents. He says that open standards for document formats and communication protocols would be ideal for free competition. He also thinks that there would not be significant improvements if all software would be open source. He thinks Microsoft's source of income would disappear which he would regard as a bad thing. Positive things would be that people could see how the software works, play with it, and improve it, and that maybe things like backdoors for the CIA would be found. For people in the Western world it would be better because software would be cheaper. But people in Pakistan who use a Blackberry now can be eavesdropped on by the government. If the software on their Blackberry was open source they could prevent that, so for people in these kinds of countries it would be more important because for them it would be about freedom, which is something people in the western world do not have to worry about. Therefore he says that for us open standards would be the most important. Interviewee #6 also states that open source should not be imposed on people, but says that it would be desirable, and a more useful business model because if people can use open source software they don't have to create their own versions of the same thing. It would also give a person more control over (for example) their privacy sensitive data, more choices, an cheaper products.

7.2.2 Belief of the person that there should be no software patents

The results for this indicator can be found in the previous section of this chapter "Why is the open source community against software patents?" in the first part of the subsection called "Opinion".

7.2.3 Belief that less patents would lead to less piracy

This indicator proved to be a bit more difficult than expected. This was due to the fact that the hypothesis was that if a particular program is patented there would be no (open source) alternative, and people would therefore be more inclined to pirate the program. In practice, as I have learned from the interviews, it is not the case that entire programs are patented, but it is components or methods used in the program that can be patented. Also the open source community does not think very highly of patents and they are therefore often ignored. For that reason it can be assumed that even though a program is patented that there will still be someone who creates and releases an alternative version of it. To obtain an answer to this question anyway a hypothetical scenario was presented of a program that is now widely used and has open source alternatives, and the people who were interviewed were asked if they thought that that program would be pirated more if it was patented and there were no alternatives. The interviewees were also asked if they thought there was a link between the piracy rate and the amount of patents.

When presented with the hypothetical scenario interviewee #1 did not think that the piracy on that product would increase, depending on the licence fees, if they are what they are today he thinks the piracy rate will be about the same. He also did not think that there is a direct link between patents and piracy. Interviewee #2 when presented with the scenario said that the company then has a monopoly on the product and can ask whatever price they would want for it, therefore he assumed that the price would rise significantly, which would in turn cause the piracy rate on that product to increase. When asked if there is a link between patents and piracy he said that he saw the link if whole programs were patented, but in practice methods in programs are patented. In that case there can be alternatives to the program, but these might not work as well, or have a less nice interface. He said that because of the existence of alternatives the company with the patent cannot raise their price significantly. He concluded that as long as the patent allows for alternatives that patents do not cause more piracy. Interviewee #3 also thinks that in the presented scenario the piracy rate would increase on the patented product. This is the opinion given by most of the people that were interviewed, with the exception of interviewee #1. When interviewee #3 was asked if he thought that more patents would cause more piracy he said *"No, I think piracy behavior has much more to do with financial reasons and laziness. It is easier to download something that to purchase it in the correct way."* Finally interviewee #4 said *"I think piracy is a problem, but I do think that the more difficult it is to obtain open alternatives, the sooner people will turn to piracy."*

7.2.4 Opinions on piracy

This subsection is a merger of three indicators, namely "belief of the person that it is okay to pirate software", "belief of the person that it is okay to pirate *patented* software", and "piracy behavior of the person". The second indicator is an issue because actual programs are not (usually) patented as stated in the beginning of the previous subsection. Therefore it is not really possible to pirate patented software, you can only pirate software that contains patented components or methods. However, it is possible that these components or methods are what makes a program more desirable than it's alternatives. Also there were some factors that affected people's opinions on software piracy, therefore these two indicators are merged into one subsection because it is more interesting to present all opinions on piracy together. The third indicator is added to this section because the piracy behavior of the interviewees is related to their opinion on software piracy.

