

## **PART IV: CHAPTER 3.5**

### **Protecting and Commercializing University Developed Software**

**By**

**Steven W. Lundberg**, Partner

and

**Michael G. Smith**, Associate

Schwegman, Lundberg, Woessner & Kluth, P.A.

Minneapolis, MN 55402

#### **3.5.1 COMMERCIALIZING UNIVERSITY DEVELOPED SOFTWARE**

Software developed in the university environment can be grouped in three categories:

- a. Business method “applications” software;
- b. “Component” software; and
- c. “Technological” software.

##### **3.5.1.A. Business Method Applications Software**

Most of the software developed at colleges and universities is not the object of research projects but rather is written for use in teaching, administration or for some other “non-technological” purpose. Educational software probably constitutes the largest portion of business method applications software. These developments are usually the most refined and easiest to license of all academic software developments. On the administrative side, a professor or graduate student may develop a database application for storing and retrieving exam questions or student grades, for tracking departmental budgets, or for planning and managing research projects. In software parlance, programs of this type are termed “applications,” and are usually written to run on standard computer platforms such as the Macintosh, SUN workstations, or IBM-compatible personal computers. If there ever was any question as to the patentability of this type of software, the

1998 decision in *State Street Bank & Trust Co. v. Signature Financial Group Inc.* has laid that question to rest. The State Street decision ruled that the “business method” exception to patentable subject matter is invalid. Therefore, a patent for a “business method” shares the same validity as a patent for a process, machine, manufacture, or composition of matter.

### **3.5.1.B. Component Software**

Another class of software can be termed “component.” Component software forms part of a technological device or system that is developed as an object of a research project, but is not the focus of the project itself. For example, an inventive new heart monitoring device, imaging system, guidance system or solar collection system would likely contain such software as an element or “component” of the overall system. Other examples of component software would likely be found in robotic systems wherein the software components include algorithms for use in controlling robot action or movement. Component software may also be found in a diagnostic method, which may, for instance, include a step that calls for software automated analysis of a tissue sample or the like. In all of these cases, the software is not the object of the development or invention, but rather a component of a larger system, device or method. In today’s technology, component software is commonplace, and will probably appear in most electronic/computer/medical devices and systems.

### **3.5.1.C. Technological Software**

The final category of software can be termed “technologic” software. In this category, the software is of a technologic nature, the actual object of technical innovation. For example, software in this category includes operating systems, network software, artificial intelligence, expert systems, memory management programs, electronic spreadsheets, computer languages, computer-aided design systems, language translation programs, database systems, word processing software, electronic mail software, data compression utilities, or user-interfaces. Technological software also differs from business method applications software in that it tends to be experimental in nature, as opposed to usable “productized” software.

### **3.5.1.D Commercializing Applications Software**

The greatest commercial value of business method applications software will usually reside in the actual program itself, and the associated programming know-

how, as opposed to the more generalized or abstract process, method or system that is used in the software. In fact, in most cases, applications software will be modeled after known approaches or techniques and will not be inventive in the patent sense. Nonetheless; new, useful, and non-obvious business methods are patentable whether or not embodied in software.

The main value of applications software to potential licensees is the right to refine or upgrade the licensed software into a commercially marketable product. Alternatively, a licensee may seek only to use the software for its internal operations or as a base for a further development.

The nature of applications software and its value to potential licensees makes it suitable for a combination copyright/trade secret licensing approach. Under this approach, the primary licensed right is the copyright in the program. The source code for the program or other non-public aspects of the software can be licensed as trade secrets along with the copyright. In essence, this is the standard software licensing approach used in and by the software industry.

Frequently, applications software will be sufficiently novel or inventive that patent protection should be considered because of the availability of broad right to basic system or method concepts. Also, some applications software may be in the nature of an “add-in” or improvement to a commercially available software package such as a spreadsheet or database system. Sometimes these improvements cannot really be commercialized by licensing the program, but are only predictable and licensable by way of obtaining a patent on the improvement, which in turn can be licensed to producers of such software packages, in the same manner as will be discussed below with respect to technological software.

### **3.5.1.E. Commercializing Component Software**

Generally, component software will not have commercial value independent of the system, method or device it is used in. It typically is non-portable, that is to say that the component itself cannot be readily adapted for use in other species of similar systems, methods or devices without substantial revision or a complete rewrite of the code. Therefore, component software *code* normally will not be licensed independently of its corresponding system, method or device.

Correspondingly, in most cases the greatest value of component software is not in the code itself but rather in the underlying techniques, methods and algorithms embodied in the code.

It is also true that the novelty of a system, method or device is often found in the software component. For example, the novelty in an inventive new nuclear resonance imaging system is often found in the algorithms used to process the data for correlation and display. For this reason, university inventors will often think that their invention lies in the component software, when from a commercial and patent point of view the invention lies in the overall system or device incorporating the algorithms and techniques implemented with the software component. Moreover, in cases in which the underlying algorithm is a mathematical formula or method for solving a mathematical problem, the algorithm must be claimed as part of the larger system or device, or it will constitute non-statutory subject matter under 35 U.S.C. § 101. Unapplied mathematical formula and procedures are unpatentable.

Thus, software components are typically treated the same as any component part of a system or device. The patent or know-how rights in the system or device are licensed, as opposed to rights in the component part alone.

### **3.5.1.F. Commercializing Technological Software**

Technological software is probably the most misunderstood and most poorly commercialized technology in the university system. There are several reasons for this. First, in many if not most cases the value of technological software is not in computer code and the attendant copyright, but rather in the methods, algorithms or techniques that are embodied in the software. The historical model for commercial exploitation of computer software is the licensing of copyright and trade secret rights. Since trade secret licensing is fundamentally at odds with faculty's need to publish, copyright is the only viable historical licensing vehicle. However, methods, algorithms and techniques are not protectable by copyright.

The commercial value of actual program code to many potential licensees is minimal, either because the code is not up to commercial "product" standards and needs to be rewritten or because it will not work on the platform of interest. If as is usually the case the code needs to be rewritten, the potential licensee is often better off merely using published information about the software and avoiding a royalty to the university. There are, nonetheless, many "research" computer programs that have licensing value to other institutions and research operations and may be able to generate substantial royalties.

The primary model for commercialization of technological software inventions uses patent protection as the licensing vehicle. This licensing approach avoids the

limitations of the historical model, as the underlying methods, systems and techniques can be patented and licensed. In this scenario, the methods, techniques and systems underlying the software can be published while maintaining a licensable right in the software.

Using the patent licensing model of commercial exploitation of software has the added benefit of providing a well-defined bundle of rights for licensing to potential licensees. This is in contrast to attempts to license the copyright in a program, wherein the technical advance may be difficult to define with authoritative precision.

Furthermore, the patent model allows for protection of software inventions even if they are still not yet even experimental in nature or embodied in a computer program. Thus, licensing is not dependent on the development of a working program, allowing for more rapid and wider-ranging exploitation of software inventions.

