So You Want To Sell Your Program?

Tips for Starting a New Software Venture

It is still possible to become a success in the software business. The trick is to find the right niche and avoid the amateur mistakes that doom so many ventures before they begin.

Getting Into the Market

The budding microcomputer software entrepreneur has two major options: (1) sell or license the software to an intermediary such as dealers, dealer representatives or distributors who in turn sell the software to the public; or (2) sell the software directly to retail outlets or to the public through advertisements, telephone, direct mail or online. The choice of options depends upon the nature of your product and the market you plan to sell to.

Selling to an Intermediary

If the software project requires a lot of development capital, it probably makes sense to consider developing the software and selling it through an intermediary. For example, if you propose to design a program to assist liquor stores in controlling their inventories, you might want to approach one of the major liquor producers. They might very well want to enter into a joint venture. In a typical joint venture, the software entrepreneur will transfer the rights to his program to the funding organization in return for a 5-20 percent royalty and payment of development fees. Or, the software entrepreneur may have developed a program to assist dry cleaners in keeping track of their goods. In such a case, it is possible that a large dry cleaning organization might be a sponsor. More recently, large software developers, such as Microsoft, have aided in the funding and development of interesting new computer programs.

In some cases, the software will have been developed and the entrepreneur can't or doesn't want to spend the time and effort to sell the software him/herself. In that situation, the entrepreneur might license the program to an organization like IBM or Microsoft. Alternatively, the entrepreneur may wish to contact any one of the numerous discount organizations that advertise in the software trade journals or online. The price that the entrepreneur pays is that the margin of profit is likely to be less when software is sold through discount distributors even though the volume of total sales might be greater.

Another source of distribution assistance is the large publishing and media houses. Many publishing houses have reluctantly entered the software business as a defensive measure to protect their eroding base in paper publications. Most publishing houses are hungry for good software ideas.

Unfortunately, a few of them, such as *Reader's Digest*, have dropped out of the software publishing business.

Selling Direct

Most software entrepreneurs are rugged individualists. They like to do things by themselves. While doing it yourself has the potential for greater rewards, it also carries greater risks. The opportunity to make amateurish mistakes at any early stage is greater when you do it yourself. Nevertheless, the dream of greater profit margins and the appeal of independence attracts many people. Software entrepreneurs can dramatically increase their chances of success in direct sales if they are well organized.

Start with a Business Plan

Without a doubt, drafting a business plan is one of the most difficult but certainly one of the most important steps in the formation of a software venture. A good business plan should be detailed and should at a minimum consider such things as: How big is the market for the software? What is the likely market penetration? What are the realistic costs of the venture? What are the cash flow projections for the next five years?

Drafting a good business plan forces the budding software entrepreneur to focus on the realities of the venture. While drafting a business plan can be difficult, there are sources of assistance. For example, a number of the major accounting firms will provide assistance at an early stage for little or no charge. Obviously, such firms are looking for a long-term relationship when and if the software venture gets off the ground.

Should You Incorporate?

The decision to incorporate is one that must take into account a number of factors. The preparation of the business plan will probably identify most of the elements that should be taken into consideration.

The major disadvantages of incorporating include the initial costs (typically between $500 and $1,500), the time and effort involved, possible additional federal and state tax on profits and the requirement of filing additional tax forms. The major advantages of incorporating include: limited liability, flexibility in raising funds, substantial corporate benefits (e.g., automobile usage, tax-free life insurance, retirement benefits, medical benefits), added prestige, etc. Limited liability is always a major factor since it can help insulate the software entrepreneur and his or her investors from personal liability resulting from losing a corporate lawsuit.

As a yardstick, many software entrepreneurs will not start a new venture and form a corporation unless they expect to receive their initial out-of-pocket expenses back within three years. Others use a guide such as exceeding $10,000 of expenses. Therefore, using these guides, if the entrepreneur expects to spend at least $10,000 in out-of-pocket expenses and there is better than a 50/50 chance that he or she will receive the $10,000 plus expenses back within three years, then it probably pays to incorporate. An excellent but slightly out-of-date guide is a book titled *Incorporating Your Business* by John Kirk, published by Contemporary Books, Inc. of Chicago, IL.

If the venture is to be incorporated and there is more than one individual involved, then it is advisable that the co-venturers have employment agreements with their company and that the shareholders have shareholder agreements among themselves. A pre-incorporation agreement is also desirable to clarify the legal positions of all parties in advance of incorporation. All agreements must be in writing. After all, verbal agreements aren't worth the paper they are written on! The agreements should anticipate all likely contingencies – both the good and the bad.

