ROLE OF INTELLECTUAL PROPERTY RIGHTS IN SOFTWARE

Sheenam[1], Sarbjit Kaur[2], Sonika[3]

Abstract

Computer software industry and Information technology is playing a key role in the development of the economic growth of a country. Software technology is evolving faster and research in software is progressing rapidly. Free software formats are moving fast in most developing countries where users are attempting to develop new products, innovations and adaptations. As software systems are deployed into real-world use, legal issues of intellectual property rights have to be protected. The purpose of this paper is to explore the current state of software protection and to develop a conceptual framework of legality concerns of intellectual property rights in software industry.

Keywords**:** Software industry, IPRs, I.T., Protection

I. Introduction

Intellectual property (IP) is a term referring to a number of distinct types of creations of the mind for which a set of exclusive rights are recognized under the corresponding fields of law. Under intellectual property law, owners are granted certain exclusive rights to a variety of intangible assets, such as musical, literary, and artistic works; discoveries and inventions; and words, phrases, symbols, and designs.

Intellectual property stems from the concept of assigning people property rights to their creations of intellect. These rights are very similar to rights in real property, such as real estate. These rights include the ability to buy, sell and license their property, the ability to use the property themselves, and the rights to prevent others from using the property without permission. Intellectual property rights are the building blocks for managing intellectual property they are a collection of registrable and unregistrable rights, which have different but sometimes overlapping uses intellectual activity in the industrial, scientific, literary, and artistic fields. Common types of intellectual property rights include copyrights, trademarks, patents and trade secrets.

A. Patents

Patents are one of the oldest forms of IP protection. Patents are legal titles granting the owner the exclusive right to make commercial use of an invention. To qualify for patent protection, inventions must be new, non-obvious, and commercially applicable. The patent system started in the 1700s. The aim of the patent system is to encourage economic and technological development by rewarding intellectual creativity. A patent is a statutory right granted for a limited period to an inventor in respect of an invention to exclude any other person from manufacturing, using or selling the patented product or from using the patented process, without due permission. The term of protection is usually limited to 20 years, after which the invention moves into public domain. Since the early 1980s, patents have also been granted for agricultural biotechnology products and processes and for certain aspects of computer software. The total numbers of the patents granted from year 2007 to 2010 is shown in table 1. (WIPO Statistics Database, October 2011).

[1,2,3] PGGCG, Sector 42, Chandigarh. (Affiliated to Punjab University, Chandigarh)
[1] sheenammadan@rediffmail.com, [2] sarbjitkaur1981@gmail.com, [3] sonikahi2000@gmail.com

Table 1: Total numbers of the patents granted from year 2007 to 2010

Year	2007	2008	2009	2010
Total Patent Grants	770,309	772,398	808,270	908,862

The organizations with the most patents granted in year 2010 are also shown in table 2.

Table 2: Most Patents granted in Year 2010

Rank	Organization	Patents
1	IBM	5866
2	Samsung	4518
3	Microsoft	3086
4	Canon	2551
5	Panasonic	2443

B. Copyrights

Copyrights are exclusive rights granted to the creators of literary, dramatic, musical, artistic works and producers of cinematographic films and sound recordings. Copyright protection arises automatically upon the creation of an original work of authorship. There is no need to "apply" for a copyright or register the copyrighted work in order for protection to exist. Generally, the duration of a copyright is the author's life plus fifty years. In the case of software created by an employee in the course of his or her employment, the resulting "work made for hire" would be protected by copyright law for seventy-five years from publication.

C. Trade secrets

Trade secrets are information that companies keep secret to give them an advantage over their competitors. The formula for Coca-Cola is the most famous trade secret. Trade secrets are not protected by intellectual property law the same way that trademarks or patents are. Protection for trade secrets is done by non-disclosure; the information must be kept confidential. A trade secret can be a formula pattern device or other information which is used in business and gives an opportunity to obtain an advantage over competitors.

D. Trade mark

A trademark is a distinctive sign or indicator, used by an individual, business organization, or other legal entity, to identify that the products or services with which the trademark appears originate from a unique source, and to distinguish its products or services from those of other entities. A trademark is typically a name, word, phrase, logo, symbol, design, image, or a combination of these elements. There is also a range of non-conventional trademarks comprising marks which do not fall into these standard categories, such as those based on colour, smell, or sound.