Another problem is considerable resistance to the patenting of computer software inventions by university faculty. Many faculty fear that widespread patenting of software will interfere with their “programming freedom.” These fears are unfounded, but are real, and will continue to present an obstacle to commercialization of university developed software until they subside. It appears the primary cause for trepidation among software developers in academia is that patents are relatively new on the software development scene, and are viewed as a potential threat to the status quo. It is believed that experience with the patent system will gradually teach university software developers that on balance patent protection will further encourage, as opposed to discourage, the development of new software inventions.

### **3.5.1.G. Trademark Licensing**

Trademarks are another valuable asset that can be used to leverage the licensing of university developed computer software. For instance, a software product developed at a university may have a well-known name that can be trademarked and licensed along with the copyright in the program code.

### **3.5.1.H. Basic Licensing Issues**

If code is licensed, the primary licensing issues are:

- a. Limiting tort, contract or infringement liability; and
- b. Licensee access to source code and corresponding support and maintenance issues.

Typically, a software license will attempt to disclaim all liability or otherwise limit exposure.

Liability issues are beyond the scope of this paper, but it should be recognized that licensing a “product” such as computer software does entail liability not present in patent licensing. This liability needs to be actively managed in licensing agreements.

The other major issue in many licensing situations is the need for ongoing support maintenance of the licensed software. In the university setting, it is probably much more likely that source code will be made available to the licensee, thus mitigating the undesirable problem of committing the university to long-term support and maintenance obligations. Universities are not well suited to meet such obligations.

### **General Observations**

Of all technologies developed in the university environment, software has the greatest number of possible licensing vehicles. All three basic intellectual property rights, patents, copyrights, and trade secrets, can be used to license computer software. If patent protection is not sought for a software invention either because there is no viable invention, or because of cost considerations, the copyright in computer code (and perhaps trade secrets) is still available to license, assuming the invention has been embodied in a program.

## **3.5.2 THE BASICS OF PATENT PROTECTION FOR COMPUTER SOFTWARE**

### **3.5.2.A. Introduction**

Software patents have been providing powerful intellectual property protection for computer software since the United States Supreme Court cleared the way in 1981. Even though software has been protectable by patent for over five years, a great deal of confusion and misinformation still exists about it. For instance, many software developers still erroneously believe software is altogether unpatentable, or that algorithms, the basic building blocks of computer software, cannot be protected. There is no need for confusion on this topic any longer, however, since it is now indisputably clear that software is no less patentable than any other

technology. In fact, thousands of software patents have already been issued by the United States Patent and Trademark Office.

A patent is a particularly useful tool for the developers of mass-marketed software. For the independent developer that licenses or sells software to a publishing concern, a patent can provide a valuable additional asset to be licensed or sold along with program code. A patent can also protect an independent developer from having the concepts or ideas in software taken by a publisher without compensation. For the larger software developer or publisher, patents can prevent competitors from copying the functions or features of a product that differentiate it from the competition. In particular, a patent can discourage or prevent “cloning.” More importantly, perhaps, a software developer can use a patent as a bargaining chip to obtain rights to a competitor’s technology by cross-licensing and as leverage in all manners of other business situations. In addition, patents are useful in marketing the product and in developing positive public relations for the product and the company that employs the inventive software developers. Lastly, patents are generally required by venture capitalists before the venture capitalists will provide investment capital.

As is amply demonstrated below and in the attached articles, the only substantial downside to obtaining patent protection is its cost. The business advantages provided by patent protection, however, clearly outweigh the cost on average. In short, everything else being equal, a software developer who obtains patent protection and assembles a patent portfolio will have a strategic advantage over competitors who do not have a portfolio.

Set forth below is some basic information on patent protection for software. More information is set forth in the attached article entitled “Twelve Myths about Patent Protection for Software,” which addresses the common misconceptions and myths about the patent system in general and software patents in particular.

### **3.5.2.B. The Basis for Software Patents**

Many people in the software industry do not understand the basis for and nature of software patent protection. However, as far as patents are concerned, software can be considered much the same as other technologies. All such products, whether hardware or software based, clearly qualify to be protected under the patent system. In the eyes of the patent system, software is typically considered patentable as a method for operating a digital computer. Alternatively, software can be described and claimed in terms of a programmed computer apparatus that

can be considered a special purpose “machine.”

### **3.5.2.C. The Power of Patent Protection**

A patent can provide protection for product features that cannot be protected under either copyright or trade secret law. For instance, a patent can protect systems, methods, algorithms, functions and other aspects of software products that cannot be protected by copyright under any circumstance and that in many cases cannot be kept as trade secrets once a product is marketed.

Patents also have a broad reach. A U.S. patent gives the owner the right to prevent others from making, using, or selling the invention throughout the United States for a term of up to 20 years from the date the patent application was filed. Patents are enforceable against an infringing software product regardless of the manner in which the product is developed. Independent development is not a defense to patent infringement as it is in the case of action for copyright infringement and trade secret misappropriation. Patents obtained in countries outside the United States provide similar rights.

Patents also provide powerful relief from infringement in the form of injunctions and monetary damages. Triple damages and attorneys’ fees are available in appropriate cases of willful infringement, which can be established by showing that the defendant has knowledge of the patent and did not act reasonably in deciding to infringe. Because of these remedies, the majority of competitors will not willfully infringe a patent.

### **3.5.2.D. Unfounded Criticism**

Patent protection is sometimes criticized for being too expensive to acquire, taking too long to obtain, costing too much to enforce, and being too unreliable. Further, it is sometimes stated that obtaining a patent requires total forfeiture of all trade secret rights in a program. As explained in section 3.5.7, “Twelve Myths about Patent Protection for Software,” all of these criticisms are greatly overstated and substantially unfounded.

### **3.5.2.E. Types of Patentable Software Inventions**

Patent protection may be used to protect virtually any aspect of software. It is the best and usually the only effective protection against product imitation and “cloning.” Examples of software features that can be protected include the

following:

- Program algorithms
- Display presentations or arrangements
- Menu arrangements
- Editing functions
- Control functions
- User-interface features
- “Add-ins”
- Utilities
- Mathematical formulas used in a program to process data or control program execution
- Spell-checking routines
- Compiling techniques
- Program language translation methods
- Operating system techniques

#### **3.5.2.F. Novelty and Nonobviousness Requirements**

A software invention must be novel and nonobvious to qualify for protection. The novelty requirement means that, in order to be patentable, an invention must be new as compared to prior technology. The requirement of nonobviousness means that the differences between the invention and the prior technology are more than “obvious” trivial variations. Unfortunately, it is not possible to give a precise definition as to what is or is not obvious, because it is to a large extent a subjective judgment. However, if the invention provides significant new capabilities or performance not found in prior technology, there is usually a good chance that patent protection can be obtained.

