

Application Software as a Long Term Investment

**Arthur J. King
SOTAS, Inc.
192 Merrimack Street
Haverhill, MA 01830**

Introduction

The purchase cycle of financial software has evolved considerably over the last decade. It has become a complex and sometimes confusing period for the purchasing company. More and more, people are realizing that the software they are purchasing should be viewed from the long term as opposed to the short term, "How do I solve today's problems?", perspective. This session will examine this evolution as well as the importance of looking at the "*big picture*" when reviewing your software needs. The four steps necessary to ensure a solid, long term investment: 1) Investment Planning, 2) Deciding on the Right Investment, 3) Protecting Your Investment, and 4) Reaping the Rewards of Your Investment will be discussed in detail. This session will also cover the importance of having a close *working* relationship between the software purchaser and the software vendor to ensure an investment that will meet their current needs and create a solid "Foundation for the Future".

I. Evolution of the Financial Software Industry

In the late 1970s, the financial software industry had just begun to get its feet wet and was experiencing its first growing pains. Many software companies were begun as one system shops. If the user wanted to purchase Accounts Payable and General Ledger, for example, the user would probably buy two different products from two different companies. This was because certain companies had established a reputation for having a good Ledger, yet they would have either no Accounts Payable or a weak AP system at best to offer. The only way to come up with a satisfactory solution for their organization was to purchase the applications separately. This meant that there were two different communications channels, two different support organizations to call, two differently designed systems and worst of all there were two separate learning curves. Unfortunately, just because the user knew how to logon and run jobs on the General Ledger system, the user would still be a novice as far as

Application Software as a Long Term Investment

0071-1

the Accounts Payable system was concerned. This became an even more frustrating situation to the user when they realized that there were many custom changes made to their system that were not documented or supported by the vendor. These changes occurred because there were usually some features or functions that were missing from the "standard" system. As the vendor had to survive on the basis of one application, they would generally try to add a quick fix to that client's system. This patch might address the client's initial problem, but as future releases of the system were sent, these patches would become outdated. The vendor might have a record of these fixes, but even if they did, it was often a lost cause trying to patch them into the new system. Usually, the user was left with two choices: 1) reinvent the wheel by rewriting a new patch, or 2) give up on the new release and struggle with the old one. Despite this gloomy situation, many users were satisfied with their situation mainly because there was no alternative. The other problem came in the area of bridging information from one application to another.

Many users wanted the information from their sub-ledger products such as Accounts Payable and Accounts Receivable to be passed automatically to the General Ledger system. With two separate applications, the only way to pass information to the General Ledger system was to either key the information in manually from an Accounts Payable report or to write their own bridge program. If the user chose the latter option, they had to carefully monitor any new releases to either system and make changes to their bridge program accordingly.

As the software industry evolved, the vendors realized that they could no longer survive as a one product company. The survivors began to expand their product breadth, in one of two ways. The first method was to establish a separate task force that was out of the mainstream so that they could develop a system faster and without interruptions. This procedure was good enough to develop their first module, so they felt that it should work for any other module. The problem with this method was that it was slow and although the people could develop a Fixed Assets system, it didn't guarantee that they could develop a reasonable General Ledger system. The second method was to acquire the software from some other vendor with the promise to make it different and to not compete with that vendor with the same product. The problem with this approach is that the vendor had very little knowledge of the application that they were supposed to be selling and supporting. In either case, the second and subsequent systems produced by the vendor seldom bore any resemblance to the system on which they had built their reputation. This was because styles are different and the separate development group had their own style just as the company that originally developed the acquired software had theirs.

Many vendors ended up choosing the acquisition route because it was faster, and to some extent, the software that they were purchasing had already proven itself in the real world. This meant that the product line offered by a vendor was stronger in that they now had more competitive products to offer. In fact, to this day, the reason that some applications look alike from vendor to vendor is because of the rampant interbreeding of some applications.