A trademark may be designated by the following symbols:

™ (for an unregistered trade mark, that is, a mark used to promote or brand goods)

□ (for an unregistered service mark, that is, a mark used to promote or brand services)

® (for a registered trademark or service mark)

II. Intellectual Property and Software

Over the past few decades the software industry has experienced tense changes and today it is one of the biggest and fastest growing industries in the world. Software is now a widely available product for all consumers who desire it and exists in almost every place you can think of. It is used in your computer that you use every day, in your phone, mp3 player, even your car has software. Software defines the qualities and functionalities of many products and services. Much of the software that is developed today is solely used for increasing efficiency, functionality and quality of production and distribution processes.

Software works can be put into three levels: the instructions, the execution and the computing results. The computer programmer writes the instructions which are called the source code. The source code cannot be read directly by a computer; therefore a compiler is needed to translate it into computer-readable code, which is called object code. The operating system then executes the instructions defined in the code to produce the computing result.

At this point there will be the existence of two kinds of users. Some users are only interested in the software for its computing results, on the other hand there also exist users who want access to the source code so they can read it, copy it and modify it. For these reasons, most commercial software is sold without the source code. They only offer the object code, in other words, denying other software programmers to see the instructions that make the software perform in the way it does. The source code is the program‗s history and is very valuable for the people behind it; it often contains comments like for example which solutions were tried and failed and how problems were solved .In the end, the source code is the key focal point of any software program and that is why it is in the centre of the debate on software protection.

A. History of software property rights

In the early history of software development, the courts took the position that patents did not apply to software. The courts' view was that computers simply used mathematical algorithms, which if viewed as mental processes are not patentable (Gottschalk v. Benson 409 U.S.63 (1972); in this case a method for converting BCD numerals into binary numerals was found not patentable. Therefore the only way to protect computer code was with trade secret laws and copyright .For trade secret laws to protect your invention, you must proactively work to keep it secret. Further, your invention needs to be something not easily discoverable. Therefore, most trade secrets tend to be production methods, where competitors cannot fathom from the end product how it was made. In contrast, computer executable code is fairly easy to disassemble and comprehend. Therefore, software developers have rarely been able enforce trade secret protection. More often they sought to protect their code with copyright law. However, copyright only protected the written expression. It did not protect any of the underlying ideas encompassed by the software. Anyone could disassemble your code, see how it worked, and then write their own code to perform the same function as your code. As time passed, people tried to stretch copyright protection to cover more than simply the written expression. This led to several of the "look and feel" cases, where courts decided that copyright protected not only the code, but the "structure, sequence, and organization" of the program.

B. Role of IPR in Software

Software interoperability has become a really important issue in a rapid interconnected world. The result of this is that IPR has been given an even more significant role in software and standardization, as technological developments have made piracy and patent infringement easier. For the software industry to work properly, in economic terms, it highly depends on legal rules which protects from free riding and duplication of products. It is obvious that this is important when there is such a huge difference in costs between developing and (illegal) copying software products. The costs for copying a program are minimal while the costs relating to the creation of a software program can be extensive. Therefore if the developers want their invested capital back it‘s important that free riding is kept in control .

C. Software patent

Software inventions have not only helped to automate the various manual processes with accuracy but also eliminate the need for storing paper records making retrieval efficient and faster. For instance, the invention of banking software eliminated the need for manual accounting of cash-flows. Likewise, the invention of interconnection of computers to form a network initially developed as 'ARPANET' and later evolved as Internet. The Internet in turn enabled the development of World Wide Web for exchange and publishing of documents. Software inventions can be filed under different verticals of Information Technology. These verticals include 'E-Commerce patents' for protecting a method of buying or selling commodities (including goods and services) over the Internet; 'Internet patents' for inventions that involve applications based on the Internet and can cover methods of transmitting information over the Internet, data compression techniques and encryption methods.

D. Software copyright

While a patent can protect the novel ideas embodied in a software program, a copyright cannot. Copyright protection extends to the particular form in which an idea is expressed. copyright in this case in computer software not only represents the exercise of power in the present circumstance, it also represents power relations far into the future and will truly monumental effects on their economic futures for decades, as well as their future use of software. Incoming decades, it will become the dead hand of the past controlling their futures. Copyrights also extend to source codes of softwares which prevent competitors from copying source codes developed by a company or individual(s). Copyright covers website content, product catalogues and brochures, drawings, artistic lables and pictures as well. Unlike other forms of IPR, copyrights are known as 'Bundle of Rights' as they give an individual a variety of rights, by virtue of which they can enable others to reproduce their work, issue copies to the public, perform in public, communicate to the public, make an adaptation, make a translation or make cinematographic film/sound recording. Another recent development in Copyright is Copyleft which is a form of licensing used to maintain copyright conditions for works such as computer software, documents, music and art which go through multiple levels of refinements and are available in different versions. Copyleft gives the creators the right to distribute copies of modified versions of a work preserving and giving the same rights as that of the original work. Copyleft has been extremely helpful, particularly in the field of software for open projects such as Linux as it has promoted development of quality software and has given the developers the assurance that his/her work will be recognized in the modified versions of the software. Thus, copyright gives the creators a boost to develop and protect their work and hamper the copying of copyrightable contents. Additionally, copyrights assure the creators that they will be entitled to receive royalty for their creations.

