Costs and Benefits of Software Patents to Society

RISTO SARVAS

Abstract
This study identifies the costs and benefits of a patent system to the society in general and recognizes the difficulties related in assessing them. The focus is drawn to patenting of software inventions. The special characteristics of software in the context of patenting are identified, and then on the basis of these characteristics the special costs and benefits of a software patent system to the society are pointed out. The results of the study show that there are distinguishable characteristics of software that present some benefits and several costs. However, without quantitative evidence no definite conclusions can be made whether the costs of patenting software outweigh the benefits. Nevertheless, the efforts of last decades to make software inventions part of the patent system seem questionable, especially when incentive for software innovation has never been a problem.

1 Introduction
Patenting software and software implemented business methods is still a controversial issue in the software industry. Ever since computer programs were deemed patentable in the late 1970s in the U.S. and in Europe by the European Patent Office, the issue has remained somewhat unsolved. The controversy is between two polarized views. On the one hand, computer programs, i.e. software, are argued to be special kind of inventions that should not be protected by patents. Opposers of software patents argue that other forms of intellectual property protection, like copyright, trade secret and trademarks, are quite enough to give software inventions the protection they require. Software patents are seen as having certain characteristics that make them fundamentally different to other forms of technical inventions.

1 Researcher in Helsinki Institute for Information Technology, Helsinki University of Technology, P.O.Box 9800, 02015 TKK, Finland. A visiting scholar for the academic year 2002-2003 at School of Information Management & Systems, University of California, Berkeley, risto.sarvas@hiit.fi
2 In this article I use the concept 'software patents' in a broad sense. I consider all software, computer programs, and computer-implemented business methods as software. The special characteristics of software apply to all software whether it is patentable or not.
3 See Diamond v. Diehr. 450 U.S. 175.
4 See Vicom / Computer-related invention EPO Technical Board of Appeals' decision T 0208/84
On the other hand, some industry professionals think that software is as patentable as any other invention, and the problems encountered in patenting computer programs are something that will be smoothed out as time goes by: patent regulations, patent case law, and patenting practices only need to adapt to computer programs as patentable subject-matter.7

In a research funded by the European Commission8 these polarized views were studied. The people opposed to software patents were identified as students, academics, engineers and start-up companies. Their concerns were that software patents threaten the open source movement and small and medium size enterprises (SMEs) who are not familiar with patenting. The threat was seen to come from lack of patenting resources, expertise and information, in addition to fear of patent litigation.

According to the same study, the people who were in favour of software patenting were mostly lawyers, established industry players, and government agencies who were concerned about protecting R&D investments. Also, equality with the U.S. patent practice and opening up of global markets were major concerns for this group. However, both of these groups agreed that some changes must be done to patent regulation in Europe to take computer programs into better consideration.

Referring to the same study and other similar studies the European Commission published a Directive Proposal on the patentability of computer-related inventions.9 The objective of the Proposal was to harmonize the practice and interpretations of software related patent regulations among the member states. However, the Proposal failed to convince that patent protection for software is beneficial. The Proposal’s explanatory memorandum explains some of the problems and issues surrounding software patents but provides no evidence or convincing arguments in favour of patenting software. Rather, the explanatory memorandum faces the fact and states: ‘… any move to strengthen IP protection in the software industry cannot claim to rest on solid economic evidence’.10 The goal of this article is to underline this fact by pointing out the special characteristics of software inventions that should be taken into account when assessing the costs and benefits of software patent protection.11

An ideal patent system would benefit all stakeholders in technological development. In the last century the idea behind patenting has formed around a trade-off between the society and individual inventors: society, in the form of a patent office, grants the

---

10 Ibid at p.6.
inventor a limited monopoly to the inventor’s invention if the inventor agrees to make the invention public.

