Bartłomiej BIGA*

iD https://orcid.org/0000-0003-0837-4923

THE NEGATIVE EFFECTS OF SOFTWARE PATENTING[1]

Abstract

Background: The approach to protecting software, which is a key issue of intangible asset management in many companies.

Research purpose: The purpose of this paper is to examine certain IT companies' assessments of software patentability and to compare them to the literature.

Methods: This is done through qualitative research by interviewing managers from ten different software companies. Comparing their reasons for not patenting their software with the generally stated reasons gathered from a literature review made it possible to identify the relationship between the theoretical and empirical findings.

Conclusions: The qualitative research also provides empirical data about the strategies that are used in certain small- and medium-sized software companies to deal with software patenting problems, especially patent thickets. Different aspects of the inadequacy of legal protection for software that may be solved at the company-level are also explored in the paper.

Keywords: intellectual property, software patents, intangible assets management, industrial property law.

JEL classification: K11, M15, O32, O34

1. Introduction

Software protection is a key issue of intangible asset management in many companies. It plays a crucial role not only in solely software companies; it is of great importance in most IT firms. On the one hand, legal protection (like patents and copyrights) is seen as a default strategy in the business. On the other

---

* Ph.D., Cracow University of Economics, Faculty of Public Economy and Administration, Department of Public Policies; e-mail: bartlomiej.biga@uek.krakow.pl

hand, in the digital world, patent procedures are too slow. Moreover, software patents are seen as a relatively weak mechanism, which is a result of the high level of unpredictability in intellectual property court cases. Yet, the practice of software patenting is widespread, mostly in the US. Additionally, the popularity of software patents is growing very fast.

Many authors like Lemely,[2] Miller,[3] Coriat and Orsi,[4] Campbell-Kelly,[5] and Mann[6] have identified the main drawbacks of software patents. Software is a peculiar category because products within this area are just the implementation of certain functionality. Software is logic algorithms for processing data that are implemented via stored instructions that reside on a disk or other storage medium or in read-only memory. This is a widely accepted definition, making it possible to select such categories from the catalog of patent classifications containing software patents.[7]

Controversies related to software patents are one of the main arguments for liberalizing industrial property law.[8] It is considered that the weaknesses and irrationality of patents, which appear under certain circumstances, are most clearly visible in the area of software. However, abandoning software patenting is hard to imagine because software is a significant part of innovation. Moreover, it is strongly interlinked with non-software inventions. Therefore, it could be very difficult to set the boundaries of a patentable area. Since society is "doomed" to keep software patents which are poorly constructed, it is necessary to improve the efficiency of this form of protection. What needs to be taken into consideration when conducting such reforms is the reason for patenting dysfunctions – their nature and mutual relationships. This is a crucial task for the whole patenting system, because the share of software patents in the total number of patents is growing rapidly.

---

[2]   **M.A. Lemley**, *Software Patents and the Return of Functional Claiming*, Wis. L. Rev 2013, p. 964.

[3]   **S.P. Miller**, *"Fuzzy" Software Patent Boundaries and High Claim Construction Reversal Rates*, Stanford Technology Law Review 2014/14, pp. 814–819.

[4]   **B. Coriat**, **F. Orsi**, *Establishing a New Intellectual Property Rights Regime in the United States*, Research Policy 2002/31, pp. 1503–1505.

[5]   **M. Campbell-Kelly**, *Not All Bad: an Historical Perspective on Sostware Patents*, Michigan Telecommunications and Technology Law Review 2005/11, pp. 246–248.

[6]   **R.J. Mann**, *Do Patents Facilitate Financing in the Software Industry?*, Texas Law Review 2005/83, p. 1028.

[7]   **J. Bessen**, *A Generation of Software Patents*, Boston University School of Law Working Paper 2012/1, p. 13.

[8]   **J. Bessen**, **M.J. Meurer**, *Patent Failure*, Princeton University Press, Princeton and Oxford 2008, pp. 2–11.

However, some authors recognize areas where software patents could be useful. They indicate that patents provide a sort of insurance against the risk associated with intangible assets for venture capitals. From the founder's perspective,[9] it cannot be denied that patents can provide a range of security options.[10]

Each computer program is protected by copyrights. The code of the program is automatically protected. However, software patents give stronger protection for authors because they cover implementation and programming methods. Consequently, if another programmer achieves the same effect with a different code, it is not a violation of copyrights, but it can be considered patent infringement. Thus, copyright law is able to protect ideas to a limited extent. There are many differences between these two protection regimes, and some creators are interested in protecting their work considering both of the regimes to guarantee as broad protection as possible. However, from the perspective of law and economics, the key question is the cost-benefit balance of such practices for society.