#### **3.5.2.G. Source or Object Code Need Not Be Disclosed**

Where detailed illustration is not essential for a proper understanding of an inventive software feature, the U.S. Patent and Trademark Office (as well as the patent offices of other countries) encourages disclosure of the invention in the application in the form of pseudo code, block diagrams, and/or flow charts. Although one may include source code as part of this disclosure, it is not required. The applicant can usually meet the duty of disclosure with relatively high-level flow charts and diagrams. Pseudo code and data or object structure diagrams can also be used and are particularly helpful in object-oriented copyright law.

Likewise, the expression of high-level flow charts, data structure diagrams, and pseudo code can also be protected under copyright law by including a copyright notice in the patent specification. Thus, obtaining a patent does not require the forfeiture of copyright protection.

### **3.5.2.H. Trade Secret Status of a Patent Application**

*During processing, a patent application is held in secret by the U.S. Patent and Trademark Office. An application will be published (i.e., distributed to the public) by the U.S. Patent and Trademark Office 18 months after the filing date of the application, unless publication is prohibited beforehand through a petition by the applicant. The applicant thus maintains control over the secrecy of the disclosed material, and the exact nature of the invention, until the patent is issued. While awaiting issuance, the inventor may use the words “patent pending” to notify others that patent protection is being sought. The uncertainty of a competitor about the coverage of the patent when it issues may discourage it from copying the protected product at least until after the patent issues. Also, the prospect of an infringement suite can hamper a competitor’s ability to raise capital to develop a competing product.*

### **3.5.2.I. Important Deadlines**

There are important deadlines in seeking patent protection. As a general rule, for protection in the United States, a patent application must be on file in the U.S. Patent and Trademark Office within one year of the date on which the invention is first sold, offered for sale, used publicly, or disclosed in a printed publication. If an application is not on file within the prescribed period, the inventor’s rights to a patent for the invention are forfeited forever. To obtain patent protection in most foreign countries, a patent application should normally be filed before any public disclosure or public use anywhere.

*The deadline rules stated above are greatly simplified and are intended only to give the reader a sense of the deadline requirement of obtaining patent protection. These rules are not legally precise and are not legal advice. They should not be relied on to determine the right to file or when to file. There are many subtleties and nuances to the deadlines within patent law that must be appreciated before a sound legal conclusion can be reached regarding any particular situation. For instance, under certain circumstances, a public use could be found where the invention is practiced secretly in a company’s proprietary computer operation.*

### **3.5.2.J. Design Patents**

In addition to utility patents, there are also “design” patents. In contrast to a utility patent that protects the functional aspects of technology, a design patent protects “ornamental” designs, wherein the invention lies in aesthetic appearance. Design patents are available to protect the graphical aspects of computer screen displays. For instance, icons, soft key menu displays, type fonts and ornamental border designs can be protected with a design patent.

Much of what can be covered by a design patent can also be protected under copyright law. There are, however, differences in the legal basis for both of these rights that can give design patents advantages over copyright protection. Although there is some legal uncertainty in the area, it does appear that both copyright and design patent protection can be obtained for the same screen display. Accordingly, it is presently considered advisable to obtain both forms of protection where possible.

Design patents are considerably simpler in make-up to utility patents and thus are considerably less expensive to obtain. In general, a design patent for a screen display costs approximately \$2,000 to obtain.

### **3.5.2.K. Summary**

Patent protection is a powerful form of software protection that can and should be used aggressively. It can be creatively applied to protect many features of an invention that cannot be protected by copyright or trade secret protection. In most cases, the only disadvantage to obtaining patent protection is the cost. The cost, however, is minimal compared to typical software development expenditures and the value of achieving the exclusive right to market the technology being protected by the patent.

## **3.5.3 THE BASICS OF TRADE SECRET PROTECTION FOR COMPUTER SOFTWARE**

Many software products or features can be protected as a trade secret. To qualify as a trade secret, the software product or software feature sought to be protected must be of value and not readily known or ascertainable to others. A trade secret must also provide a demonstrable competitive advantage and be subject to reasonable efforts to protect and maintain it.

### **3.5.3.A. Reasonable Efforts to Maintain Secrecy**

A software company should undertake certain internal and external procedures to enhance the company's claim that its trade secrets are subject to reasonable efforts under the circumstances to maintain their secrecy.

Implementing internal and external safeguards to maintain the secrecy of trade secrets and other confidential and proprietary information is of critical importance. Instituting a comprehensive employee program, placing special protective notices and legends on materials that indicate that such materials contain trade secrets or other confidential and proprietary information, maintaining physical security measures, and placing appropriate nondisclosure provisions in agreements with third parties are some of these internal and external safeguards.

### **3.5.3.B. Internal Procedures to Maintain Secrecy**

There are a multitude of internal procedures that a company can adopt to enhance its claims that it is reasonably protecting its trade secrets and to demonstrate its intention to keep its trade secrets and other confidential and proprietary information confidential. Establishing an employee program to protect trade secrets is the most important internal procedure.

Such a program includes the following:

- Requiring all salaried employees to sign written nondisclosure agreements as a condition of their hiring, and
- Similarly requiring all hourly employees to sign statements reciting the company's policies with respect to its trade secrets and other confidential and proprietary information.

The employee program should also include placing notices of the company's policies with respect to its trade secret and other confidential and proprietary information prominently on bulletin boards and other areas in which employees would be likely to see them (e.g., the company lunchroom), and disseminating memoranda to employees on an annual basis to remind them of company policies. Finally, a company's employee program should include the performance of exit interviews when employees leave the company's employment. During these exit interviews, the employee should be reminded of the company's policies with

regard to its trade secrets and other confidential and proprietary information. In addition, the employee should be reminded in the interview (and in a follow-up letter to the employee and to the employee's new employer) of the general areas of expertise and information developed by or disclosed to the employee during employment, reminding him or her that this expertise and information cannot be disclosed to others, including the new employer.

In addition to the employee program, there should be a reasonable level of physical security at company offices to physically safeguard its trade secrets and other confidential and proprietary information. Some common physical security measures that should seriously be considered include:

- Keeping all trade secret and other confidential and proprietary information under lock and key (or, if "on-line," subject to computer security measures to prevent unauthorized access) and restricting access to such information to only those employees who need access to perform their respective jobs;
- Restricting access to certain parts of company offices to authorized personnel only;
- Keeping all important doors locked;
- Maintaining a register to be signed by all visitors;
- Requiring that visitors be escorted;
- Requiring approval of a corporate officer for any office tours and restricting the scope of such tours; and
- Requiring all employees and visitors to wear distinguishing badges, especially to gain access to restricted areas.

### **3.5.3.C. External Procedures to Maintain Secrecy**

In addition to internal procedures to maintain the secrecy of all trade secret and other confidential and proprietary information, there should also be certain external procedures that will reasonably protect information outside the company. The single most important external procedure that must be adopted is the insertion of nondisclosure provisions in all agreements under which trade secret or other confidential and proprietary information maybe disclosed. Without nondisclosure

provisions in all agreements with “outsiders” (any nonemployees), any information that is disclosed to outsiders could become public domain information free from any trade secret obligation, thus causing complete loss of trade secret protection with regard to such information. Also, any trade secret or other confidential and proprietary information disclosed to an outsider should contain a protective legend (see “Protective Legends” section below).