Society benefits from patenting by getting a public database of state-of-the-art technology in the form of public patent specifications, but loses in free markets by creating artificial monopolies due to exclusive rights to the patented invention. The inventor, on the other hand, has to disclose the details of the R&D work and results in the patent specifications, but gains a monopoly on the invention for twenty years. Also, potential inventors are given an incentive to innovate knowing that they will get a monopoly on the invention if a patent is granted, and that their research and innovation efforts will be hence rewarded.

The basic patent system trade-off naturally assumes that from the society’s perspective the benefits of patenting outweigh the costs. In other words, the society gains so much in having inventions published and inventors motivated that it is worth to grant the monopolies. For example, companies are more willing to invest in long-term expensive research if they know that there is a possibility to get a twenty-year monopoly on the end results.

This research paper outlines some of the issues surrounding the costs and benefits of a patent system. Special focus is given to software inventions and patents because they have special characteristics that change some fundamental assumptions about the patent system.12

2 The costs and benefits of a patent system

In this section, the costs and benefits of having a patent system are listed in more detail. It is good to bear in mind the history of patents. First patents were granted in the fifteenth and sixteenth century as special privileges to manufacturers and traders. For example, in England the Crown gave exclusive rights to practice trade or manufacture certain commodities, and for the monopoly rights the traders and manufacturers had to pay royalties to the Crown.13 Reaching the twentieth century the patent system had achieved worldwide popularity, almost every western country having its own system, and the principles that patents should be granted only to new, useful and non-obvious inventions had formed. However, some countries like the Netherlands and Switzerland abandoned their patent systems in the beginning of the 20th century for a while only to establish them again after a period of confusion.14

In other words, a patent system is very much an artificial system to promote innovation and diffusion of technology, and this system or its predecessors have been used in industrial countries for a few hundred years.

---

12 The issues surrounding patenting software were under much discussion in the European Union after the publication of the Directive Proposal in February 2002. However, many of the discussions, especially in Finland, focused on the differences between the Directive Proposal and European Patent Office practice. Much less attention was given to the more general question that are software patents beneficial or not.


2.1 The benefits of a patent system

The purpose of a patent system can be put very nobly. For example, the Finnish patent office states: “The patent system is a system for distributing information. It exists to promote the development of technology, the well being of the nation. To fulfil this purpose it is essential to provide the society with novel technological information.” The patent office continues: “[The patent database] has grown into a huge repository of information. The repository is a cross-section of world-wide research and development practices – a cross-section of research results, developed products, production processes, and their markets.”

In other words, the Finnish patent office, like many other patent offices, advertises the public benefits of patent systems. They underline the publicity of all patents, and how it provides the society a library of state-of-the-art technical inventions: anyone from individual inventors to large corporation R&D managers can search the patent database, and see the specifications of every patented invention. This advances the diffusion of technology in a society.

However, the primary reason for a patent system is not its catalytic effect on diffusion of technological information. The main reason to maintain such a system is encourage the making of inventions and the subsequent innovative work that will put those inventions to practical use. To put it simply, the more there are inventions and inventive activity the more there are ideas and technological advancement, thereby increasing the range of products and reducing the cost of products to society.

To encourage innovation, the state gives the inventor an economic monopoly for a limited time in return for a public description of the invention. The limits of the economic monopoly are stated in the each country’s patent law. The patent laws in industrial countries are very similar, and generally speaking the given monopoly consists of exclusive rights to sell, produce, import and use the patented invention. The rights are limited to the country or territory of the patent system, and a maximum time of the monopoly is usually twenty years.

In providing inventors a reward in the form of a twenty-year monopoly, patent systems encourage long-term expensive research. It is argued that long-term research in certain areas of technology would not be economically feasible without a patent system. Research in pharmaceutics is often given as an example of a field where the low costs of imitation makes long-term research unfeasible, unless the invention can be protected with patents. It is good to bear in mind, that unlike copyright, a patent prohibits even the use of inventions that were discovered independently of the patented invention.