E. Software Tradesecret

A trade secret can theoretically last forever -- for as long as its owner uses reasonable efforts to keep it secret and someone else doesn't independently create or "discover" it. Many features of software, such as code and the ideas and concepts reflected in it, can be protected as trade secrets. This protection lasts as long as the protected element retains its trade secret status. Unlike patents, trade secret protection will not extend to elements of software that are readily ascertainable by lawful means, such as reverse engineering or independent development. Trade secrets are not subject to being "infringed," as with patents and copyrights, but are subject to theft. Their legal status as a protectable intellectual property right will be upheld if the owner can prove the trade secret was not generally known and reasonable steps were taken to preserve its secrecy. Maximizing the economic value of a software asset critically depends on understanding the nature of the intellectual property rights involved and how best to use the available forms of legal protection to protect those rights.

III. Importance of protection of software

The protection will vary depending on the type of software, investments and the usage and constraints of the software. Today software is traditionally protected by copyrights, trade secrets and patents (while trademarks are only used to protect a word, name, symbol or most commonly a brand-name). This has led to a number of challenges because the source code of the published software can easily be rewritten.

Trade secrets are used to protect valuable information in a program, but since programming languages and operating systems became more standardized in the 1980s, an increased possibility to understand the source code of a program became evident and trade secrets could no longer be used alone to protect the source code, therefore companies turned to copyright.

A. Copyright Protection and Restriction

Copyright is meant to protect direct copying, but the copying of the idea or concept is still allowed. Said from a more legal point of view, copyright is a contract between an author and a user which specifies what the user can or cannot do with the author‘s work.

Beginning in the early 1980s, a number of governments in the developed world decided, after extensive lobbying by some (though not all) sections of the software industry, that computer software was analogous to the traditional copyright category of an original literary work of authorship and hence should be protected as a literary copyright. The essential argument was (and still is) that the thousands or even millions of lines of binary code found in a program, the series of instructions (that is, the symbols .O. and .1. found in infinite patterns in an object code) which are given to the computer hardware can be best understood, as a matter of legal classification

B. Patent protection and Restriction

Patents are similar to copyright; it does not protect the underlying idea, but the new technical application of an idea. Furthermore, a patent gives the creator of the product or process, a monopoly right which can last for 20 years. The right to exclude has made patents very effective.

C. Trade Secret Protection and Restriction

Trade secret law operates on a different basis than the copyright and patent protection offered to software .Rather than focusing solely on expression or demanding novelty as a prerequisite to protection, the law of trade secrets will protect the ideas underlying particular software, including the software’s structure or architecture and organization, and various features, routines and processes within the software, novel or not, so long as those ideas are not generally known (or readily ascertainable from the marketed software) and give, or have the potential to give, a competitive advantage by virtue of the fact that others do not know them.

The mixed nature of software makes the protection by IPR complex. The basis of software, which is the source code, is looked upon as an expression of an idea. The developers are given copyright for the source code of the software. When the source code is put to life in compiled form, the software performs technical work and can be defined as an engineering solution. Actually this was the statement in favour of applying the patent system to protect software technologies. In the United States and the European Union, this mixed nature of software is another reason for the great number of ad hoc decisions made in cases of IPR in software technologies.

IV. Difference between software life cycle and the patent-based software life cycle:

A. The software life cycle:

In Software engineering, the software life cycle is a frequently used manner of organizing the software development process. Figure [1] shows a strongly simplified version of the life cycle taken from a standard textbook. It consists of the following phases:

- **Requirements engineering**: Collect the requirements and expectations from the future owners and users of the system.
- **Design:** Translate the requirements in a specification that describes the global architecture and the functionality of the system.
- **Implementation:** Build the system, and transforming the design into software source code.
- **Testing**: Test that the implemented system conforms to the specification.
- **Maintenance**: Install, maintain and gradually improve the system.

It should be emphasized that the software life cycle covers design and construction of a software product as well as its use. Each phase contains a Validation and Verification (V&V) sub-phase in which the qualities of the deliverables of that phase are controlled. Also note the backward arrows that make this into a real “cycle”: it is possible to discover in later phases that decisions made in a previous phase have to be revised.