2. Literature review

The results of this text correspond with some reasons for software patent problems that have been indicated in the literature. The review of the literature and available data makes it possible to indicate several reasons for software patent problems. Relatively little attention is devoted to the relationship between these determinants. However, such analysis is necessary to understand the complexity of the discussed phenomenon. First of all, one needs to identify which causes are basic (primary). In other words, what circumstances and practices related to software are the real sources of dysfunctions in this part of the patent protection system. These factors subsequently cause several more detailed problems, which have other implications.

It is worth putting the above-mentioned results together with general factors for software patents flaws. The selection below is the result of desk research. In the literature, authors have identified 12 factors that are responsible for the negative assessment of patenting in the context of software. Some of

---

[9]   **L.F. Beltrán-Morales et al.**, *The development of ecosystems for technology transfer in Mexico: the role of Patenting Centers*, Queen Mary Journal of Intellectual Property 2018/8, p. 333.

[10]   **C. Dent**, *New Insights in Patent History: an Application of Evolutionary Theory*, Queen Mary Journal of Intellectual Property 2018/8, p. 171.

them describe the same issues using different terms, but none identified all 12 factors. The authors focus on the details of a certain issue. As a result, they do not see the whole picture or the relationships between the issues (in particular, what the reasons and consequences are). Therefore, the wider analysis in this text provides new insights.

The list of issues related to the negative effects of software patenting includes:

1. The lack of a need for patenting in the software industry;
2. The use of patents against their intended socio-economic purpose;
3. Determining the requirement for non-obviousness is at a low level;
4. The rapid growth in granting software patents;
5. Patent offices accepting functional claims;
6. Fuzzy boundaries of certain patents;
7. The low average value of software patents;
8. The emergence (formation) of "patent thickets";
9. The formation of a convenient space for non-practicing entities;
10. Major obstacles for free software developers;
11. Increasingly tight restrictions on the possibility of implementing large projects;
12. The high risk of litigation.

FIGURE 1: *The scheme of causal relationships between the reasons for difficulties related to software patents*
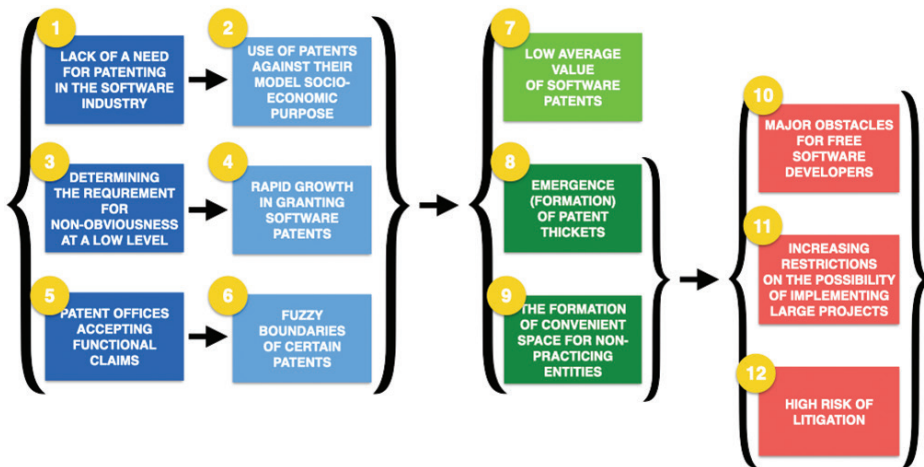
S o u r c e: author's own compilation.

1. The lack of a need for patenting in the software industry

Initially, general skepticism for this kind of intellectual protection was visible in the software industry. In subsequent years, most software patents were submitted by companies that were not strictly involved in the software industry but in the IT sector. Thus, it can be concluded that software patents do not bring substantial net benefits to the software industry. As software patents generate many lawsuits, there are high social costs. Moreover, as Miller[11] noticed, there is strong empirical evidence that software patents have contributed to a considerable decline in the quality of the whole patent system.

2. The use of patents against their intended socio-economic purpose

As software patents were not needed to provide incentives for innovation or to spread the results quickly, the possibility of using software patents for alternative purposes has emerged. Thus, software patents are used to create patent portfolios, which are useful when negotiating cross-licensing, as they decrease the risk of being sued by competitors, and sometimes they simply block new players.[12] Such entities do not use patents based on their main economic purpose. The growing share of such activities undoubtedly contributes to the deformation of the patent protection system.