Everyone that was interviewed was in principle against piracy, but for some people there were a few exceptions in which they state that piracy could be acceptable. However the general opinion is that laziness and people not being willing to pay money for software when they can get it for free is seen als the general motivation for piracy and this is seen as a non-excuse for piracy. Interviewee #3 states *"If someone gives you something under an open licence, thus explicitly giving you permission to use it then it's okay. If someone says no, that is not allowed, then you are to respect that."* Interviewee #4 does find a scenario in which case piracy is acceptable. This is for example when you need Microsoft Word to open documents of the corresponding file format, or when only documents of a certain (closed) file format are accepted. In that case interviewee #4 says it is fair to pirate the necessary software. But this is only because you are forced to use it, not because you want to use it, or will continue using it for other purposes. Interviewee #5 states that if there are no alternatives for a program that he would find piracy acceptable, but on the other hand, piracy does not solve the problem of there not being open standards. For that you would need the source code, and piracy does not give you that. So in principle you may even start contributing to the problem by illegally downloading Microsoft Word because then you might start sending people documents in that formal as well. Interviewee #6 states that if you prefer Microsoft Office over the alternatives then you should be willing to pay for the extra value that it provides. He says he worked on the import filter of an alternative to Microsoft Word to improve the interoperability, and that is his way to break the monopoly. Interviewee #6 states *"Well, in most cases I think it's wrong and for the most part it is based on laziness. Me, I'm prepared to use software that may be a little less, may be a little bit more difficult to use, if the software is open. But I don't expect a taxi to take me into town for the same price as the bus. If I want that extra comfort then I will pay for it, and that means that I will have to make choices."* The last indicator for subquestion 4 is "Piracy behavior of the person". Most of the people that were interviewed do not pirate software. Only interviewee #4 stated that he had pirated Microsoft Word. This was not because he intended to use it to create documents, but because he kept receiving documents that he could not open without the program. Aside from that he does not pirate software.

Chapter 8

Discussion & Conclusions

The first two sections of this chapter consist of the answers to the subquestions that will be used to answer the main question. The third section will discuss these results. Next the conclusion of this research will be presented, and following that recommendations for further research will be done.

8.1 Open source – patents – piracy

In this section the answers to subquestions 2 and 3 will be presented. These questions were written to determine if patents are the link between open source and software piracy.

Sub question 2 asks “what effects do software patents have on software piracy?”. To answer this question research into relevant literature has been done. Section 6.2 of this thesis covers this literature. To sum up, this states that countries that have high protection for intellectual property have lower software piracy rates than countries with less protection for intellectual property rights. The literature also shows a strong negative correlation between the measure of patent protection and the rate of software piracy in 57 countries. This has led to the conclusion that patents have a negative effect on the software piracy rate. Therefore it cannot be said that if there are less software patents there is less software piracy, which disproves the original hypothesis. However, given that there is a strong negative correlation between the amount of patents and the amount of piracy, meaning if there are more patents there is less piracy, something can be said about a connection between open source and software piracy. Given that the open source community opposes software patents, and less patents means more software piracy, this could be a link between open source and software piracy.

Sub question 3 is “why is the open source community against software patents?”.

The reasons that were found for this can be found in sections 6.3 and 7.1 of this thesis. The point of this question was to find out if these reasons were connected to piracy, which could then be a link between open source and piracy. However these reasons did not have anything to do with piracy and thus do not link open source to piracy.

8.2 Open source – piracy

This section will present the results of sub questions 1 and 4. These questions were written to determine if there is a direct link between open source and piracy.

Sub question 1 asks "what are the differences between open source and software piracy". The answer to this question can be found in section 5.3 of this thesis. To sum up, open source differs from software piracy on all but one of the factors used for comparison. These factors were licences, motivation, business methods, and distribution. The only likeness can be found in distribution. Disregarding licences, with piracy, software is made available for (free) download as long as someone is distributing it. The same is the case with open source software. From a theoretical standpoint this is the factor on which open source and software piracy meet.