Application Software as a Long Term Investment

Product Line in Late 1970s

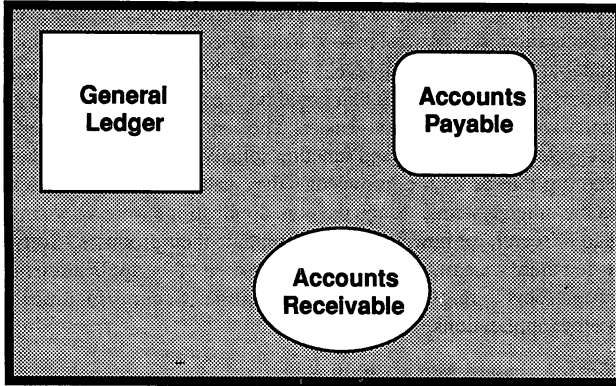


Figure I-1

With the situation changed at the vendor level, the user had some extra choices to make when they decided to purchase software. Although Vendor A's General Ledger was stronger than Vendor B's, Vendor B's Accounts Payable system might have significantly more bells and whistles than Vendor A's. The user would then have to decide whether the advantages of having the "best" systems (buying one from each vendor), were outweighed by the advantages of purchasing both systems from one vendor. There would be some cost savings through purchase price discounts, multiple product support licenses and the ability to contact one organization for all their software needs. However, there was still the problem with the learning curve, because even if both systems were from the same vendor, they probably looked different. The vendor might possibly build the bridge between the sub-ledger systems and the General Ledger system for the user, and if the user was persistent enough, they might even maintain these bridges as new releases came out. To some extent, there was still the problem with customization as many users pressured the vendor to add features to the system so that they would not lose as much by purchasing from one vendor instead of purchasing the two best systems. Although more and more users were switching to a one vendor solution, there were still many users who preferred dealing with multiple vendors to get the best solution.

As the 1980s progressed, the software industry matured even further. Many vendors realized that the key to increased sales was through two avenues. One was to improve user friendliness and the other was to make the entire product line more similar. This increased the likelihood that the user would purchase more than one software product from the vendor. As the features and functions became more and more similar between vendors, the buzzword

Application Software as a Long Term Investment

0071-3

became “user friendly”. In other words, systems were finally being designed so that they could be used by the end user without heavy involvement from the data processing staff. This meant that documentation, screens and procedures had to be redesigned in such a way that it was understandable at a nontechnical level. Another change to improve user friendliness was the increased use of common tools that could be used by more than one application to make the data entry and reporting options easier. The vendors also established some design standards so that systems would look and function somewhat similar. This helped the user reduce their learning curves as the knowledge they had on one system was somewhat transportable to the other systems. Since the core applications were very similar between vendors, they had to search for new ways to differentiate their products. Software vendors began to realize that there were many areas outside the *core* of the application that they could focus on. They began developing Query/Reporting facilities, Security facilities, Links to PCs and hooks into other applications.

Product Line in Mid - 1980s

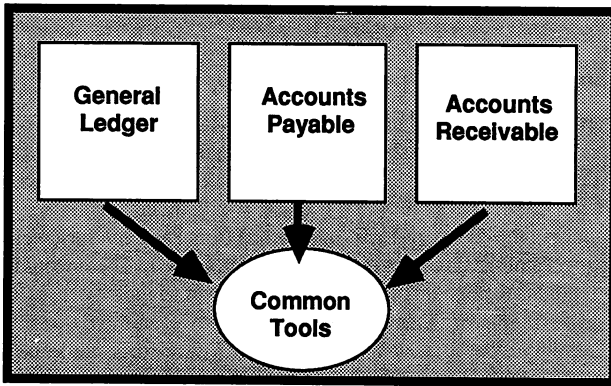


Figure I-2

This latest step in the evolution cycle was a welcome change to the end user. Firstly, they no longer had to sacrifice major feature/functionality when choosing between vendors. When evaluating software, the user could spend more time focusing on how the software fit in their environment from a usability standpoint. This change in focus put a much higher priority on the user friendliness of input screens, the flexibility of reporting and general ease of use of the applications. Also, the user did not have to be as concerned with software customization because, with the wide range of choices available, they no longer need a quick fix to meet their needs. Most of the changes could be made external to the application or through the use of the hooks that the vendor had established. It was becoming more and more advantageous to purchase all of the financial systems from one vendor because of the reduced learning curves that resulted from the similarity of products as well as the ease of having one