3. Determining the requirement of non-obviousness at a low level

In discussions about software patents, the question about restricting patenting only to great ideas is often raised. The global standard, which requires something be non-obvious to a skilled person in a particular art or science, is, in fact, very low.

Mandel[13] demands legal reforms to provide patent offices with strong enough bases to reject many weak patents. Critics of the practice of requiring non-obviousness do not deny that only significant inventions deserve patent protection. However, rejecting patent applications on the current interpretation of non-obviousness could be considered illegal. Therefore, there is an urgent need for deep legal reform as the patent system needs a legal method to protect inventions that are not significantly innovative.

---

[11]   **S.P. Miller**, *"Fuzzy" Software Patent Boundaries*…, p. 838.
[12]   **J. Bessen**, **R.M. Hunt**, *An Empirical Look at Software Patents*, Journal of Economics & Management Strategy 2007/16, p. 184.
[13]   **G.N. Mandel**, *Patently Non-Obvious: Empirical Demonstration that the Hindsight Bias Renders Patent Decisions Irrational*, Ohio State Law Journal 2006/67, pp. 1451–1455.

4.   The rapid growth in the number of software patents

The weakening in respect in terms of the non-obviousness requirement creates the possibility for rapid growth in the number of software patents. In the context of several patented software solutions, which give wider protection than copyrights, this could significantly block development. In some areas, the risk of an unconscious breach could be so high that programmers could be discouraged from working. In practice, the inability to use optimal solutions in fundamental aspects results in wider monopoly power of the patent holder than should arise directly from the patent itself. In the analysis by Bessen,[14] the growth in the number of software patents is faster than the growth in total patents granted. Thus, the share of software patents in the general patent pool is growing, too.

5.   Patent offices accepting functional claims

The next issue is the permission to patent ideas, which is characterized by a high level of abstraction, and their impact on specific technical solutions is only theoretical in the context of industrial applicability requirement, which is blurred in industrial property law. Software patent lawyers are writing more and more patent claims in broad functional terms. Put another way, patentees claim to own not a particular machine or even a particular series of steps to achieve a goal, but the goal itself. As a result, overbroad patents overlap and create patent thickets.[15] If the aim of the patent is to encourage innovation, the scope of protection has to be related to the scope of the disclosure.[16]

Therefore, criticism of functional claims is widespread. Anderson and Menell[17] draw attention to the fact that patent offices do not have enough competence to interfere in the shape of such claims during the examination process. They analyze the impact of the claims form on decisions and subsequently on litigation cost.

---

[14]   **J. Bessen**, *A Generation of Software Patents…*, p. 15.

[15]   **M.A. Lemley**, *Software Patents and the Return of Functional Claiming*, Wis. L. Rev 2013/905, p. 964.

[16]   **S. Graham**, **S. Vishnubhakat**, *Of Smart Phone Wars and Software Patents*, The Journal of Economic Perspectives 2013/27, pp. 83–84.

[17]   **J.J. Anderson**, **P.S. Menell**, *Informal Deference: a Historical, Empirical, and Normative Analysis of Patent Claim Construction*, Northwestern University Law Review 2004/108, pp. 76–77.

6. Fuzzy boundaries of certain patents

The fuzziness of boundaries of individual property rights seems to be their defining characteristic since their precise delimitation can only be achieved through the process of interpretation, rather than relying solely on organoleptic tests. This weakness in an area of software that practically has no physical manifestation is particularly evident. The lack of tangible elements makes it hard to describe the limits of a particular exclusive property right precisely.

7. The low average value of software patents

These conclusions are consistent with the results of research conducted mainly in the United States. The value of most patents is calculated based on the analysis of the royalty rate. Thus, it has been shown that software patents generally have a lower value than other types of patents. Noel and Schankerman[18] linked the low average value of software patents to the market value of software companies. They demonstrated that there are large, positive technology spillovers from R&D for software firms, but patenting by technology rivals reduces the firm's R&D investment, patenting, and market value. Moreover, the greater fragmentation of patent rights increases both R&D and patenting by the firm (reflecting a greater need to have an arsenal of patents to resolve disputes when there are many players), but it lowers market value because transaction costs are higher.