#### **3.5.3.D. Legal Duties Providing Foundation for Scope of Protection**

Trade secret law prevents those who are under a legal duty from disclosing trade secret and other confidential and proprietary information to any third party who does not have right to know the information. Therefore, trade secret protection hinges upon establishing a legal duty under which the trade secrets will be protected.

Typically, there are two foundations for establishing this legal duty. First, there is an implied duty at law (arising without any agreement) for employees not to disclose trade secret and other confidential and proprietary information of their employers to third parties. It is important to realize that employees are under a duty even without a written agreement. However, a written employee nondisclosure agreement is by far the best way of establishing this duty. A written document can more clearly establish the employee’s knowledge of the existence of trade secrets and other confidential and proprietary information. Such a document can also more directly establish a company’s reasonable efforts to protect not only its own information, but also the trade secrets and other confidential and proprietary information that belongs to third parties and is disclosed in confidence to the company.

Second, there is a duty established by an appropriate agreement not to disclose trade secret and other confidential and proprietary information to third parties. Generally speaking, a contractual legal duty is the only way to establish the legal duty with respect to parties that are not employees. Therefore, all written agreements between a company and nonemployees that contemplate the disclosure of trade secret or other confidential and proprietary information should include a nondisclosure provision that establishes the legal duty of the recipient of the information not to disclose it to third parties.

#### **3.5.3.E. Major Advantages and Disadvantages to Trade Secret Protection**

There are three major advantages to trade secret protection of software technology.

First, trade secret protection protects valuable software ideas and not merely the expression of those ideas as copyright protection does. Second, it can effectively protect special features in marketed software products as long as the features are not readily ascertainable by observation or through reverse engineering. And third, if the trade secret protection can be maintained, there is no limit to the length of time under which protection is available, unlike patent and copyright protection.

However, there are a few major disadvantages to trade secret protection. First, trade secret protection will only protect trade secrets against disclosure by parties who have a legal duty not to disclose. So, in most situations, only employees and parties under a contractual duty not to disclose are affected by this protection. Second, trade secrets cannot protect mass-marketed software product features that are readily ascertainable. Third, unlike patent protection, trade secret protection does not prohibit independent development. If a company independently develops a trade secret even though another company had developed it first, the second company would not be infringing the initially developed trade secret. It is a trade secret of each company as long as both keep it a secret. Finally, the internal and external procedures required to maintain secrecy can be quite expensive.

#### **3.5.3.F. Enforcement**

Remedies for trade secret infringement include, under appropriate circumstances, both injunctive relief and damages, although damages may be awarded independent of the injunctive relief. Damages may be based on the infringed party's losses or the infringer's profit or some combination; in addition, under appropriate circumstances, punitive damages may be awarded.

#### **3.5.3.G. Summary**

Trade secret protection is available and should be used to protect any software product or special feature in the product (including documentation) that gives a competitive advantage in the marketplace and is not readily ascertainable to others using the product. While it is an excellent protection under such circumstances, it cannot protect many important ideas, information or features that are published or readily ascertainable to others using the product. It is best used in combination with patent and copyright protection, which can often protect the other important ideas and features.

### **3.5.4 THE BASICS OF COPYRIGHT PROTECTION FOR SOFTWARE**

### **3.5.4.A. The Basic Features and Limitations of Copyright Protection**

Copyright protection is another powerful form of protection for a software product. The owner of copyrighted software has the exclusive right to control:

- Duplication or reproduction of the software other than for archival purposes;
- Preparation of derivative versions of the software for distribution to others; and
- Distribution of copies of the software to the public, by sale, rental or otherwise.

The Copyright Act strictly prohibits anyone from attempting to perform the above exclusive rights without permission from the copyright owner.

### **3.5.4.B. Duplication/Reproduction Right**

The exclusive right to control duplication protects the owner of a software product against unauthorized copying, such as where the source or object code of the product is copied verbatim. It can also protect against certain indirect forms of unauthorized copying such as unauthorized translation of a product's code to a different programming language, and it may even protect against use of the code's structure, sequence and organization as a detailed outline to prepare a substantially similar program.

### **3.5.4.C. Derivative Works**

The right to control the preparation of derivative versions of a software product is an important and valuable aspect of copyright protection. This right makes it unlawful for someone developing a competing product to modify, adapt or change software product without authorization from the owner of the copyright in the product. There is an important exception to this right. Users of a software product who own a copy of the software may make or authorize the making of an adaptation of that software product, provided that the adaptation is created as an essential step of the owner/user in the utilization of the software product in conjunction with a machine and that it is used in no other manner. This right does not necessarily apply in the case where the software is licensed if the licensee bargains this right away in the license.

#### **3.5.4.D. Distribution Right**

The right to control distribution is another independent right granted to the copyright owner by copyright law. This right can be transferred by the owner to a distributor while, for example, the right to reproduce copies for distribution by the distributor remains with the owner.

#### **3.5.4.E. Term of Copyright**

A work that is created on or after January 1, 1978, is automatically protected from the moment of its creation and is ordinarily given a term enduring for the author's life plus an additional 70 years after the author's death. In the case of "a joint work prepared by two or more authors who did not work for hire," the term lasts for 70 years after the last surviving author's death. For works made for hire, and for anonymous and pseudonymous works (unless the author's identity is revealed in Copyright Office records), the duration of copyright will be 95 years from publication or 120 years from creation, whichever is shorter.

Thus, a copyright lasts considerably longer than a patent (up to a 20-year term for a U.S. utility patent) but potentially not as long as a trade secret, which can theoretically last forever.

#### **3.5.4.F. No Protection for Ideas**

It is important to understand that copyright law cannot protect the ideas embodied in a software product, just as a copyright cannot protect the ideas communicated by a story or a description in a book. The ideas embodied in a software product include any methods, systems, algorithms, and applied mathematical formulas used or implemented by the product. Copyright law cannot protect a product from being imitated or cloned so long as only unprotected ideas are imitated or copied. Thus, copyright law has a broad reach in the sense that it is available to protect virtually any software product, but the protection afforded is very limited in subject matter scope in comparison to patent and trade secret protection.

#### **3.5.4.G. Code Protected**

Copyright protection is available to protect any type of code, including source code, object code, machine code, firmware, and on any medium (such as magnetic disc, ROM or in print form). Moreover, it does not matter whether the software is

an application program or an operating system.

#### **3.5.4.H. Protection of “Look and Feel”**

Copyright protection also appears to be available to protect certain aspects of a software product’s user-interface, or what has come to be known as its “look and feel.” Although some courts have found to the contrary, the protection available to user-interface features through copyright law will likely be limited in the long run to artistic expression that can exist independently of the utilitarian features of the screen display.