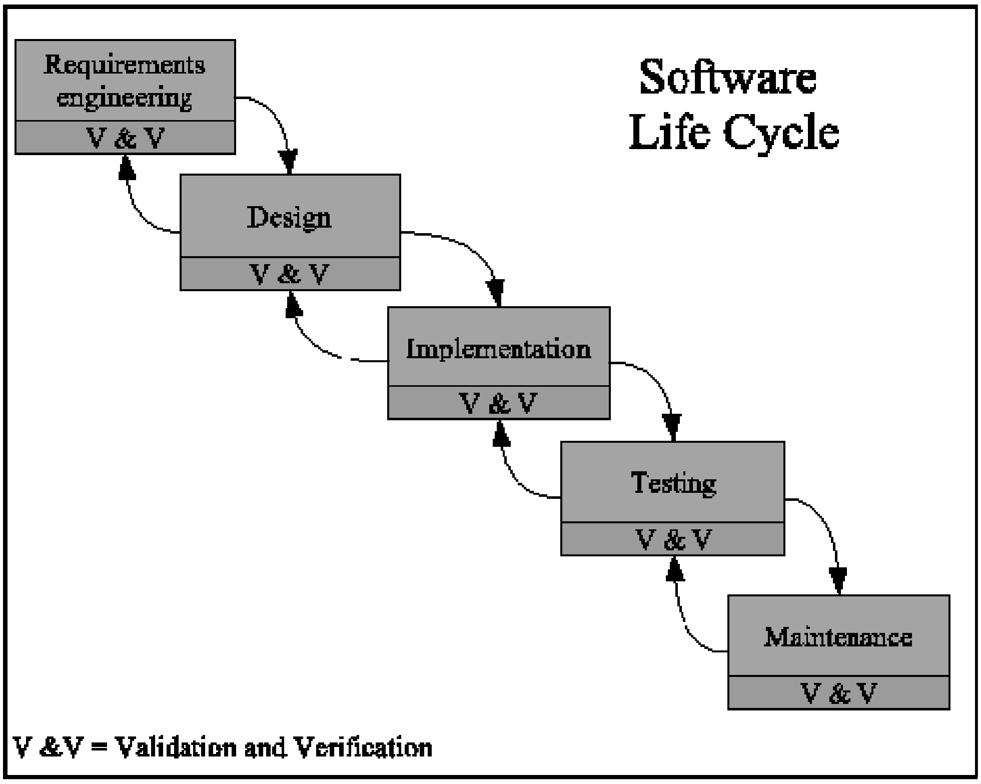

Figure 1: Phases of Software Life Cycle

We will now proceed in three steps. First, a defensive Patent-aware Software Life Cycle is sketched that ensures that the software development organization does not infringe patents of third parties. Next, a more offensive Patent-based Software Life Cycle is described that also considers the options to file patent applications for knowledge that has been generated in each phase of the life cycle. Finally, the IPR-based Software Life Cycle extends the previous one to all IPR options: secrecy, copyrights and patents.

B. The Patent-based Software Life Cycle

It is, however, possible to go one step further. In Figure 2, we sketch a Patent-based Software Life Cycle in which yet another sub-phase has been added that performs patent applications whenever possible. We conjecture that this strategy is only available to the software development organizations with the deepest pockets. For each phase now further questions apply, such as

- Does this phase generate patentable knowledge?
- Should we file a patent application for this knowledge?
- Are there other means to avoid that this knowledge generates an advantage for our competitors?

In many large software development organizations there exist “Chinese walls” between software developers and patent attorneys. This is not only the case for large commercial organizations but also for large open source projects like the Apache Foundation. The rationale being that the less software developers know about patents the stronger the position of the organization is in legal disputes. Implementation of the Patent-based Software Life Cycle may require similar measures. Of course, such measures completely defeat one of the primary goals of the patent system, i.e., knowledge dissemination.