It is very common for small business ventures to fail, and when they do fail, a lot of grief can be avoided by spelling out in advance the manner in which assets will be disposed of in the event of dissolution. It is not pleasant to think of the possibility of failure while one is savoring the euphoria of potential success. However, experience dictates that those possibilities should be addressed to minimize heartbreak and the potential for litigation if the venture is not successful.

Incorporation is probably the most common form of doing business when promoting software. There are other modes that might also be appropriate under the right circumstances, for example: sole proprietorship, partnership, R & D Limited Partnership, etc. All of these alternative forms have their own distinct advantages and disadvantages.

For example, if you go into business alone and take no steps to form a corporation or partnership, the law will probably consider you to be a sole proprietor. The costs of going into business in this manner are relatively low; however, the risks are relatively high.

More recently, Limited Liability Corporations, referred to as LLCs, have become popular. They combine many of the best features of partnerships and corporations and have few disadvantages. They are, however, a bit more difficult to form.

Alternatively, an R & D Limited Partnership can be highly desirable where it is necessary to raise large amounts of funds. However, the costs of forming an R & D Limited Partnership are substantial. All options should be carefully considered during the business plan phase before a commitment is made to any particular form of doing business.

Marketing the Software

The software entrepreneur should have a clear idea at the very beginning of how he or she intends to market the software. The business plan should clearly indicate the target consumer and the manner in which the consumer will be reached. Certain types of software are easier to license to an intermediary while other types of software are easier to sell directly to retailers or to the ultimate consumer. For example, educational software is probably best marketed through established channels, such as textbook publishing houses, that already have distribution and sales networks. Spreadsheet, word processing, communications and database programs, on the other hand, are not as audience specific and therefore can be sold directly.

If the software entrepreneur chooses to license the software, he will find that licensing agreements will vary considerably from situation to situation. Royalty rates appear to range from 5-20 percent of net sales. If the software entrepreneur is dealing with a big organization, it is likely that the organization will have form agreements of its own. Generally, the bigger the organization, the less room there is for negotiation.

There are a number of critical terms that should be scrutinized in a software license. These critical terms include:

- Is the license "exclusive" or "nonexclusive?"
- What happens if the marketing organization sits on its hands?
- Will it pay a minimum royalty?
- Will it pay "upfront" money for development?
- What about documentation? Who prepares what?
- Does the organization have rights to future products that the software entrepreneur might produce?
- Does the license forbid the development or sales of similar programs to other organizations?

The financial implications of the proposed license should be explored with an experienced CPA, and the legal implications of the license should be explored with a competent attorney familiar with this area of the law.

The alternative to a licensing agreement is to sell directly to retailers or to the ultimate consumer. Where the software entrepreneur might get a 5-20 percent royalty for his or her efforts from a license, it is possible that he or she might get a 100 percent mark-up or better from direct sales. If the product is sold through an intermediate distributor, it is likely that the entrepreneur will realize a profit per unit somewhere between that available from a joint venture license and that available from direct sales.

Since the mark-up from direct sales is so high, why don't more people do it? The answer is that direct sales require a lot of effort and, if the software entrepreneur doesn't have the money, time or experience, distribution through an intermediary might be the smartest way to proceed.

Raising Money

The purpose of the business plan is to help the software entrepreneur organize in the most efficient manner. Another purpose is to impress potential investors with the viability of the project. A good business plan should make intelligent assumptions about sources of funding and cash flow. Many software entrepreneurs assume that they must raise large amounts of money ($100,000 to $1,000,000) to get their venture off the ground. This is not necessarily so. Unless it is absolutely necessary to raise large sums of money for the project, it is more prudent to start off as small as possible and bootstrap the venture to success by rolling initial profits back into the business.

The advantage of starting small is that if the venture is not successful, then the out-of-pocket losses are relatively modest. For example, there may be no market for the software, or a competitor might step in and take the market away. The half-life of success in the microcomputer business is relatively short; witness the fate of VISICALC® and Osborne Computers.

The budding software entrepreneur will probably not find initial funding for the venture from large, established, venture capital organizations. More often than not, large venture capital organizations are looking for minimum funding amounts in the neighborhood of $5 million or more. They are, however, excellent sources of finance and assistance after the venture is solidly off the ground and the entrepreneurs are looking for second or third round financing. It is also desirable to have an established venture capitalist on your board of directors since his or her experience and credibility can be very useful at the early stages of company development.