8. The emergence (formation) of "patent thickets"

When analyzing the social costs of software patents, the issues of discouraging investment in innovation take the spotlight. In this area, exclusive rights not only secure the results normally expected in terms of the patent, they may even lead to the opposite results. In this sector, companies acting as "patent thickets" have been noted. When several players gather a significant number of patents, in practice, they create a barrier to market entry that independent investors find difficult to overcome, thereby lowering the overall level of innovation.

9. The formation of a convenient space for non-practicing entities (NPEs)

The essence of using patents against their intended purpose is an NPE activity – "patent trolls." They abuse the imperfections of industrial property rights and thus often achieve high profits. Most often, they take over already

---

[18]   **M. Noel**, **M. Schankerman**, *Strategic Patenting and Software Innovation*, The Journal of Industrial Economics 2014/61, pp. 514–515.

widespread solutions that no one had reserved before, but they do not intend to use these solutions in their products. Then they threaten entrepreneurs whose advanced products infringe their patents with costly lawsuits, in the end usually offering a settlement. The level is usually estimated in such a way that it becomes undoubtedly favorable for a large company – with no need for costly, and above all, risky litigation. "Patent trolls" are excellent at creating proposals for settlements, which are often concluded even when lawyers assess the lawsuit's chances of success as very low.[19]

10. Major obstacles to free software developers

One of the main antagonists for software patenting is the developers of Free Libre/Open-Source Software (FLOSS), who fear it will make programming impossible for them. They are deprived of defense mechanisms enjoyed by large corporations, such as blanket- and cross-licenses. In fact, they are often not even able to participate in any financial flows.

The expansion of patent rights to cover information processes has caused a stir in the free software movement. Previously, software was protected as an artistic work under copyright law. Software patents pose a threat to General Public Licenses because companies can follow copyright law and abide by the terms specified in the free license while restricting access to the source code through patent law.[20]

11. Increasingly tighter restrictions on the possibility of implementing large projects

Patents are a threat to software developers because they impose a monopoly on ideas. Creating advanced programs becomes dangerous, and it might become impossible very soon as each program combines many different ideas, with large ones implementing thousands of them. In 2004, it was estimated that the Linux system uses around 100,000 solutions patented earlier. According to Ravicher, the Linux kernel itself could violate 283 patents (http://www.cnet.com/news/group-linux-potentially-infringes-283-patents/).

12. The high risk of litigation

A particular disadvantage for entities operating in the software area is the extremely high risk of litigation. With software, it is difficult to protect oneself

---

[19] **M.A. Lemley**, **A.D. Melamed**, *Missing the Forest for the Trolls*, Columbia Law Review 2013/113, pp. 2180–2181.
[20] **J. Soderberg**, *Hacking Capitalism*, Routledge, London 2008.

effectively from being sued because of, among other things, the high cost of searching databases of existing patents in the context of fuzzy boundaries (6). Another reason is the large number of patents granted (4), which are nourishment for Patent Trolls (9) who act within patent thickets (8). As a result, the risk of unwittingly violating patents is also extremely high.

3. Method

This paper is based on a legal and economics approach, which puts the economic efficiency of regulations in the center. The literature review is supplemented with qualitative research based on interviews with company owners. The interviewees were selected from a population of technology SME's operating under the umbrella of Krakow Technology Park in Poland, an institution supported by local authorities as a part of the Krakow Special Economic Zone located in the biggest city of the Malopolska Region. Most of them are start-ups, and they were selected according to their business profile, i.e., companies were identified whose core activity is related to creating software. The interviews were conducted with the owners of these companies who are also involved in day-to-day management (they perform functions such as CEOs or CTOs).

Additionally, the use of several cognitive perspectives provides an opportunity to build a list of reasons why software patenting is so controversial and widely criticized. It is even more important to identify the relationships between these reasons. Naturally, it is practically impossible to carry out experiments in this area. As the result of the increasing global unification of intellectual property protection, a comparative analysis of legal systems is also increasingly difficult. In the face of the methodological difficulties described above, it is necessary to seek causal relationships based on rational choice theory in the context of the legal and economics approach.

The subject of the analysis covers the following stages of business activity:
a) deciding to enter a software-related business (preceding any patent-related issues);
b) creating intellectual property (IP) – stimuli to create stop creating a certain type of IP, the speed of patent races;
c) selecting IP management methods – keeping company secrets, applying for a patent, relying on copyrights, making defensive publications, contributing to the open source, *etc.*;
d) enforcing rights – tendency to sue, predictability of court verdicts, *etc.*

4. Results

Only one of the ten studied companies is involved in patenting software. An additional one has patented inventions with a small software component. The only strong reason to patent software was indicated by a company that is involved in products that use the Bluetooth standard and that is interested in contributing to this patent poll. It wants to be a part of the Bluetooth ecosystem of products as it opens up a very wide market for its products. Consequently, this is a unique situation because this company does not patent to exclude others from using its inventions, but to put it into the open standard and to encourage others to use this particular innovation in their products, as well.