#### **3.5.4.I. Obtaining Copyright Protection**

A copyright is automatically obtained when software or other copyrightable work is created. If the software is published, an appropriate copyright notice should be properly placed on all published copies but is not required of works published on or after March 1, 1989. Nonetheless, there are significant advantages under U.S. law to including a copyright notice on all published works. For example, the defense of innocent infringement is not available to infringers if the infringed work properly includes a copyright notice.

Further, as is explained in the Protective Legends text of this booklet, using an appropriate copyright notice will still be important to maintaining international protection of published works under the Universal Copyright Convention and the Buenos Aires Convention, which confer protection in certain foreign countries. In addition, use of a copyright notice is an extremely practical and useful device from a deterrence standpoint, since an observer of the work will know immediately that the work has not been donated to the public domain. Accordingly, use of an appropriate copyright notice on a published work is important.

#### **3.5.4.J. The Benefits of Registration**

Formal copyright registration with the U.S. Copyright Office is not necessary to obtain or maintain copyright protection. However, for U.S. Copyright owners, a registration must be obtained before the copyright can be enforced. Nonetheless, under the Berne Convention as implemented in the U.S., registration is not required for non-U.S. copyright owners to bring suit.

If software is registered with the U.S. Copyright Office before an act of infringement, the copyright owner is entitled to two substantial benefits not

otherwise available in enforcing its copyright. The first benefit is the ability to force the infringing part to pay the copyright owner's attorneys' fees for enforcing the copyright. The second benefit is the ability to obtain "statutory damages" from the infringer. "Statutory damages" allows a court to make the infringer pay substantial damages without any proof of actual damages, which are often difficult to prove. These two advantages are so valuable that a copyright registration should be acquired promptly as a matter of course on each released version of a software product. The registration process itself is discussed below.

#### **3.5.4.K. Publication**

Software that is distributed to the public without a signed license agreement is normally considered "published." However, software that is under development or that is licensed to a limited number of users under a trade secret license is usually considered "unpublished."

#### **3.5.4.L. Copyright Notice**

Any software product, whether published or not, should bear an appropriate protective legend. In the case of published software, this legend should include a copyright notice. In the case of unpublished software, a copyright notice should also be included, but it should be included in a manner that ensures that others understand the unpublished status of the software.

Generally, software products published prior to March 1, 1989, must have a copyright notice. However, under some circumstances deficient or missing notices for such products can be corrected retroactively. This is discussed further in the "Protective Legends" section, which also addresses the proper form and placement of protective legends for published and unpublished software products.

#### **3.5.4.M. Maintaining Archives**

It is important to maintain two copies (a second copy can be easily reproduced from one copy) of each version of a software product and related software documentation. The copies should be kept in archives so that they can be later used to aid an infringement investigation and/or to obtain registrations, if necessary. The original and each released version should be retained in these archives as long as any version of the software is supported (e.g., marketed software) or used (e.g., internally used software).

### **3.5.4.N. Copyright Registration**

Generally speaking, a copyright in a product can be registered in the United States at any point during the life of the copyright. However, to qualify for the benefits of attorneys' fees and "statutory damages" as discussed earlier, registration must occur prior to the act of infringement against which the copyright is sought to be enforced.

To obtain a registration, a developer or owner of the copyright must file with the U.S. Copyright Office a properly completed application (usually a Form TX), a \$30 fee and a copy of at least a limited portion of the code sought to be protected (the "deposit"). If the software contains trade secrets, the Copyright Office has provisions that allow the registrant to meet the deposit requirement by supplying the first 25 and last 25 pages of source code with portions containing trade secrets blocked out; or the first 10 and last 10 pages of source code alone, with no blocked out portions; or the first 25 and the last 25 pages of object code plus any 10 or more consecutive pages of source code, with no blocked out portions; or for programs of 50 pages or less in length, the entire source code with trade secret portions blocked out. Please keep in mind that that no copyright is conferred to blocked out code.

To have the best evidence for use in a potential infringement suit, it may be wise to deposit virtually all pages of the source listing, or at least to deposit representative portions of the source code from each module in the program (with trade secret portions blocked out). In any event, the trade secret status of the software can be maintained in registered software. Typically, the Copyright Office will process a normal software registration within two to three months after receiving the publication (three to four months for "special relief" registrations). The Copyright Office will expedite handling of a copyright registration of the copyright is the subject of a lawsuit. A special request to expedite the handling of the registration must be made with an additional fee of \$330 in addition to the normal \$30 filing fee.

Since using the wrong dates or mischaracterizing the software or its status on the application can render a copyright registration ineffective, it is strongly suggested that a knowledgeable attorney be consulted to review both the copyright application and the deposit before it is submitted to the Copyright Office.

### **3.5.4.O. Shrinkwrapped licenses and clickwrapped licenses**

Courts have generally held shrinkwrapped licenses to be unenforceable since purchasers have no ability to review the license before the purchase, therefore, a shrinkwrapped license undercuts copyright rights and are thus preempted by federal copyright law. However, in software distributed on the web, purchasers are able to review the license beforehand and "click" their agreement, and therefore a "clickwrapped" license is more likely to be held as enforceable.

#### **3.5.4.P. Enforcement**

Remedies for infringement of the exclusive rights of a copyright owner include injunctions, impounding and disposing of infringing articles, and monetary damages. Statutory damages, costs and attorneys' fees are available if, *and only if*, a registration is obtained before the act of infringement requiring suit. Importation of infringing copies can also be prevented under laws pertaining to United States Customs.

#### **3.5.4.Q. Summary**

Copyright protection is available to and should be used to protect virtually all software products. While it is excellent protection against unauthorized duplication, it cannot protect many important features of a software product. It is best used in combination with patent and/or trade secret protection, which can often protect the other important features.

### **3.5.5 PROTECTIVE LEGENDS**

As discussed in the "Copyright Protection" section of this chapter, any software product or similar copyrightable work, whether published or not, should bear an appropriate protective legend. The purpose of the protective legend is to deter copyright infringement. The U.S. copyright statute no longer requires use of a copyright legend on copyrighted material.

An effective notice may contain the following three elements:

1. The letter "c" in a circle, such as "©", the word "Copyright," or the abbreviation "Copr.";
2. The year of first publication of the work; and
3. The full name of the owner of the copyright, or an abbreviation by which the name can be recognized, or a generally known alternative designation of the

owner.

Although an effective notice for the United States may use either the “©”, the word “Copyright” or the abbreviation “Copr.”, *there are certain foreign countries in which the “©” must be used in order to have an effective copyright notice.* While it may not be possible to use the “©” within the code itself because there is no ASCII character for it, it should at least be used on labels affixed to the software or if possible used on the screen displays of the software where it can be graphically generated. Within the code itself where the “©” is not available, it is recommended to use the symbol “(C).” At least one court has found the “(C)” symbol effective as notice of copyright. However, it is very possible that this alternative copyright symbol will not be accepted in foreign countries. Thus, it is of paramount importance that the “©” be used on labels on the media containing the software, or on associated documentation.