Local banks are an obvious place to start looking for funds. However, many local banks are uncomfortable doing business with high technologies. They would rather put their funds in established and conventional vehicles such as home mortgages, etc. Moreover, even if a local bank does provide funding, it is likely to be in the form of a secured loan. Therefore, the question frequently asked by the start-up entrepreneur is: Should I take a second mortgage on my house? Avoid doing so if you can.

More often than not, the software entrepreneur will find funding closer to home. Friends of the promoter are the most common source of funding for new software ventures. Your CPA is likely to know people who might be interested in investing $10,000-$50,000 in a new venture. Professional groups such as doctors, lawyers and accountants are always looking for good tax shelters.

Finding Professional Help

Comparison-shopping is the key to finding good professional help. Most new software businesses require the assistance of a trademark/copyright attorney, a business attorney and a CPA. The yellow pages of the telephone directory are a good place to start. The Internet has some excellent resources. But the best place to start is to ask other software entrepreneurs who they used when they got started. Word of mouth recommendations are generally the most reliable.

A list should be prepared of all possible professional candidates. After that, each candidate should be called and interviewed on the telephone. Questions such as the following should be asked:

- How much experience have they had with new software ventures?
- Do they have suggestions concerning the potential financing of your venture?
- Do they have contacts with the venture capital community?
- How much do they charge for services?

Your professional assistants should be brought in at the very beginning. You will probably find that they can be of significant assistance in helping you prepare the business plan, especially the CPA.

If your attorneys and CPA are identified in your business plan, it will give your overall plan a more professional image. In the software start-up business, what you don't know **will** hurt you. Good advisers can help you avoid serious mistakes.

How To Protect Your Program

If the business plan indicates that the venture has merit, then the next step is to nail down your property rights in the software.

Copyright Protection

Be sure that your software contains the correct copyright notice in the correct location. The copyright notice should preferably appear on the screen in the first frame of the presentation. A valid copyright notice includes three parts, namely:

- The word "copyright" and/or the international symbol "©." Some programs use the symbol "(C)." The use of the "C" in parenthesis, e.g. "(C)" should not be used without the word "Copyright" because there is some doubt as to its legal validity.
- The year date of creation, e.g., "2007."
- The name of the author.

For example, a correct copyright notice might read as follows:

Copyright © 2007 XYZ Software Company, Inc.
All Rights Reserved.

Or, if your software is not capable of generating the symbol "©," then the following copyright notice is also correct:

Copyright 2007 XYZ Software Company, Inc.
All Rights Reserved.

The term "All Rights Reserved" should be incorporated if possible to obtain the benefits of international copyright conventions.

The copyright notice should be visible in the first frame as it appears on the computer screen. It should also appear on the first page of the source code printout of the computer program in a form capable of being read and understood by the individual reading the printed out text. This, for example, might take the form of a REM statement in BASIC source code.

The copyright notice should be supplemented by a warning statement to discourage reproduction of the program by unauthorized individuals. A number of different warning statements are found in almost all software programs. Choose one and modify it to meet your needs. An example of a typical, acceptable warning statement is the following:

WARNING

THIS SOFTWARE IS PROTECTED BY UNITED STATES COPYRIGHT LAW (TITLE 17 UNITED STATES CODE). NO REPRODUCTION OF THIS WORK IS PERMITTED WITHOUT THE EXPRESS PRIOR WRITTEN CONSENT OF XYZ SOFTWARE COMPANY, INC. UNAUTHORIZED REPRODUCTION BY COPYRIGHT INFRINGERS MAY RESULT IN IMPRISONMENT, CRIMINAL FINES OR CIVIL LIABILITY.

Trademark Protection

The brand name, or "trademark," of the software program is often more important from the marketing point of view than the program itself. For example, some older word processing programs have developed a life of their own, despite the fact that more recent word processing programs are functionally superior.

Choosing a trademark often provokes a tug of war between legal and marketing considerations. From a marketing point of view, it is often easier to sell a product if the brand name tells you what the program does or who the intended audience is, i.e., "THE STATISTICIANS PLOTTER," "DIET MANAGEMENT," "VOCABULARY BUILDER." However, choosing a descriptive mark raises two potential problems.