Despite the lack of patenting in most companies, almost all of them take into consideration the fact that software is patentable. There are many reasons why they do not patent their software. The most common answer was a conviction that they do not need this kind of protection. They believe that in their branch, a time advantage is a better source of competitive advantage. They want to focus on developing a better version of a product than waste resources gaining protection for something that can get devaluated in the meantime.

What has been noticed is that it is necessary to describe the details of an invention in a patent application, making imitations easier. However, some software could be relatively easily decompiled, so even without revealing details of the invention in a patent application, imitators can obtain this kind of information. Yet creators of Software as a Service platforms rely on the protection that results from running the whole process on their own machines. Additionally, in some cases, managers desire patent protection for software, but they recognize that their products are too complex to cover such a wide area with patent claims. They also indicate that inventing around a patent is relatively easy in the context of software.

There are also examples of a lack of trust in the legal enforcement of invention protection. Court disputes are seen as unpredictable and expensive, and the researched companies could only occasionally afford to bear the cost of getting worldwide patent protection. At the same time, they do not have enough resources to defend their rights in the case of any violations, and they also mentioned a lack of lawyers who understand IT well.

Nevertheless, the researched companies are aware of the legal framework of invention protection. They declare that they could be forced to pay more attention to software patents if they did business in the US, and that they are aware that the role of software patents is significantly smaller in Europe.

TABLE 1: *Results of interviews*

	COMPANY A	COMPANY B	COMPANY C	COMPANY D	COMPANY E	COMPANY F	COMPANY G	COMPANY H	COMPANY I	COMPANY J
Basic information about the company	Software for the Internet of Things	Monitoring smart building construction systems	Technology for smart real-estate management	Application of artificial intelligence in economy	Solutions for verifying online identity	Creating internet sales platforms	Business analytics tools	Decentralized communication between smart devices	Code analysis tools	Picture analysis tools and sensors
1	2	3	4	5	6	7	8	9	10	11
Do they patent software, and if so, why?	YES – About 50 patents in the US; about 30 are expanded to other countries – US patents are enough in most cases because this market is sufficient as an entry barrier on the global scale – They bought several third-party patents in the past – Only defensive reasons – they are not going to sue anybody – They do not patent for protection needs – They patent to be able to contribute to the Bluetooth standard	YES/NO – They do not patent strictly software, but they have other patents – Their projects are too big to be able to patent them as a whole – They do not want to waste time checking the patent state-of-the-art – As a small company, they could have difficulties executing their rights in the case of infringement – The cost of patenting is seen as a problem – They expect to get some revenue from selling a certain product before bearing the cost of patent procedures	NO – It is seen as too costly and difficult – Several years ago, they considered software patenting, but they decided that it was not necessary – They have a competitive advantage even without registered intellectual property rights	NO – It is hard to patent software because, theoretically, it is not patentable – If it was possible to patent software without violating laws, they could patent and bear the costs of this procedure	NO – They do not patent because they act mostly in Poland and in Europe – If they open a company branch in the US, they will have to reconcile patenting software – Patenting does not bring operational benefits	NO – They believe that in Poland, there is no need to patent software	NO – There is no need – They registered a trademark once – They do not patent because they are convinced that the product protects itself and the code itself does not represent the main value on its own	NO – They think that time advantage is more important – The problem in patenting is that it is necessary to reveal a lot – how something works must be described in detail – They believe that it could be hard to execute their rights in court (plus the problem with the duration of such procedures)	NO – There is a lack of lawyers who understand IT – It is hard to find competent people to examine a patent's purity – The process of patenting is too slow. – It is more important to get a time advantage – It is impossible to successfully protect against IP violation	NO – Patenting procedures take too much time – They do not want to reveal solutions in patent applications because it is easy to change a small detail of a patented invention and create a working imitation – They have had some bad experiences in registering a trademark in the US.
Purposes of patenting software (in general, not only from this company's point of view)	– To increase the company's valuation and to be more credible for investors	– Possessing patents matters when a company is seeking investors, especially when a product can be easily copied – However, in most cases, a business plan and the market details are more important for investors	– It is useful in the context of a company's valuation; therefore, they suppose that it would be useful for them in 3–5 years	– When there is no efficient technical protection, then legal methods could be the only way to secure intellectual property	– It matters in the context of acquiring investors, but it is not necessary	– It is expected by some investors	– In Poland, even investors rarely ask about software patents – it is not important for them	– It is useful only in small, narrow markets with very specific software (for example, for oil companies) – Some investors ask about patents, but it is possible to convince them that a company policy is different, and finally, it is not a problem	– The only purpose of patenting software is to troll others – Very little usefulness in attracting investors could be indicated	– It is expected by foreign partners