Also, in many Latin American countries an enforceable copyright notice must include the phrase “All Rights Reserved.” Therefore, by combining all the recommended elements, the recommended notice for a published software product that was published in 2001 and owned by XYZ, Inc., would look like this:

© Copyright 2001 XYZ, Inc.  
All Rights Reserved.

If a published software product is substantially enhanced, the date in the copyright notice may be left the same; having too old of a date in the copyright notice will not take away the effectiveness of the notice, even though it may shorten the normal duration of protection. However, it is typically preferable to replace the original publication date for substantially revised published works. The original publication date can either be replaced with the year in which the substantially revised version is published, or the original date can be left in place and the date of each substantial published revision added, with each year being separated by a comma.

The latter approach may be preferable, since the copyright notice then essentially reveals the history of the original publication and each substantial revision. Using this approach, it is more difficult to make a fatal error in the copyright notice, since the original date is still present and since the new date would apply only to changed material in the copyrighted work (if too recent a date is used, it can have the effect of omitting the copyright notice altogether). If only trivial revisions are made, or if one is indefinite about whether substantial revisions have been made, use only the prior date or dates.

In contrast to published software, many types of software are unpublished. Most software

products are *unpublished* during their development phase. Except for mass-marketed microcomputer software products, most licensed software products remain unpublished, assuming that they are distributed through license agreements that are *agreed to and signed by* all parties along the distribution chain, including the end user. Finally, even if the object code of a software product is distributed with a copyright notice applicable to published works, the source code of the product and other confidential and proprietary software documentation is virtually always maintained as a trade secret and is, therefore, unpublished. These unpublished works benefit from a form of copyright notice that will not imply publication and potential loss of trade secret status. The applicable notice should be placed on all media, should be embedded in the source and object code of the product (preferably in each module) so that it will appear on any object or source code listing, and should appear on the product screen display at initialization or log-on.

Since unpublished works are also normally maintained as trade secrets, it is appropriate to combine an unpublished copyright notice with a legend applicable to protecting trade secrets and other confidential and proprietary information. Set forth below are two examples of a combined unpublished copyright notice and trade secret legend. Where space on a product permits, it is suggested that the following larger combined copyright/trade secret protective legend be placed on all trade secret or other confidential and proprietary media and be embedded in all trade secret or other confidential and proprietary code, preferably as a header in each code module. In the legend, “**CLAIMANT**” should be replaced with the name of the copyright owner, which is normally the name of a company (in the copyright notice at the end of the legend, the formal name of the company, e.g. “XYZ, INC.” is preferred; in the earlier locations of “**CLAIMANT**,” the common name of the company, e.g., “XYZ,” may be used).

**CLAIMANT CONFIDENTIAL AND  
PROPRIETARY**

THIS WORK CONTAINS VALUABLE CONFIDENTIAL AND  
PROPRIETARY INFORMATION. DISCLOSURE, USE OR  
REPRODUCTION WITHOUT THE WRITTEN AUTHORIZATION OF  
**CLAIMANT** IS PROHIBITED. THIS UNPUBLISHED WORK BY  
**CLAIMANT** IS PROTECTED BY THE LAWS OF THE UNITED STATES  
AND OTHER COUNTRIES. IF PUBLICATION OF THE WORK SHOULD  
OCCUR THE FOLLOWING NOTICE SHALL APPLY:

“COPYRIGHT © 20XX **CLAIMANT**  
ALL RIGHTS RESERVED.”

Although the fully copyright/trade secret protective legend indicated above is preferred, the following abbreviated protective legend may be used where there is insufficient room for the full legend.

**THIS IS AN UNPUBLISHED WORK CONTAINING CLAIMANT  
CONFIDENTIAL AND PROPRIETARY INFORMATION. IF  
PUBLICATION OCCURS, THE FOLLOWING NOTICE APPLIES:**

**“COPYRIGHT © 20XX CLAIMANT  
ALL RIGHTS RESERVED.”**

As was indicated previously, **“CLAIMANT”** in this legend should be replaced with the name of the software owner, which is usually the company name of the software developer or other owner, and may be the common name of the company, e.g., **“XYZ,”** rather than the formal name, e.g., **“XYZ, Inc.”**

In addition to the complete copyright/trade secret protective legend being embedded in chips containing trade secret, licensed, or other confidential and proprietary software, it is suggested that, at a location on or in the machine containing the chips, the following notice be included:

**CLAIMANT CONFIDENTIAL AND  
PROPRIETARY**

**THIS MACHINE CONTAINS VALUABLE CONFIDENTIAL AND  
PROPRIETARY INFORMATION. DISCLOSURE, USE OR  
REPRODUCTION OUTSIDE OF CLAIMANT IS PROHOBITED EXCEPT  
AS AUTHORIZED IN WRITING. THIS MACHINE CONTAINS AN  
UNPUBLISHED WORK WHICH IS PROTECTED BY THE LAWS OF THE  
UNITED STATES AND OTHER COUNTRIES. IF PUBLICATION OCURS,  
THE FOLLOWING NOTICE SHALL APPLY:**

**“COPYRIGHT © 20XX CLAIMANT  
ALL RIGHTS RESERVED.”**

In those cases where space is limited, and preferably in all instances, an abbreviated external legend should also be placed on each chip containing trade secret, licensed, or other confidential and proprietary software as follows:

**UNPUBLISHED © 20XX**

## CLAIMANT CONFIDENTIAL

In these notices, 20XX is preferably the year of creation. The year date 20XX is preferably updated, using the previously described guidelines for unpublished works, whenever substantial revisions are made. Again, the word “CLAIMANT” should be replaced with the name of the copyright owner. For protection in Latin American countries, the phrase “ALL RIGHTS RESERVED” should be added to the above abbreviated notice.

Finally, if distribution of a software product to the United States government is contemplated, be forewarned that in many instances special United States government notices and legends must be placed on the product to maintain copyright and trade secret protection. The government regulations discussing these notices and legends are too long and complex to address in this booklet. However, before agreeing to distribute a product to the U.S. government, you should consult an intellectual property attorney knowledgeable about these regulations. *If this is not done, intellectual property rights in a software product can easily be permanently lost.*

The previously described protective legends are generally very useful. However, because of the fast changing law with respect to software, and because of the widely varying circumstances that may be used in marketing trade secret, licensed, or other confidential and proprietary software, *it is generally advisable to review each software product protection scheme on an annual basis with an intellectual property attorney knowledgeable about software protection.*

### **Summary**

Any software product, whether published or not, should bear an appropriate protective legend. Software that is distributed to the public without a signed license agreement is normally considered “published.” On the other hand, software containing trade secrets that is under development or that is licensed to a limited number of end users under a trade secret license is usually considered “unpublished.” In the case of published software, the legend should include a copyright notice. In the case of unpublished software, a copyright notice can be included, but it should be included in a manner that ensures maintaining the unpublished nature of the software. The protective legend for unpublished software should also include language indicating the trade secret nature of the software.