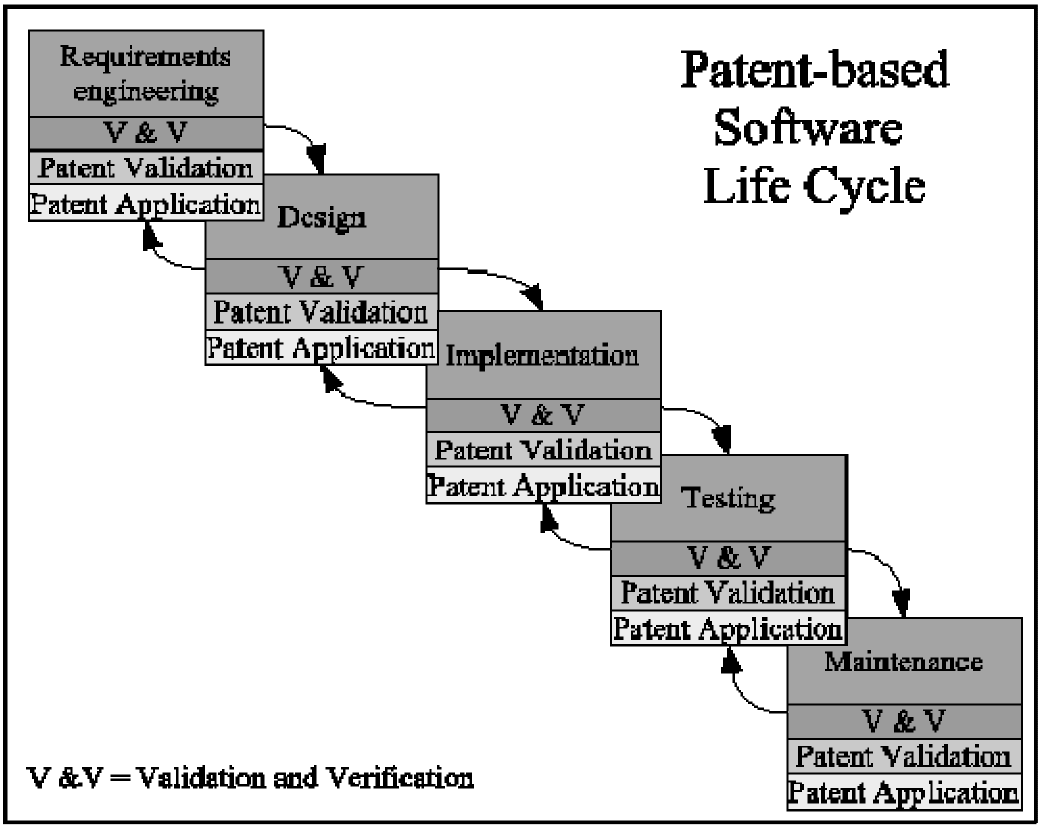

Figure 2: Patent based Software Life Cycle

V. Conclusion

Engineers have always known that innovation is important, and that patents protect their inventions. Software engineers know that the programs they write automatically protected by copyright. As Software engineering systems convert software production from an art to an engineering discipline. Software programs are different than other electromechanical devices because they are designed solely in terms of their function. The inventor of a typical electromechanical device must design new physical features to qualify for a patent. On the other hand, a software developer need only design new functions to create a working embodiment of the program. Patents have been used for years now in the software industry as a blunt weapon to suppress innovation, kill competition, and generate undeserved royalties. It is time to revisit the value of patents as they relate to software and test some of the policy reasons for awarding patents in the software context. Perhaps, we have lost sight of the true meaning and purpose of patents. A reason for awarding patents and the ensuing monopoly is that "without the possibility of patent protection, many people might not take the risk of investing the time or money necessary to create or perfect new products. On the other hand in case of open source software for example, the freely available operating system Linux, has reached a level of sophistication putting it at least on a par with the quality of its proprietary rivals by Microsoft, Sun and other providers. Interest in open source is growing rapidly in industry and government agencies in response to the demonstrable cost benefits and presumed advantages in terms of stability and security. Innovators have to step out of the fear and adopt the provisions of IPR to promote growth and economic development and benefit the mankind and in return reap incentives and recognition for their work.

Bibliography

Bergstra, J.A. and Klint, P., About 'Trival' Software Patents: The Isnot Case, *Science of Computer Programming*, 16 (3) (2007) 264-285.

Bessen, J. & Hunt, R.M., 2003. An Empirical Look at Software Patents, *Federal Reserve Bank of Philadelphia Working Paper*, 03-17(2003).

Cundiff, V.A., 1998. *Protecting Computer Software as a Trade Secret*. New York: Practicing Law Institute.

Festa, P., 2001. Governments Push Open Source Software. CNET News [Online] 29 August. Available <http://news.cnet.com/2100-1001-272299.html> [Accessed 12 January 2012].

Michel, N. J., 2006. The Impact of Digital File Sharing on the Music Industry: An Empirical Analysis. *Topics in Economic Analysis & Policy*, Vol. 6, No. 1, Article 18.

Potter, S.W., 2000. Opening up to Open Source. *Richmond Journal of Law and Technology*, 6(5) 24.

Story, A., 2002. *Copyright, Software and the Internet Copyright*. Study Paper 5, Commission on Intellectual Property Rights, London.