Sub question 4 asks "what does the open source community think of software piracy?" To help answer this question there were six indicators, of which the outcomes were as follows. Indicator 1 is as follows "belief of the person that all software should be open." The overall opinion on this was that it would be a good if all software were open source, although there might be negative consequences for large commercial companies (this was regarded as good by some, and as bad by others). But the main opinion was that it should not be mandatory. It should be the choice of the person who creates the software whether the software is open or closed. Indicator 2 is "belief of the person that there should be no software patents." On this the majority opinion was that software patents in the way they exist today do not fulfill the purpose that patents were made for. It is said that there are many negative aspects to software patents. It is also the general opinion that having no software patents would be preferred over the current system for software patents. Indicator 3 is "belief that less patents would lead to less piracy." To this the majority reaction was that in the hypothetical scenario the piracy rate would increase, however this was an unlikely scenario. Most of the people interviewed did not think that patents had anything to do with piracy. Indicators 4 and 5 were taken together in the results. The indicators are "belief of the person that it is okay to pirate software" and "belief of the person that it is okay to pirate *patented* software". The general opinion was negative regarding piracy, it is seen as something that is motivated by laziness and unwillingness to pay for software. Here an important opinion again is that it is people's own choice what they do with their software and if they want to make it closed they can do that, but this does not justify piracy. On other motives for pirating software the opinions are divided. These can be found in subsection 7.2.4. The makes it easy to predict the outcome of the last indicator "piracy behavior of the person". The people that were interviewed (generally) do not pirate software, because they believe it is wrong, but also because they mainly use open source software.

8.3 Discussion

In section 3.4 it is says that there would be at most six interviews due to time contraints. Six interviews have been conducted, this is not a very large number, however the data collected is believed to be representative for the target group. This is the case because the ideas and beliefs expressed in the interviews were quite consistent on a general level. On a more specific level the opinions were not always the same, but it is not the case that everyone had a completely different view on the subject. Therefore it is not expected that further interviews would change the results significantly. Because of this data saturation has been accomplished.

In the first section of this chapter it is reasoned that because the literature shows a negative correlation between the measure of patent protection and the software piracy rate (meaning less patents means more piracy) and the open source community is opposed to software patents, this would link the open source community to software piracy. However this correlation may also be the result of cultural factors meaning that if people live in a collectivist culture they find that intellectual property rights are public property and it is therefore not higly protected, and people do not see sharing of software (in the form of piracy) as something that should not be done. It is also shown that in individualist countries this is the other way around. This could be the cause of the correlation, which would mean that the measure of protection for intellectual property does not influence the software piracy rate, but the culture does. This would then mean that the link that was drawn between open source and software piracy through patents is invalid. Another factor that indicates this connection to be invalid is that the results of the interviews of all but one person show that the people interviewed do not think there is a connection between software patents and software piracy.

The previous section of this chapter shows that from a theoretical standpoint distribution is the factor on which open source and software piracy meet. However this likeness is only true when you disregard licences. This is because an open source licence allows someone to freely redistribute the software and a commercial licence (most likely) does not, which makes it piracy. The question is if the open source community agrees with the violating of commercial software licences. The results of the interviews show that this is not the case, it is shown that the attitudes of the people that were interviewed are negative towards software piracy.

8.4 Conclusion

The previous sections in this chapter have concluded that there is no significant connection between software piracy and the open source community through software patents. It has also been concluded that the theoretical likeness between open source and software piracy is not significant enough to establish a relationship between open source and software piracy. This leaves the results of sub question 4 to establish a direct relationship between the open source community and software piracy.

The answer to sub question 4 is that choice is the key term for the open source community. Yes, they do believe that it would be better if all software were open source, and yes, they are opposed to software patents. However, it has been said that open source should not be mandatory because it is the choice of the person who created the software if the software is made open or closed. It is the opinion that if people want to make their software closed that that is their choice and that this is not an exuse for people to pirate that software. The open source community makes alternatives to many commercial software products, therefore they say that if you want the added value that the commercial software offers you should pay for it. It is also claimed that it is unlikely that there is software for which alternatives are not made due to software patents, therefore this cannot be used as an excuse to pirate software. The people that were interviewed also claim to not engage in software piracy. The reasons for this is because they think it is wrong, and also because they mainly use open source software.