Application Software as a Long Term Investment

contact for all the user's systems. It was also a real time saver if the user was interfacing sub-ledger products to the ledger. The interfaces were already written and supported by the software vendor. There were no longer any worries that when a new release came out the systems would not interface with each other. In summary, there was a much wider and more complex choice of solutions for the user. The advantages of going with one vendor were very clear; the concern was in choosing the right one. One way to help clarify the decision is to take a look at what the future of financial software applications.

In the 1990s, the evolution of the financial software industry will focus even further away from the individual core applications and it will focus on integration and other external factors. The user of financial software in the 1990s will also be focusing on many nonapplication specific areas. These changes will be in three areas: 1) User Tailored Drivers, 2) Integrated/Shared Databases and 3) Comprehensive Services.

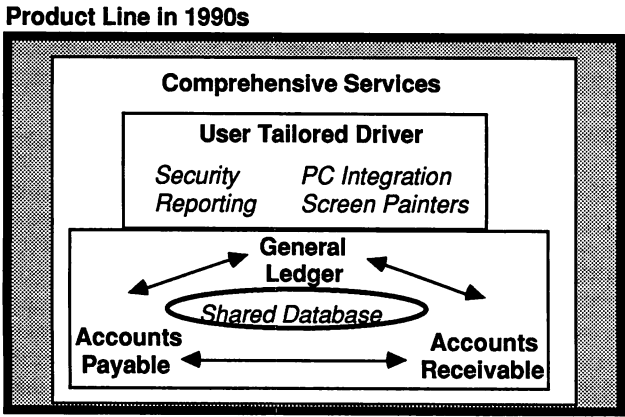


Figure I-3

In the area of user tailored drivers, the user community will demand and the software vendor will supply more PC oriented applications that work as an integral part of the main application. This will allow the user who has invested in PCs and LANs to be even more productive with the tools that they are already using. These PC modules may serve as an entire subsystem or they may serve as a tool to enhance an existing application process. The end result is that it will make the applications more accessible to more users. Another area that has already received attention is the area of security. As the software application has been opened up to a wider range of users, there has been a need for tighter control over who can access what information. As the applications become integrated even further, there will be a need for user defined security down to the level of restricting certain users from accessing

Application Software as a Long Term Investment

certain fields while performing certain tasks. These security modules will reside over all applications and will allow the user to have complete control over the level of security that they are looking for. As always, there will also be a need for completely flexible reporting and inquiry tools. Software vendors will enhance existing canned reports to allow for greater flexibility from both a format and a data selection standpoint. There will be a change in philosophy in regards to report writers with a greater dependence on tools that already exist that can access any application. Many software vendors will turn away from writing their own report generators and instead integrate their software with the report and inquiry tools that the users are already using at their site to access their existing applications. Another important piece of the User Tailored Driver facility will be the screen painters. These screen painters will allow the user to tailor all of their screens, whether they are input or inquiry screens, to their environment. The software vendor will supply skeletons of preprinted screens with prelabelled fields. If the user wants to change these labels so that they are more meaningful, it would be a simple matter of entering the new name into the system through the screen painter facility; and that new field label would be displayed throughout the system. These User Tailored Drivers allow the user to custom tailor the system to their specific needs without modifying the system. This drastically improves the maintainability of the system for both the user and the vendor.