TABLE 1 (cont.)

1	2	3	4	5	6	7	8	9	10	11
The scope of using alternative methods of intangible asset management	– Company secrets are too "soft" because clients and investors seek patents	– Trade secrets are a good alternative for patents – Know-how is sufficient protection, especially in more complex products which are joint with unique tangible elements of a product – Their products protect themselves with their complexity	– Based on time-advantage and quick development – They did not value their intangible assets; however, they want to know the value of their intangible assets, but the process is too long and costly	– Trade secrets are useful – The key aspects are the speed of action and customer service – The software itself is not unique enough, and the main value for the customer is beside software – They are aware of a big discrepancy between market value and replacement value	– Time-advantage is the most important – Trade secrets are useful – However, the most important element of a protection system is the hardware devices (where their software could be launched)	– They do not need any particular protection because their software works on their servers – Trade secrets are present in their company but mostly in an informal form – The best strategy is to share the main solution cheaply or even for free and eventually earn on after-sale support	– The code is not the value – people are more important – Even if someone duplicated their software exactly there is no danger of destroying the company's competitive advantage (because they provide a unique user experience) – They do not value intangible assets because their business is based on a personal brand which cannot be valuated properly – Trade secrets are important because of data provided by their customers – The software itself can always easily be copied; therefore, they rely on different aspects	– Wide application of trade secrets (both in employee contracts and with external entities) – They do not valuate intangible assets because there are no reliable methods to do that in start-ups. Moreover, they do not need this information for any reason – They widely share their solution because they care about the range of their peer-to-peer network – They appreciate a strategy based on free-sharing and earning from providing support	– Time-advantage is most important – Company secrets are important mostly in the context of being stolen by employees – know-how plays a crucial role, but it cannot be protected efficiently	– They rely on time-advantage – Additional protection comes from the fact that their software works only on their hardware – They valuated intangible assets on regular basis – but not as a replacement value but in terms of market potential
Dealing with the risk of unintended violations of somebody's intellectual property	– The company is not afraid of unintended violations – However, they admit that generally in the branch, this risk is very high – The Bluetooth standard is some kind of protective umbrella	– If they entered the US market, they could insure themselves from an unintended violation – In such a situation, they could carefully examine the-state-of-the-art – This risk was not the subject of detailed analysis – only in a general way	– In their business, there is no high risk of unintended violation – As long as they do not exceed the local scale, they feel secure	– They have never experienced any unintended violation problems, and therefore, they do not identify such a risk	– In Europe, nobody cares about this risk, and almost nobody examines a patent's purity	– They do not have any anxiety and do not take any preventative actions	– They have never taken this problem into consideration – They believe that they are too small to experience this kind of problem	– They analyze similar solutions and competitors' activities – Sometimes, they hire patent attorneys – Despite such actions, the risk of unintended violation cannot be excluded	– As a result of being Open Source, they are more exposed to intellectual property accusations – However, the open-source society is liked and respected, and this discourages people from suing – They did not take actions to reduce the risk of unintended violation, although it is high	– They examine a patent's purity at a very late stage – when they really commercialized (This stage occurs much later than preparing or even shaping a product for a particular order) – The risk of unintended violation is present in this branch
experiences with patent trolls	– The company is not threatened by patent trolls, and their ability to take action was significantly reduced by US court decisions	– They were warned about patent trolls, but they have not experienced patent trolls' activity	No problems	No problems	– As a result of doing business in Europe, they are not threatened, but they are aware that in the US, they could have such problems	No problems	No problems	No problems	No problems	– They received a proposal to patent software and become a patent-troll, but they refused

S o u r c e: own research.

In Poland, only one purpose of patenting software is widely recognized. It is useful for some investors, but only some of them ask about software patents. For these investors, software patents could increase the valuation of the company. A significant number of investors, however, understand there are multiple strategies in managing intellectual property, and they do not consider the lack of patents a problem.