The first problem is that there is a good chance that someone is already using an identical or similar mark on similar goods. Keep in mind that there are more than 2,000,000 federal trademark registrations, about 750,000 of which are still active, and many more unregistered marks. In contrast, there are only approximately 30,000 words in the active English vocabulary. Since there are only a limited number of words that can be used to describe any particular computer program, there is a very good chance that someone has already adopted the same mark if you choose a mark that is descriptive. Since rights in a particular mark depend, to a large extent, upon who was the first to use the mark in a particular line of commerce, the software entrepreneur may find himself changing marketing horses midstream should some prior user object to your use of his or her mark.

The second problem is that descriptive marks are very difficult to register with the United States Trademark Office. The Trademark Office has taken the position by statute that descriptive marks are **not distinctive** and that registration may not be obtained on the preferred Principal Register unless the public recognizes the mark as a brand name as well as a descriptive name. Typically, the Trademark Office will not agree that the required brand name recognition exists unless the descriptive mark has been in use for **at least** five years.

It is for these reasons that legal counsel usually suggests the adoption of an arbitrary or fictitious mark. Coined or fictitious words such as PARTRELLIS, BENDOSTAK OR LOORDS or words that have no relationship to computer products such as JAGUAR or PINWHEEL are much less likely to have been used before and are considered to be registrable without a showing of distinctiveness. They are also entitled to a broader scope of protection and their value tends to grow as product popularity increases. Eventually there comes a point where use of the name on any product will increase that product's marketability.

EXXON® and KODAK® are two excellent examples of successful fictitious marks. Unfortunately, when a company is launching its first product, it often needs all the marketing help it can get. This includes a mark that says something about the product, even if that something is not descriptive.

As a compromise, many entrepreneurs adopt marks that are characterized as suggestive. Suggestive marks say something that arouses the prospective purchaser's interest without going so far as to describe the goods. The descriptive marks previously mentioned could be changed to suggestive marks with just a little bit of thought. The descriptive term "STATISTICIANS PLOTTER" could be the suggestive mark "STATISTICIANS HANDLE." The descriptive term "DIET MANAGEMENT" could be the suggestive mark

"DIET GAME." The descriptive term 'VOCABULARY BUILDER" could be "VOCAB-U-ROBICS." These new marks still say something about the goods but don't raise as many problems as descriptive marks because they only suggest, rather than describe, the salient attributes of the software programs. Suggestive marks are stronger than descriptive marks but weaker than arbitrary and fictitious marks.

Once a prospective trademark is selected, it is critical that a professional trademark search be performed to determine if the mark is available. At a minimum, the marks in the U.S. Patent and Trademark Office should be searched. A search of the records of the U.S. Patent and Trademark Office should include not only all active marks but all pending trademark applications (since they can block subsequent trademark applications) and all abandoned trademark applications.

It is also prudent and highly advisable to search beyond the records of the U.S. Patent and Trademark Office. A thorough trademark search should include a review not only of the records of the U.S. Patent and Trademark Office but also all of the records of state trademark registrations (since trademarks can also be registered state-by-state), all of the business name (i.e., company name) records of each state and all of the marks that might be listed in Trade Directories but not otherwise registered with the state or federal government. It is important to search beyond the records of the U.S. Patent and Trademark Office because first users of unregistered marks can block attempts by subsequent trademark users to register their trademarks. A first user can also get a court of law to enjoin a subsequent user from using an infringing trademark.

If it is possible for a trademark owner who has a better date of first use but an unregistered trademark to block the trademark application of a subsequent trademark user, then why do trademark owners attempt to register their marks with the federal government? The answer is that registered trademarks are much easier to enforce. A federal trademark registration allows the trademark owner access to the federal courts and permits the imposition of substantial federal penalties. Federally registered marks are also easier to transfer if the software line or the entire software company is sold.

If the trademark search indicates that the mark is available, then it should be registered with the U.S. Patent and Trademark Office. To do so, the trademark must **first be used on the product** and the product must **first be sold** in interstate commerce **before** the trademark application is filed with the U.S. Patent and Trademark Office. Alternatively, it became possible in 1989 to file U.S. Trademark Applications based upon intent-to-use (ITU). ITU applications can be filed before actual use but will not be permitted to become U.S. Trademark Registrations until they are approved by the U.S. Patent and Trademark Office, published for 30 days and acceptable label specimens filed after publication. The U.S. Patent and Trademark Office requires three specimens of the mark as actually used. Label specimens are preferred.

Therefore, if the software entrepreneur chooses the mark XXX™ to identify the software product, then the following label might appear on the software disk:

Statistical Software XXX™

By

XYZ Software Company, Inc.
Copyright © 2007 XYZ Software Company, Inc.
All Rights Reserved.