In the case of integrated databases, this would mean a fundamental change in the way that both the user and the vendor view financial applications. Currently, a financial application is considered a separate module that performs a specific function; such as, General Ledger, Fixed Assets, or Payroll. As the applications and sophistication of both the user and the vendor has evolved during the last decade, the borders between these applications are starting to be broken down. As you have already seen, there has been a trend towards making applications look similar, integrating applications and the sharing of application tools. This has allowed the user to easily navigate between two separate applications. However, if you step back and take a look at the whole picture, you will see that each of these so called modules, you will see that each of them is really a function of the entire financial process. If, for example, your company was to purchase a machine that manufactured widgets, and you produced those widgets for wholesale widget distributors, you might step thru five separate functions which would encompass five separate financial systems. First, you would write a purchase order for the widget machine using the purchase order system. Next, you would enter the invoice for the machine on the Accounts Payable system. You would then have to set up the machine on the Fixed Assets system for depreciation purposes. After the machine is up and running, you must keep track of the manufactured widgets thru the Inventory Control system. Orders for the new widgets are entered thru the Order Entry system and finally all billing for these orders would be processed thru the Accounts Receivable system. There are six different but related accounting functions which would require journal entries to the General Ledger system. In the financial system of the future this would be handled by six functions within *one* umbrella financial application sharing one database. Some of these functions would be automatically generated for the user instead of

Application Software as a Long Term Investment

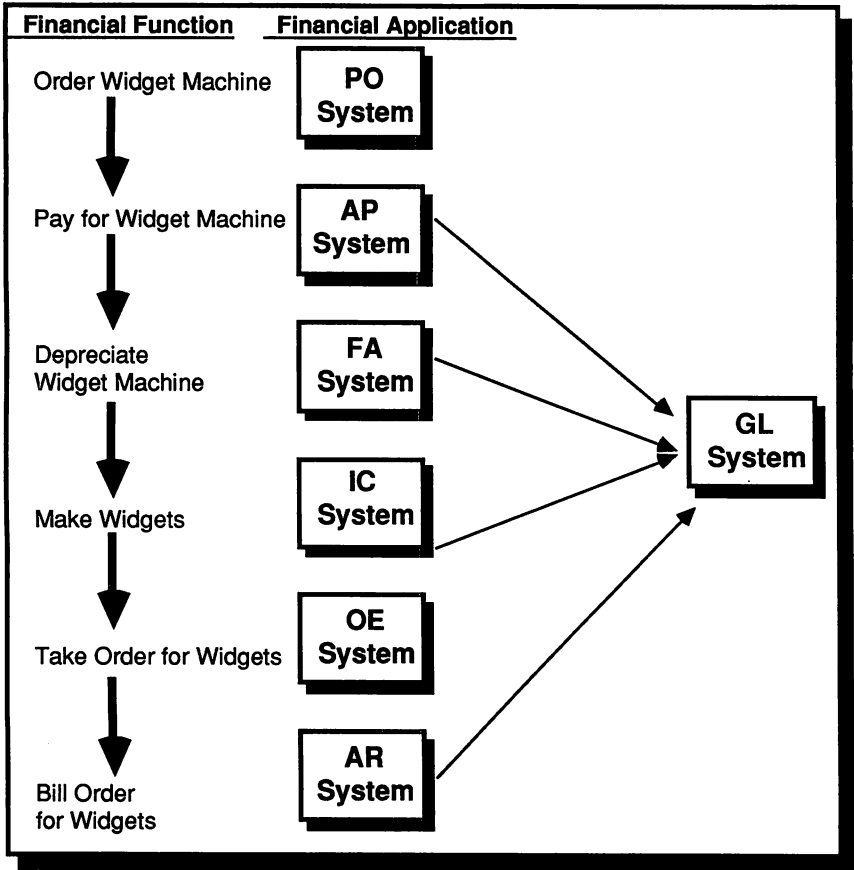


Figure I-4

being manually entered thru six separate systems. The redundancy of data would be significantly reduced thru the integration of the financial software database. In short, the user would have increased control and simplified input of information. It would help streamline the entire financial function.