[1] claims that the opinion of other towards piracy is often positive. The people that were interviewed say that they advise people not to pirate software. This suggests that the open source community's attitude towards software piracy is different from the general attitude towards software piracy. This attitude is the relationship between the open source community and software piracy. Of course further research will have to be done to prove that the attitude of the open source community is significantly more negative than the general attitude. However it

seems safe to say that this is the relationship between the open source community and software piracy.

8.5 Future research

For future research on this topic there are a few topics that could be interesting to do futher research into.

First of all, because the literature and the interviews have shown that the open source community opposes software patents in the current situation. One interviewee has also claimed that large software companies are also unhappy with software patents, and only have large patent portfolios to defend themselves against patent claims. If there were no software patents it would be less cost and time consuming to create software, due to the fact that no effort or money has to be put towards software patents. So first of all it would be interesting to find out if there is merit to this claim. Second of all it would be very interesting to find an alternative to software patents that would please both the open source community and commercial software companies.

Another interesting point is that it is claimed that many software patents are trivial and should never have been granted in the first place. It would be interesting to research why this is the case and if this is the case in other fields as well, and how this can be prevented.

Finally it is claimed in this paper that the open source community's attitude towards software piracy may well be more negative than the general opinion towards software piracy. There is much research to be found on factors that influence piracy and also on the attitudes of some specific groups towards software piracy, however it would be interesting to know for example what the attitude of people from the Netherlands in general is towards software piracy. This could prove or disprove the hypothesis that the attitude of the open source community towards piracy is more negative than the general attitude.

Bibliography

[1] Al-Rafee, S., Cronan, T. P., 2006. Digital piracy: Factors that influence attitude toward behavior. *Journal of Business Ethics*, 63(3), pp. 237-259.

[2] Andrés, A. R., 2006. The relationship between copyright software protection and piracy: Evidence from Europe. *European Journal of Law and Economics*, 21(1), pp. 29-51.

[3] Business software alliance, 2010. *Definition of Software Piracy - Business software alliance*. [online] Available at: http://www.bsacybersafety.com/threat/software_piracy.cfm [Accessed 25 November 2010].

[4] Business software alliance, IDC, 2010. Seventh annual global BSA/IDC global software: 09 Piracy study, [pdf] Available at: http://portal.bsa.org/globalpiracy2009/studies/09_Piracy_Study_Report_A4_final_111010.pdf [Accessed 5 November 2010].

[5] Couser, J., 1999. Software piracy and the Doris Day syndrome: some legal, ethical and social implications of contemporary conceptions of property. *International journal of Law and Information Technology*, 7(1), pp. 1-50.

[6] Dakin, K., 1997. What if there were no software piracy? *Software, IEEE*, 14(1), pp. 20-21.

[7] DiBona, C., Ockman, S., 1999. *Open sources: Voices from the open source revolution*. [e-book] O'Reilly Media. Available at: http://oreilly.com/catalog/opensources/book/toc.html [Accessed 11 November 2010].

[8] Guntersdorfer, M., 2003. Software Patent Law: United States and Europe Compared. *Duke Law & Technology Review*, 6.

[9] Hars, A., Ou, S., 2002. Working for Free? - Motivations of Participating in Open Source Projects. *International Journal of Electronic Commerce*, 6(3), pp 25-39.

[10] Hecker, F., 2002. Setting up shop: The business of open-source software. *Software, IEEE*, 16(1), pp. 45-51.

[11] Hunt, R., Bessen, J., 2004. The software patent experiment. *Business Review, Federal Reserve Bank of Philadelphia*, Issue Q3, pp. 22-32.

[12] Klemens, B., 2010. *Bilski and software patents - opensource.com.* [online] Available at: http://opensource.com/law/10/7/bilski-and-software-patents-0 [Accessed 5 November 2010].