In summary, the three main benefits of a patent system are diffusion of technology in the form of public patent databases; incentive for inventors to produce more inventions that

---


17 Especially, among computer programmers not familiar with intellectual property rights this difference between copyright and patent law can come as a surprise.
benefit the society in the form of better products; and thirdly, patenting provides a system that encourages long-term innovation that would not be feasible without patent protection, and therefore advances technological development.

2.2 The costs of a patent system
Above the patent system is explained as a trade-off between inventors and society. The inventors make their work public so that the society benefits from it, and the society gives the inventor a monopoly to the published invention. By giving the inventor a legal monopoly on the invention the society willingly bears the potential losses due to monopolistic exploitation of the invention. However, the cost that competition may suffer due to the monopolies created by patents can be lessened if these monopolies facilitate entry into the industry by new and innovative firms.

For the inventors, individuals and companies, the patent system poses extra costs. Applying for a patent costs thousands of euros, depending on the patent system. In addition to patent application costs, the granted patent has annual costs to keep it valid. The patent application and annual fees pay for the administrative costs of the patent office, therefore the patent office does not bear any extra costs to the society. Nevertheless, these direct costs for the patent owner are indirectly paid by the society: the costs of applying for a patent and the patent’s annual fees are manifested in higher prices of products.

Patent litigation costs can be significant, and a small enterprise may not be able to handle the costs involved in patent litigation, or even threats of litigation. In small and medium size companies patenting (or defensive action against patent claims) will take time from the same few people that are the creative brains behind the inventions. The same goes for patent searches and patent evaluations. Searching patents, applying for patents and reading patents requires special knowledge of patent law and patent strategies in addition to technical knowledge. Again, small and medium size enterprises may not have the resources and competence required to do all of these effectively.

Lastly, patents inhibit the invention of complementary and downstream products. A product, which is highly dependent on a patented invention, has extra costs due to the required licensing, and this might make the whole product economically unfeasible.

2.3 The net sum
Overall, assessing the benefits and costs of a patent system is very difficult. The general sources of costs and benefits can be identified, as done above, but placing a value on each cost or benefit has proved an extremely difficult question to answer. The

---

18 For example, the European Patent Office approximates the cost of an average patent in July 1999 to be 29 800 €. See ‘Cost of an average European patent’ available at http://www.european-patent-office.org/epo/new/kosten_e.pdf.
19 Private discussions with Mr. Eero Bomanson, National Board of Patents and Registration of Finland.
21 See Bakels and Hugenholtz, supra note 11.
relationships between the listed costs and benefits are complex and tend to change in time and across industries.\textsuperscript{22}

One major factor in favour of a patent system is the patent system itself. The costs of abandoning or even significantly changing the patent system would be very high. With harmonizing international treaties like the European Patent Convention (EPC), Trade-related aspects of intellectual property rights (TRIPS), and the Patent Cooperation Treaty (PCT) it is highly improbable that a single patent system, e.g. some national patent system will be dramatically changed, and even minor changes become less probable. Some studies have criticized the harmonization of patent regulations between countries because that has created a situation where it is very difficult for a single patent system to evolve and adapt.\textsuperscript{23}

Fritz Machlup is very often quoted in the context of the economics of a patent system: “If we did not have a patent system, it would be irresponsible, on the basis of our present knowledge of its economic consequences, to recommend instituting one. But since we have had a patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it.”\textsuperscript{24}

3 Special characteristics of a software patent system

Patent systems are artificial incentive systems tailor-made to provide the society with inventions and advance technology. The systems have been developed in the last couple of hundred years to suit the needs of the inventors and the public. When new fields of technology rise to become significant industries they turn to the patent system for intellectual property protection, and the patent system has to take these industries into account. This has happened more or less to biotechnology, pharmaceutics and information technology. They are all new and economically significant fields of technology that have not suited very well to the traditional concepts of patent regulation. For example, the basic types of inventions in patent systems are apparatus, processes and products/manufactures – how do these types fit into types of software inventions?\textsuperscript{25}

Computer programs are much younger than the patent system. First computer programs were written in the latter half of the twentieth century, although some mechanisms and methods prior to mid-1900s can be seen as computer programs. After the Second World War, computing machines developed and became more and more important economically and technologically. The increasing significance and dependence on computers reached a new level by the end of last century, and has shown no signs of slowing down.