### **3.5.6. GUIDELINES FOR DETERMINING WHEN TO PATENT SOFTWARE AND OTHER PROCESS INVENTIONS**

These guidelines have been developed to assist decision makers in deciding, on an

invention-by-invention basis, whether to protect software (or process) inventions through a patent, as a trade secret, or through a combination of both. While this section is limited to discussing patent and trade secret protection issues, the reader should be aware that copyright protection is available to protect software with or without patent and/or trade secret protection.

In using these guidelines, affirmative answers to the questions following below weight in favor of utilizing patent protection. However, each situation calls for its own analysis. Therefore, while one affirmative answer in one situation may outweigh all others so that patent protection is pursued, in a second situation, several yes answers may not overcome other considerations.

***Will others in the industry be likely to make the same invention; i.e., are others searching for a solution to the problem the invention solves?***

If others in the industry are likely to make the same invention, the risk of being precluded from use of the invention by others who obtain patents on parallel research and development is significantly increased. If two independent parties develop technology, the party who patents the technology will have the right to preclude the other from using it. This may even be the case where, for example, a company is the first to make the invention but it protects the invention as a trade secret (versus as a patent), and a competitor subsequently makes the same invention and patents it. The patent law has indicated its preference for a second inventor patentee over a first inventor who keeps the invention secret, stating:

As between a prior inventor who benefits from a process by selling its product but suppresses, conceals, or otherwise keeps the process from the public, and a later inventor who promptly files a patent application from which the public will gain a disclosure of the process, the law favors the latter.

W.L. Gore & Assoc., Inc. v. Garlock, Inc., 721 F.2d 1540, 1550 (Fed. Cir. 1983).

***Is broad patent protection available?***

If the invention constitutes a substantial advance (as opposed to being a minor improvement), a patent on the invention normally will have substantial commercial value. Nonetheless, the law does not require any particular extent of advance or novelty. Any novelty is sufficient novelty.

***Can patent protection be obtained for portions of the software or process while keeping the remainder as a trade secret?***

Often, both patent and trade secret protection can be used to protect the same product. What needs to be disclosed in order to patent an inventive aspect or portion of a process or software package may and often will constitute a relatively minor portion of the total technology used in the package or process. While trade secret rights will be forfeited in the material that is disclosed to support the claim to the invention covered by the patent, that which is not disclosed may be kept as a trade secret. Typically, there is no need to disclose source or object code so this can usually be kept secret. Further, some parts or modules of a software package may not be disclosed in the patent application at all. Accordingly, some aspects of a package may be protected with trade secret law and other parts may be covered by one or more patents (computer code is protectable by copyright in either case).

***Is there a strong market for the invention?***

If there is a strong market for the invention, others, including competitors, will likely try to develop the invention and share the market with the original inventors. This can best be precluded by obtaining a patent on the technology forming the invention, thus precluding use by competitors even if they independently develop it.

***Will the invention have a long market life?***

One must be sure to distinguish between the life of an invention and the life of the product it is first used in. Many inventions are used in a series or family of products over a long period of time. If the life of the invention is long, it is even more important to patent rather than to attempt maintaining the invention as a trade secret since trade secrets can be discovered by reverse engineering or can be independently developed by others. Otherwise, what could have been retained exclusively in, for example, a series of products through patent protection, will instead be eventually copied by competitors, taking away the competitive edge.

***Can the invention be licensed to others, thus generating direct royalty income, or can it be used in a cross-licensing arrangement to avoid royalty liability and obtain design freedom?***

Many companies license their patents to generate royalty income as a supplement to operating income. For some companies, such as Texas Instruments, royalty income can approach or exceed operating income. Also, cross-licensing usually reduces the royalty charged for use of another party's patent(s). Further, a blanket cross-license with a competitor provides design freedom, i.e., there is no need to design around or to be concerned with infringing the other party's patents.

## ***Will a patent protection on the invention be useful as a defensive measure?***

In addition to being used affirmatively, a patent can be highly effective as a deterrent to competitors who might, but for the fear of a counter suit, press infringement charges. Similarly, if a competitor does assert its patents, having a strong patent portfolio to retaliate with can be extremely effective.

## ***Is it impossible or difficult to protect the invention as a trade secret?***

For any number of reasons, some inventions cannot be protected as trade secrets. For instance, the user-interface of a software product is generally difficult if not impossible to protect as a trade secret. Moreover, some technology that is potentially protectable by trade secret law is available to a very large portion of a company's employee population and, therefore, may not be under sufficient control to be protectable as a trade secret, either legally or practically.

Also, some trade secret processes can either be reverse engineered from the end product, or can at least be generically detected from the end product. Such technology is best protected by a patent, since otherwise competitors will likely learn or develop the technology and use it freely.

In addition to a company's trade secrets being discovered by others through reverse engineering, there may be many other sources through which a competitor learns of a company's trade secrets. An analysis of whether competitors have this ability can include the following factors:

- Who are the competitors?
- How much do competitors know about what the company is doing?
- Is there significant turnover of the company's research and development personnel?

A company's employees who are in contact with competitors may be in a position to observe and question employees of those competitors to determine how safe a company's trade secret technology might be. This same approach can be used to detect infringement of the company's patents.

## ***Is it likely that a patent covering the invention will disclose no more than employees might ordinarily disclose anyway?***

Each employee within a company's organization has knowledge of some portion of the company's overall technology. Unless a company is extremely vigilant about training each of its employees with respect to the existence of the particular trade secret technology that the employee has access to, there is a likelihood that employees will not realize that much of this technology (e.g., algorithms developed by programmers) constitutes trade secret material, rather than merely the tools of the employee's trade. This is particularly true of computer professionals, who are highly mobile and whose culture fosters sharing of information. Accordingly, the likelihood of many trade secrets being disclosed by departing employees, by employees' sharing of information with outside peers, and by employees' sharing of information at seminars must be considered in deciding whether or not to patent an invention. Once the invention is patented, the employee will be safe in sharing the invention with others, thus providing a springboard to learn even more technology.

***Is the work being done under U.S. government contract, thus requiring public disclosure?***

When a development is made under a U.S. government contract, public disclosure of the invention often results. In most instances, however, the contractor is permitted to retain all commercial rights by patenting the invention (except for a royalty-free license to the government). If the invention is not patented, it will be available to anyone, including the company's creditors.

***Is the work being performed in cooperation with a university?***

In the university environment, professors are encouraged to publish their work. Patent protection is thus well suited to this environment because trade secret protection is usually not a viable option.

***Is the work being done under joint development with another company?***

In work being done under a joint development program, patents help define technology initially contributed from each party, as well as helping to define rights in the jointly developed technology. Therefore, a healthy company patent portfolio can be very beneficial when joint development efforts are undertaken.