Instead of patent protection, most of the managers who were interviewed use alternative methods of intangible asset management. The most popular one is the above-mentioned time advantage. The second popular instrument is trade secret, which is used together with the time advantage. There is also a widespread conviction that an appropriate merging of software and hardware (e.g., the offered software can be launched only on the machine created by the same company) is better protection than patenting.

In some cases, companies provide their products for free (or at a very low price), but they build their business model on providing after-sale services. As creators, they are aware that they are the most desired trainers, implementers, and servicemen for these products. In such situations, they do not care about excluding anybody from using their products because the key issue is building as big a market as possible to provide complementary products and services. Consequently, such solutions will become the most important source of income. Such steps are mostly visible in peer-to-peer mechanisms.

In the context of intangible asset management, the interviewed managers do not recognize any appropriate measurement or valuation methods. Even if they are aware that they do not have enough knowledge in this area, they cannot find any appropriate solutions. They know that the real value is quite different from the simply calculated replacement value.

The research revealed an important role of Open Software. Even if a company does not contribute, it uses many open-source solutions. Sometimes this kind of software is recognized as being poor quality, but even then, some companies are interested in developing products that are enhanced versions of open software.

In the literature, one of the biggest problems related to software patents is the high risk of unintended violation. Most of the researched companies are aware of this danger, but they do not take any significant actions to reduce it or actively manage it. They believe that in Poland, they can feel safe. However, all the interviewed managers are aware that if they were to operate in the US or significantly increase the scale of their activities, they would need to examine patent purity. This sense of security is justified because the interviewed companies have not been affected by lawsuits.

Therefore, it is not surprising that most interviewees favor the idea of excluding software from patenting. This kind of protection is useless or inaccessible to small and medium-sized companies. However, they have to deal with patent thickets, which are created by mass software patenting.

5. Discussion

Software patents have very deep roots, and they appear as elements of innovation in so many different sectors that today even radical opponents of software patents do not propose completely abandoning them, only far-reaching restrictions. Currently, one can specify a range of problems that have been caused by the liberalization of patent office policy in the area of software patenting.

There is no way to resolve one of the fundamental problems, i.e., the existence of patents in an industry that does not really need this form of intellectual property protection (aspect 1 from the literature review). This, in turn, has negative consequences with certain entities abusing software patents against their intended purpose, which leads to all further consequences (aspects 7–12).

However, there are also examples of the negative consequences of non-patenting strategies due to the widespread conviction that a patent is the main innovation indicator. As a result, many small companies are undervalued when they do not own a patent portfolio. It is hard to indicate the precise difference, but many investors are looking for patented technologies. Thus, a more open strategy could slow down the development of such companies.

6. Summary

The interviews with managers of IT start-ups revealed that not every issue that was identified in the literature of software patents exists in these kinds of companies. However, the gathered data shows that software patents are not, in fact, useful for these companies. This is in contrast to the officially declared purpose of legal invention protection – to protect small inventors against bigger competitors.

Moreover, in most companies, the fact that software is patentable is a source of problems in the context of unintended violation risk. The interviewed managers see patents as a game that is reserved for the biggest players. And this game is not about protecting inventions; it is an armed race.

As described above, the large scale of software patenting has negative effects for each of the analyzed stages:

a) before the creation of intellectual property – because it discourages people from doing business in an area that is covered with patent thickets. As a result, there is an extremely high entrance barrier caused by uncertainty in many areas. It mitigates innovation in key economic branches when software patenting is widespread (USA), and it does not have such consequences if software patents are rare (e.g., in Poland);

b) the creation of intellectual property – it encourages owners to base their companies on the software-as-a-service model, even if it is not the optimal business strategy. Moreover, it discourages them from creating complex projects because of the high risk of litigation which could delay the premiere or even block the entrance to the market;

c) the selection of IP management method – companies decide to patent not because it is seen as the best method of protection. They do it despite the fact that it is not the most suitable tool to manage their IP. They are forced to patent against its social and economic purpose;

d) the enforcement of the rights – weak patents make court disputes unpredictable. On the one hand, the plurality of patents creates a wide area of huge unpredictability; on the other, it provides a relatively low probability of effective legal enforcement of IP rights.