\textsuperscript{25} A computer program invention can be claimed to be a method, a system, and a product. See, e.g. patent specification for EP0457112B1 ‘Asynchronous resynchronization of a commit procedure’ which is the patented invention in the EPO Board of Appeals’ decision T 1173/97 Computer program product / IBM
Especially, characteristics of software technology and software industry pose some significant problems to patent regulation. Patent regulations and practices are well suited for more traditional and older industries like mechanical industries or chemical processing. Some of the inherent characteristics of computer programs do not fit this framework, and some of the characteristics of the software industry are something not encountered in older industries.

3.1 Pure information goods

Computer programs are instructions for a computing machine, *i.e.* knowledge on how to make a computing machine do certain things. Other similar pure information goods are books, musical compositions, games, research discoveries, and audio and video productions. Typical for pure information goods is that the production of first copy is most of the costs and once the initial copy is made, the cost of subsequent copies is next to nothing. ²⁶ For example, the production costs of a movie are astronomical in relation to the costs of copying it digitally.

Pure information goods suffer from the so-called ‘public goods problem’. Public goods have two distinguished features: nonexcludability and nonrivalrous competition. The former means that it is very difficult to exclude those who do not pay for the good from consuming it. An example of this is the easy copying of digital music or computer programs. The latter characteristic, nonrivalrous competition of pure information goods is that additional consumers of the good do not deplete the supply. In other words, the same information can be sold over and over again to other people without any additional costs. For example, downloading an html-page from the Internet does not limit the number of possible later downloads at all. ²⁷

Another characteristic of pure information goods, especially computer software, databases, scientific knowledge, and research tools is that it is very difficult to distinguish the good from the knowledge required to produce it. In conventional goods the knowledge is embodied in the commercialised product, and it can be attained by reverse engineering or by reading patent specifications. In pure information goods like computer programs the product itself is the knowledge that will be used as an input to the creation of future products. In practice however, computer programs or databases are usually not merely information but the products include convenient packaging, documentation and services. ²⁸

Also, many of pure information goods are characteristically produced by creative individuals using very few inputs other than their labour. Hall ²⁹ argues that for many of the producers of such goods, the utility and the fame from invention and creation is its own reward, and requires little incentive from systems like patenting. These kinds of

---

²⁷ See Menell, *supra* note 16.
products, like academic papers or open-source software, are likely to be produced even in the absence of patent protection.

Patent systems have in the past dealt with pure information goods as unpatentable subject matter. For example, European Patent Convention from 1973 explicitly states that inventions that are mathematical methods, scientific theories, discoveries, aesthetic creations, schemes or rules for playing games or doing business, presentations of information or computer programs, are not patentable as such.\textsuperscript{30} This list of unpatentable subject matter is similar in all patent systems, for example, the U.S. case law states that mental processes, scientific principles, laws of nature, and mathematical algorithms are not patentable.\textsuperscript{31}

Therefore, it can be said that prior to the rise of information technology in the last decades pure information goods were considered to be outside the scope of patentable inventions, and traditionally their form of intellectual property protection has been copyright.\textsuperscript{32}

3.2 Other special characteristics of software

In addition to the characteristics of pure information goods, software has some other attributes that are worth pointing out.

The first characteristic is the high interdependency of computer programs. Software products are a hierarchy of computer programs and functions that are previous inventions themselves. Large software systems are a network of computer programs that can all be reduced to single instructions. It is very typical of software production that a second-generation invention builds on the previous invention, which has been built on earlier inventions and so on. Therefore, secondary invention in software technology is as important as first.\textsuperscript{33} Also, the density of inventions in a single product is very high, for example a mobile phone that is a mixture of hardware and software inventions can contain hundreds of thousands of inventions.