The trademark notice should appear in the first frame of the program along with the copyright notice. It should also appear legibly in the printout of the program itself. The TM symbol is generally added on the upper right hand shoulder of the last letter of the mark to indicate that the mark is being used in the trademark sense. The TM symbol does not confer any legal rights but it does put the world on notice that you are using a particular mark as your trademark. Therefore, it helps to minimize ambiguity.

In no case should the ® symbol be used until after a federal trademark registration is received from the U.S. Patent and Trademark Office. The ® symbol can only be used if the mark is federally registered. Misuse of the ® symbol is a violation of Title 15 of the United States Code §1111 and could hurt your chances of obtaining a federal registration.

Can Software Be Patented?

Software programmers frequently ask if they can patent their software. As of this writing, the U.S. Patent and Trademark Office takes the position that patent applications directed towards software alone are not patentable. There are some exceptions. For example, if the software is only a small part of a larger overall system (e.g., an assembly line robot), then it might be possible to obtain a patent on the overall system. Novel algorithms, e.g., data compression or cryptography, might be patentable as method steps in a method patent. In general though, the software entrepreneur who only produces software would do well to concentrate on trademark and copyright protection, which is considerably less expensive and more readily obtainable than patent protection.

Trade Secret Protection

Trade secrets conjure up the murky world of private detectives. It is one of the least understood areas of our law. In a nutshell, trade secrets will protect object codes and source codes as long as that information is not made public. Once it is made public, the protection disappears. Therefore, trade secrets have their greatest value prior to the point at which the software is distributed to the public. The most effective way to protect trade secret rights in software is through the use of nondisclosure agreements and warning statements on proprietary information.

Nondisclosure agreements are written contracts between the owner of secret information and the user or recipient of that

information. The software entrepreneur should have nondisclosure agreements with **everybody** who comes into contact with the software prior to its initial distribution.

It is advisable to have written nondisclosure agreements with all employees, all outsiders and all end-users who may come into contact with the software prior to mass distribution.

Frequently, nondisclosure agreements are incorporated into the employee's general employment agreement. It is not necessary to have a nondisclosure agreement with an attorney since the software entrepreneur is automatically covered by the attorney/client privilege, which prevents the attorney from discussing anything disclosed to him or her in confidence. A word of caution: Be sure your former employment agreement doesn't prevent you from competing with your former employer. This is a trap that frequently catches the unwary. Review your previous employment agreement with your attorney at the beginning.

Licensing Protection

The software entrepreneur should attempt to establish a license agreement between him or herself and any end user of the program. Basically, there are two types of end-user license agreements.

The first type is an agreement that is not signed by the end user. These agreements are known in the trade as "shrink-wrap," "box top" or "tear-me-open" agreements and there is considerable legal doubt if they are valid and binding. A "shrink-wrap" agreement typically includes a self-serving statement such as – if you open this package, you, the end user, will be deemed to have agreed to the terms of the enclosed software licensing agreement.

The second type of end-user licensing agreement is one that is signed by the end user. Such agreements are considered to be valid and binding on both parties. The trick is how to get the end user to sign such an agreement. If your software package has a very high price and a very limited audience, it is likely that you will come into face-to-face contact with the ultimate end user. In that case, it is much easier to get the end user to agree to sign the licensing agreement. However, if the software package is a low price, mass-market item, it is very difficult to get the end user to sign such an agreement.

Obviously, the software entrepreneur can't station a representative in each computer store in the country to make sure that all purchasers sign the licensing agreement. Such activities would obviously discourage potential end users. Some manufacturers have the end user sign a card that is returned to the software producer and entitles the end user to updates or enhancements of the software as they are produced. If the language on the card is appropriate, it is likely that the courts would find such a licensing agreement to be valid.

Fox Rothschild LLP
ATTORNEYS AT LAW

For more information contact:

Fox Rothschild
Princeton Pike Corporate Center
997 Lenox Drive – Building 3
Lawrenceville, New Jersey 08648
Tel: 609.896.3600
Fax: 609.896.1469

© 2012 Fox Rothschild LLP. All rights reserved. This publication is intended for general information purposes only. It does not constitute legal advice. The reader should consult with knowledgeable legal counsel to determine how applicable laws apply to specific facts and situations. This publication is based on the most current information at the time it was written. Since it is possible that the laws or other circumstances may have changed since publication, please call us to discuss any action you may be considering as a result of reading this publication.

ATTORNEY ADVERTISEMENT