[13] Lakhani, K. R., Wolf, R. G., 2003. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. MIT Sloan Working Paper No. 4425-03. Available at: http://ssrn.com/abstract=443040 or doi:10.2139/ssrn.443040 [Accessed 11 November 2010].

[14] Limayem, M., Khalifa, M., Chin, W. W., 2004. Factors motivating software piracy: a longitudinal study. *IEEE Transactions on Engineering Management*, 51(4), pp. 414-425.

[15] Marron, D. B., Steel, D. G., 2000. Which countries protect intellectual property? The case of software piracy. *Economic Inquiry*, 38(2), pp. 159-174.

[16] Rouvinen, P., Stankiewicz, R., 2009. Are Intellectual Property Rights Hindering Technological Advance? The Need for Technological Commons. *Review of policy research*, 26(1-2), pp. 195-217.

[17] Sims, R. R., Cheng, H. K., Teegen, H., 1996. Toward a Profile of Student Software Piraters. *Journal of Business Ethics*, 15(8), pp. 839-849.

[18] United States Patent and Trademark Office, 2004. *What Are Patents, Trademarks, Servicemarks, and Copyrights?.* [online] Available at: http://www.uspto.gov/web/offices/pac/doc/general/whatis.htm [Accessed 11 November 2010].

[19] Ward, J., 2010. *Bilski v. Kappos. Anticipate This! Patent and Trademark Law Blog.* [online] Available at: http://anticipatethis.wordpress.com/2010/07/01/bilski-v-kappos [Accessed 5 November 2010].

[20] Wikipedia, 2011. *Octrooi op software - Wikipedia.* [online] Available at: http://nl.wikipedia.org/wiki/Octrooi_op_software#Stemming_Europees_Parlement [Accessed 23 March 2011].

[21] Wikipedia, 2011. *Software patent debate - Wikipedia, the free encyclopedia.* [online] Available at :http://en.wikipedia.org/wiki/Software_patent_debate [Accessed 23 March 2011].

Appendix A

Interview guide

This is an English translation of the interview guide that was used to conduct the interviews, the translation was performed by the author. Note that the headings represent the indicators from sub question 4. The questions below the headers are not necessarily asked, but function as extra questions that can be asked if necessary.

Introduction

In this section of the interview the interviewee will be thanked for participating in the interview. Futhermore the topic of the thesis will be explained, as well as the reasoning behind this topic. Then some questions about the open source activities of the interviewee will be asked, such as the ones below.

- Can you tell me about the open source project(s) you work on?
- Is there a proprietary alternative?
- How much time do you spend on this work?
- How many people work on the project? Are these people only from the Netherlands?
- Why is this project open source?
- Is everyone on the project a firm believer in open source?
- What do you gain by working on this project?

Should all software be open?

- Why do you think open source is important?
- What would be the advantages (for consumers/businesses/developers)?
- How important/widespread is open source nowadays?
- What aspects of open source are most important to you?
- What would change if all software were open source?

What do you think of software patents?

- What do you think of proprietary software companies that sue open source initiatives for patent infringement?
- Does the open source project you work for suffer under this?
- Do you experience restrictions due to software patents?
- Do you check if there is a patent on the kind of software you are creating?
- Should there be any software patents?
- What effects do patents have?
- Some state copyright is enough to protect software, do you think this is the case?
- What do you think of patents in general?
- What do you think of copyrights on software?

What is the effect of patents on piracy?

- Do you believe that people are less inclined to illegally download a software product if there is a free alternative?
- Do you believe less software patents will lead to less piracy?

Is it okay to pirate (patented) software?

- Do you believe that it is justified to illegally download software for which there is no free alternative?
- Do you believe it is justified to illegally download patented software?
- Do you believe it is justified to illegally download proprietary software even if a free alternative exists?

Own piracy behavior

- Have you ever pirated software?
- Have you ever made proprietary software available for others to download?

Conclusions

- Summarize most important statements person has made.
- Ask if person would like to add something or has any comments.