Product development of software can be very rapid when compared to other fields of technology. With almost no production and the low cost of producing copies, an idea can be made into a commercialised product in a year or two. Especially when compared to industries like pharmaceutics or biotechnology, software products’ lifecycles are much shorter, and some of the products become obsolete in a very short time frame.\textsuperscript{34}

\textsuperscript{30} See European Patent Convention Article 52(2) and 52(3).
\textsuperscript{31} See Menell, supra note 16.
\textsuperscript{32} See van den Berg 'Patentability of Computer-Software-Related Inventions' The Law and Practice of the Enlarged Board of Appeal of the European Patent Office during its first ten years (Carl Heymanns Verlag: Köln 1996) p.31 ‘Software seemed to be something outside the real world of engineering. --- It also seems not to have been generally recognized at the time how technically important software would become, and in particular that it could – as indeed later did – replace hardware to carry out inventions which were susceptible of industrial application. Software was considered as something abstract and weird which did not accomplish anything which in conventional terms would have been regarded as worth protecting under patent law.’
\textsuperscript{33} See Menell, supra note 16.
\textsuperscript{34} See Bakels and Hugenholtz, supra note 11.
3.3 The special benefits of a software patent system

Above are mentioned the special characteristics of software that differ from more traditional inventions that the patent system is designed for. As patenting software has become less and less restricted by regulation and practice it has provided some benefits to the society. Once again, the benefits presented here and the costs presented in the following section are not empirically measured – they are only identified. Just like assessing the costs and benefits of the patent system in general, it is very difficult to quantitatively assess patenting benefits even for a single field of technology.

As mentioned above, once the pure information good is produced for the first time, it can be reproduced with practically no cost. Therefore, information goods are easily copied and imitated without paying any reward to the inventor. This can be argued to be a loss to the society in general, because inventors have little incentive to produce inventions that are easily copied or imitated. This problem is the same as the public goods characteristic of nonexcludability mentioned earlier.

The patent system takes care of this problem, and provides the inventors with an incentive. Once a patent is granted the inventor has rights to prevent any use, importing, and production of the patented invention. The patent holder’s rights to the invention last for maximum of twenty years. The society benefits from getting more innovative action in information goods, as inventors have strong incentives to make inventions.  

Another benefit of the patent system relates to the software business’ dependence on high upfront investments: investors look for any signal that the firm has a potentially rent-generating asset, and obtaining patents or at least pending patents can be of significant importance. In a business, like software business, where an idea can be the only thing a firm possesses, realizing that idea into a patent can be crucial. This possibility for small companies to enter the market more easily is beneficial to the society.  

In summary, the benefits of a patent system in software development is that it solves the public good problem of getting all users of an invention to pay a reward to the inventor thus creating an incentive for further innovation. Also, patents can be very helpful in acquiring investment capital for small companies.  

3.4 The special costs of a software patent system

Many of the special characteristics of software cause extra costs in patenting. Some of the costs are due to the fundamental characteristics of software as information, some are typical of software development and production, and some are due to the patenting process itself.

One of the basic arguments against patenting software is that computer programs are pure information, and they should not be given patent protection. This is quite the
opposite argument to the fact that patenting solves some of the public goods problems inherent to pure information goods. One reason for opposing patents on software is the fact that patents on other information goods are also forbidden: as mentioned above, patents can not be applied for discoveries, mathematical methods, scientific theories and presentations of information as such. The grounds for excluding these items are difficult to present unambiguously, because the line drawn between patentable and non-patentable subject matter is a matter of definition.\textsuperscript{38} What can and cannot be patented is a matter of industrial economics rather than borders defined by common sense on what kind of inventions are inherently suitable for patenting. In other words, the answer to the question of software patentability should be the outcome of assessing its economical benefits to the society, but as this paper points out the assessment is an extremely difficult task.