The researched companies demonstrated that it is possible to find non-obvious business models that let them profit. However, it requires extremely fast improvements to be able to use the time advantage. Sometimes, there is a need to find the main source of profits beyond the primary activity and gain income from after-sale services and selling complementary goods and services. The researched companies declared that they would be forced to pay more attention to software patents if they did business in the US, and they are aware that in Europe, the role of software patents is significantly smaller.

References

Anderson J.J., **Menell P.S.**, *Informal Deference: A Historical, Empirical, and Normative Analysis of Patent Claim Construction*, Northwestern University Law Review 2004/108, pp. 1–83.

Beltrán-Morales L.F. et al., *The development of ecosystems for technology transfer in Mexico: the role of Patenting Centers*, Queen Mary Journal of Intellectual Property 2018/8, pp. 333–347.

Bessen J., *A Generation of Software Patents*, Boston University School of Law Working Paper 2012/1, pp.1–20.

Bessen J., **Hunt R.M.**, *An Empirical Look at Software Patents*, Journal of Economics & Management Strategy 2007/16, pp. 157–189.

Bessen J., **Meurer M.J.**, *Patent Failure*, Princeton University Press, Princeton and Oxford 2008, pp. 1–313.

Campbell-Kelly M., *Not All Bad: An Historical Perspective on Softare Patents*, Michigan Tele-communications and Technology Law Review 2005/11, pp. 191–248.

Coriat B., **Orsi F.**, *Establishing a New Intellectual Property Rights Regime in the United States*, Research Policy 2002/31, pp. 1491–1507.

Dent C., *New Insights in Patent History: An Application of Evolutionary Theory*, Queen Mary Journal of Intellectual Property 2018/8, pp. 171–190.

Graham S., **Vishnubhakat S.**, *Of Smart Phone Wars and Software Patents*, The Journal of Economic Perspectives 2013/27, pp. 67–86.

Lemley M.A., *Software Patents and the Return of Functional Claiming*, Wisconsin Law Review 2013, pp. 905–964.

Lemley M.A., **Melamed A.D.**, *Missing the Forest for the Trolls*, Columbia Law Review 2013/113, pp. 2160–2111.

Mandel G.N., *Patently Non-Obvious: Empirical Demonstration that the Hindsight Bias Renders Patent Decisions Irrational*, Ohio State Law Journal 2006/67, pp. 1391–1462.

Mann R.J., *Do Patents Facilitate Financing in the Software Industry?*, Texas Law Review 2005/83, pp. 961–1030.

Miller S.P., *"Fuzzy" Software Patent Boundaries and High Claim Construction Reversal Rates*, Stanford Technology Law Review 2014/14, pp. 808–845.

Noel M., **Schankerman M.**, *Strategic Patenting and Software Innovation*, The Journal of Industrial Economics 2014/61, pp. 481–520.

Soderberg J., *Hacking Capitalism*, Routledge, London 2008, pp. 1–252.

Bartłomiej BIGA

NEGATYWNE SKUTKI PATENTOWANIA OPROGRAMOWANIA

Abstrakt

Przedmiot badań: Wykorzystywanie różnych sposobów ochrony oprogramowania, co jest kluczowym zagadnieniem dla zarządzania zasobami niematerialnymi w wielu przedsiębiorstwach.

Cel badawczy: Celem artykułu jest zbadanie, jak osoby zarządzające firmami z branży IT postrzegają możliwość patentowania oprogramowania oraz porównanie ich oceny z problemami związanymi z tą formą ochrony własności intelektualnej, które zostały opisane w literaturze.

Metoda badawcza: Artykuł opiera się na badaniu jakościowym – wywiadach z osobami zarządzającymi firmami produkującymi oprogramowanie. Następnie, w celu zidentyfikowania relacji między teoretycznymi i praktycznymi wskazaniami dokonane zostało porównanie rzeczywistych powodów niepatentowania oprogramowania z motywacjami odnalezionymi w przeglądzie literatury.

Wyniki: Badanie jakościowe dostarcza empirycznych danych o strategiach ochrony własności intelektualnej w małych i średnich firmach produkujących oprogramowanie w kontekście problemów wynikających z dopuszczalności patentowania oprogramowania, ze szczególnym uwzględnieniem zjawiska gąszczy patentowych. Artykuł wskazuje ponadto możliwe sposoby ograniczania negatywnych skutków dominującej formy ochrony własności intelektualnej, które są możliwe do zastosowania na poziomie poszczególnych przedsiębiorstw.

Słowa kluczowe: własność intelektualna, patenty na oprogramowanie, zarządzanie zasobami niematerialnymi, prawo własności przemysłowej.