**Conclusion**

Many factors should be considered to determine whether or not patent protection is appropriate for a software or other process invention. Generally, as illustrated above, patent protection is appropriate when others in the industry are likely to make the same invention

or if broad protection is potentially available. In addition to using patents affirmatively, such as to generate royalty income or to stop others from using the patented technology, it is also important to build a patent portfolio for cross-licensing and defensive purposes. Accordingly, patent protection may be pursued in certain cases even where trade secret protection is otherwise a viable option.

### **3.5.7 TWELVE MYTHS ABOUT PATENT PROTECTION FOR SOFTWARE**

Based on a wide range of contacts with not only computer lawyers, patent lawyers, and general lawyers, but also with industry, we have found a continuing, pervasive misunderstanding of the patentability of software. Patent protection is routinely available to protect functions and features of computer programs, including look and feel features, that are left unprotected by copyright. This section on Myths is not intended to be exhaustive. Rather, it is intended to help dispel the common misunderstandings about the patent system in general, and software patents in particular.

#### **Myth Number 1**

*Software is not patentable.*

Software is patentable and has been patentable for a long time. Actually, a large number of software patents have already been issued by the U.S. Patent and Trademark Office.

#### **Myth Number 2**

*Software is only patentable if it is used to control some physical thing or process connected to the computer, like a robot or chemical processing equipment.*

There is not such requirement to obtaining a software patent. As viewed by the patent law, software can be considered a method for operating a digital computer. The thing controlled need be nothing more than the hardware every computer includes, e.g., its memory, registers or CRT display. For instance, a software method can be patented in terms of controlling the flow of information in a system, the interaction of the user interface with the user or with the underlying operating system, or as a method for carrying out a function such as adding or deleting rows in a spreadsheet. A computer programmed with software can also be considered a special purpose electronic device having the functions it is programmed to carry out. Electronics devices have always been patentable.

### **Myth Number 3**

*Algorithms are not patentable.*

Far from being unpatentable, algorithms can constitute the very method that is patented. It is only mathematical procedures and formulas that are not patentable.

### **Myth Number 4**

*Software inventions that rely on mathematical formulas or procedures cannot be patented.*

A mathematical formula or procedure can be patented as part of a software system or method. In fact, the mathematical formula or procedure may be the very feature that makes the software novel.

### **Myth Number 5**

*Patents take too long to acquire to be of value in protecting fast-moving software technologies.*

While the average software product may only have an 18-month lifespan, most meaningful technological *advances* (as opposed to a product), whether large or small, have a 10-year and often greater lifespan. For instance, the basic technology in the Apple Macintosh interface was developed at least 20 years ago, and the precursor of the Lotus 1-2-3 electronic spreadsheet, VisiCalc, was available in the late 1970s. It is only unsuccessful products and technologies that are short-lived, and there is no need to protect such technologies since they have limited value.

### **Myth Number 6**

*Patents take too long to obtain to be of value in protecting a product.*

For many technologies patents can often be obtained within one year of filing. Software patents in the U.S. tend to take approximately three years to obtain. While this delay may appear to be significant to the uninitiated, in reality it is only rarely a problem, and is usually advantageous. It normally takes at least six months to a year before competitors decide whether software is worth copying or cloning. After deciding to clone the software, it usually takes at least one year before a competitor is able to market

the competing product. By then, however, the patent will normally be issued and enforceable. Moreover, since competitors do not know precisely what protection is obtained until the patent issues, they cannot design around the patent with any certainty until it is issued. Thus, a competitor takes a risk in copying a product or product feature before the patent on it issues.

### **Myth Number 7**

*Patents are too easy to obtain, and therefore meaningless.*

The standards for patentability are much higher than commonly believed. Uncreative rehashing of old technology will not result in a patentable invention. However, most software developers do lots of creative designing, inventing, and innovating. Moreover, software technologies lend themselves to more creativity than other technologies. It is precisely this type of technical contribution that can be patented. One of the best measures of patentability is the degree to which the technology is copied. The more it is copied and imitated, the more likely it is patentable. Thus, it is the products that most need protection that are easiest to protect through patents.

### **Myth Number 8**

*Patents are too expensive and difficult to enforce.*

Patents on true inventions are rarely found invalid or unenforceable and even marginal patents stand a very good chance of being held valid and infringed. In addition, there are very stiff penalties for infringing a patent particularly where it is done with premeditation. As a result of these factors, patents are to a surprising extent self-enforcing.

Patent attorneys know from experience that few companies will risk millions of dollars on the hope that they can convince a judge or jury that the patent they are infringing on is invalid. Moreover, the patent system has been reinvigorated as the result of a new court of appeals that was created ago to handle all patent appeals. This court has made it much easier to enforce patents, as many disgruntled defendants have discovered. Patent litigation is expensive, but it is mainly because of the technical nature of the subject matter, and because of the safeguards against unwarranted monopolies that are provided in the patent law. In any event, a software copyright infringement action is also very expensive to prosecute, particularly in close cases where the scope of the copyright is in issue or where independent development is raised as a defense. The cost of obtaining a

U.S. patent is normally less than \$12,000, a very small sum compared to the research and development costs of a typical software product.

### **Myth Number 9**

*Software patents will tie-up the industry in litigation.*

First, software patents have been available for a number of years. Many developers already have obtained them. Yet, the software industry has not seen widespread litigation. Secondly, it is unusual that any one patent or group of patents can dominate an entire industry or technology. This is because a patent, unlike a copyright, has clearly defined limits so a competitor can avoid infringing designs.

### **Myth Number 10**

*To get a patent, a software developer has to disclose source code and other trade secrets.*

Source code need not be disclosed to the Patent and Trademark Office to obtain a patent. This would be the equivalent of requiring blueprints for mechanical inventions or transistor level circuit schematics for electrical inventions. Far from being a requirement, the Patent and Trademark Office actually discourages the filing of this kind of design detail. In contrast, to obtain a copyright on software requires disclosure of the source code. As for trade secrets, the software inventions that are most ripe for protection are those that cannot be protected as trade secrets, like a new product function or feature that can be readily observed by the use of the software. In addition, it is often only a small part of the product that needs to be disclosed. This disclosure can be made in block diagram or flow chart form to obtain patent protection. For example, to patent a carburetor, you do not need to show how to build the engine. Finally, software developers lose much more proprietary technology through employee job-hopping than they could ever hope to lose in a patent application.

### **Myth Number 11**

*You won't be able to detect infringement if the software invention is entirely "internal" to the system.*

It's a small world. In almost all cases word will get around if a competitor has misappropriated your patented technology.

## **Myth Number 12**

*Patents for software are not available outside the U.S.*

Software patents are generally available in every national jurisdiction. However, often, with different qualifications than in the U.S.

### **3.5.8 SAMPLE PATENTS**