The cumulative nature of software inventions and the high density of inventions per product is a problem in the context of patenting. Searching for all possible patent infringements for even a small software product is a difficult task requiring some special training.\textsuperscript{39} Even if all the patented inventions in a single product were identified, the cumulative licensing fees for all those patents could render the whole product unprofitable.\textsuperscript{40} This has caused that it is more than common for individual and open source programmers to ignore patents altogether to avoid purposeful infringement accusations and not to use their time in searching for patents whose rights they cannot afford to respect.\textsuperscript{41} This is an example of how the patent system is not working at all: the inventors do not get their reward because their patent rights are ignored, and on the other hand the patents inhibit innovation because the amount of patented inventions is too high for anyone to pay for all licenses.

In products that are pure information the difference between the product and the knowledge to produce it is almost nonexistent. In more traditional products, the knowledge to produce the product is embedded into the product itself, and it must be deducted or reverse engineered to separate it.\textsuperscript{42} In information products like software the knowledge is in the computer programming code, which is the product. This means that the use of the knowledge in traditional products is not restricted unless a product is produced or a process implemented. In information products, the mere use of the knowledge is using the product itself, and limited by patent holder’s rights. It has not been measured what could be the effects of this difference and how it affects innovation.

\textsuperscript{38} An example of this is how the European Patent Office’s Board of Appeals’ definitions concerning computer program patentability in relation to the concepts of \textit{technical character} and \textit{technical contribution} have changed in the last two decades. See, e.g. Sarvas ‘A software engineering view to the decisions of the European Patent Office concerning computer programs’ (2001) Master’s thesis, Helsinki University of Technology, available at \url{http://www.hut.fi/~rsarvas/Risto_Sarvas_thesis.pdf}.

\textsuperscript{39} See Bakels and Hugenholtz, \textit{supra} note 11.

\textsuperscript{40} For example, if a software engineer identifies 101 patented inventions in her product, and each patent holder requires one percent in royalties it would create a cumulative royalty rate of 101%. To remedy this problem, see \textit{e.g.} Nokia Press Release on May 8th 2002 suggesting industry-wide commitment to 5% cumulative IPR royalty for WCDMA technology thus lowering the barrier for market entry. \url{http://press.nokia.com/PR/200205/858681_5.html}.

\textsuperscript{41} Linus Torvalds, the inventor of Linux operating system and a public figure in the programmer community has commented: ‘I do not look up any patents on _principle_ because (a) it’s a horrible waste of time and (b) I don’t want to know. The fact is, technical people are better off not looking at patents. If you don’t know what they cover and where they are, you won’t be knowingly infringing on them.’

\textsuperscript{42} See Hall, \textit{supra} note 26.
One of the main grounds for arguing that the patent system is not required to provide incentive for software innovation is in the history of information technology. Software development has its roots in the academic world, which is an example of innovation without patenting.\textsuperscript{43} Also, some court decisions in the 1970s ruled computer programs outside the scope of patentability,\textsuperscript{44} and patent protection of software was not an option until the 1980s when a series of another court decisions made it possible.\textsuperscript{45} Nevertheless, software development thrived for many years absent of patent protection, and even today, whether legally possible or not, most software innovation is not actually being protected by patents.\textsuperscript{46} Therefore, it has been demonstrated that patenting is not essential to innovation in software.

Another example of software innovation without patents is open source development. At the time when Linux operating system, a popular example of open source software, was initially created patent protection was available. Nevertheless, Linux has grown into one of the world’s most used operating systems praised for its quality and flexibility without any patents to protect its inventions. However, it is highly probable that Linux infringes on several patents, because the open source developers hardly ever consult patent databases and the amount of innovation in Linux is high. Open source software is also especially vulnerable to accusations of patent infringement because its source code is public unlike the source code of proprietary software.

Open source software and the history of software development show that incentive for innovation in the absence of patenting seems not to be a problem in software. Therefore, the society can be seen to uphold an incentive system that is not required, and from the perspective of open source development it is a system that actually produces negative incentives for innovation.

Due to the relatively young age of software and the late adoption of software patents the patent offices have not been able to maintain high quality in software patents. The novelty and inventiveness of software patents is a problem still today. Patent offices maintain their knowledge on the state-of-the-art of a given field of technology mainly by searching patent databases. If the patent application does not refer to a granted patent it is very difficult for a patent examiner to assess novelty or inventiveness from other literature than patents. Also, because software patenting started much later than software development, the state-of-the-art was not in the patent databases. The result was, and still is, that the patent offices grant so-called bad patents, \textit{i.e.} trivial and old inventions get patent protection. This problem might be remedied in time when the patent databases get more and more software patents, but this assumes that enough state-of-the-art software technology is patented which is not true at the time.\textsuperscript{47}

\begin{footnotesize}
\textsuperscript{43} Scientists and universities do apply for patents and own patents, but the main channel of knowledge diffusion is the ‘open’ scientific research community.
\textsuperscript{45} See Hall, supra note 26.
\textsuperscript{46} See Bakels and Hugenholtz, supra note 11.
\textsuperscript{47} See, \textit{e.g.} Merges ‘As Many as Six Impossible Patents Before Breakfast: Property Rights for Business Concepts and Patent System Reform’ (1999) \textit{Berkeley Law Technology Journal} 14; Aharonian ‘Why many invalid patents are being issued by the Patent Office’ available at \url{http://www.bustpatents.com/main.htm#INV}
\end{footnotesize}
In relation to the rapidly developing information technology the patenting process itself is slow. From the date of application it takes a minimum of eighteen months for the application to become public. In Europe, the practice is that an application is published after eighteen months for anyone to oppose it. In the U.S. the application can be held secret until the patent is granted which takes about two to three years. However, the patent holder’s rights start from the application date, and the invention also becomes prior art for the patent office, although it is not published. This means that all the public patent databases are at least eighteen months old, and in the U.S. the information can be even older. In software business eighteen months can be enough time to implement an idea into a product - just in time to find out that someone else has patented it, and it was impossible to acquire this information earlier. Therefore, in addition to granting noticeable amounts of bad patents, the patent system is working in a too slow pace for software inventors to take it into account.

In summary, several of the costs to the public come from the non-functionality of the patenting process: patent offices grant low quality patents; the patent application process is too slow for professionals to follow state-of-the-art patents; and the patent databases are rarely searched for any other than legal purposes. These faults or dysfunctions of the patent system weaken the very basic principles of patenting, namely presenting a public database of state-of-the-art technology for inventors to use as a source of information and means of protecting innovation. Especially, when it has been shown that innovation in software has been, and still is, very active without patenting.

4 Conclusions

The simple question remains: is patenting software beneficial or not. Above are arguments for and against software patents, but no hard empirical data is available to really assess the importance of each factor, and therefore drawing any conclusions in favour or against software patenting should be done with caution.

Patent regulations are adapting to take the special characteristics of software into account. As mentioned previously the European Commission published a directive proposal in February 2002 on software patents that tries to deal with some of these problems. Also, the U.S. Patent and Trademark Office and the European Patent Office have adapted their practices to take software better into account. But these adaptations are minor changes to the existing patent system. The system still assumes that software is patentable and software patents benefit the public and the inventors. The system assumes that the problems involved are just problems of adapting the patent system to a new technology.

---

48 See European Patent Convention Article 93.
50 For example, anecdotal rules of thumb like ‘Moore’s law’ and ‘Parkinson’s law of data’ demonstrate the fast pace of development in information technology. Moore’s law predicts that the number of transistors in a computer processor (CPU) doubles every 18 months and Parkinson’s law states that the memory usage of evolving systems doubles also every 18 months.
51 See Bakels and Hugenholtz, supra note 11.
However, as discussed above, it is common knowledge that the benefits of a patent system are anything but unambiguous, and one of the reasons having a patent system is that getting rid of it is too expensive. Therefore, the big effort of last twenty years to make software inventions part of the patent system seems questionable, especially when incentive for software innovation has never been a problem.