The third change that will take place in the product line of the 1990s is the increase of built-in and add-on services from the vendor. The one area of financial software that has not yet evolved is the area of software services. In many cases, the only support that a client may receive from the vendor is telephone support. Any major new enhancements may be at an additional charge. The software vendor of the future will see this as an area in which they can offer a major advantage over the competition. These new services are already starting

Application Software as a Long Term Investment

to take hold as users are demanding a better value for their maintenance dollar. These services could include *Comprehensive* support which covers the user from installation and planning thru custom enhancements and upgrades. Phone support will be augmented by dial-in support and increased use of electronic mail. This will speed up the problem solving process. Systems may be shipped with self-diagnostic modules that will assist the user in pinpointing and solving the problem. There will also be a noticeable change in the relationship between the user and the vendor. There will be an increased sharing of ideas. Vendors will work closely with their user base to plan and develop new ideas. Users will share their ideas and modifications that they have made with the vendor. It will no longer be an *us against them* relationship as both the user and the vendor will realize the benefits of working together. The user will be able to get the enhancements that they want into the system, resulting in a happy customer reference base which will subsequently enhance their sales. It will truly be a joint relationship for the good of all.

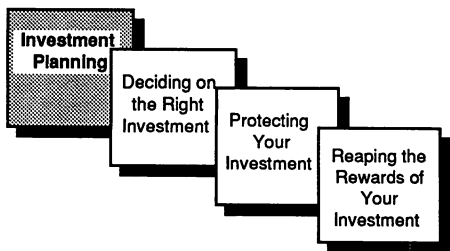
Now that we have examined the evolution of the financial software industry, let's take a look at how this impacts the software purchasing cycle.

II. Ensuring a Solid Long Term Investment

The evolution of the financial software industry has drastically changed the way in which users purchase software. In the 1970s many users went through a limited evaluation. This was because of the reputations that had been established by the software vendors. Unless there was an extraordinary need, there were certain vendors from whom users knew they should buy certain software. As the industry evolved, the purchase cycle became more complex and more involved, users began to do more research into which product or products would best fit their needs. They began talking to other users, reading documentation and having demos. As the product lines evolved even further, it became even more critical to review in detail all of the factors prior to making the purchase decision. As users began to view the purchase of software as a long term investment, it became important to take a calculated step-by-step approach to making and nurturing that investment..

The rest of this discussion will focus on a four step approach: 1) Investment Planning, 2) Deciding on the Right Investment, 3) Protecting Your Investment and 4) Reaping the Rewards of Your Investment.

Application Software as a Long Term Investment

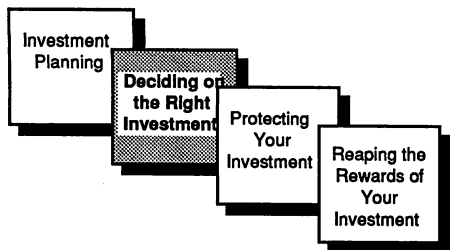


III. Investment Planning

The first step is Investment Planning. At some point prior to making the purchase decision, you should go through a cost analysis to determine how much it would cost to develop and support the application in house versus purchasing the software and the support from a software vendor. Generally speaking, you will find that there are significant savings, especially in the area of maintenance, by purchasing the application. This will give you a cost bench mark to use for comparison purposes. It is very important to decide what your needs and priorities are *before* getting seriously involved in the purchase decision. For example, before shopping at a car dealership, you would first determine your basic type of car preference and which options were high priorities.

To do this you must assemble a team made up of users, accountants, data processing staff and other appropriate decision makers. This will help assure that everyone that is affected will be represented in the decision process. It is also important to spread the responsibility of the research amongst several members of the team so that not just one or two people bear the total responsibility of the decision, especially in the early stages, when you should get as many different opinions as possible. Next, a list of your needs from hardware to detailed features and functions and from service, to future plans should be compiled. Each of these detailed items should be prioritized, and then you should determine the vendors that could possibly meet those needs. The avenues that can be pursued to come up with a long list of potential vendors include scanning directories, talking to other companies that you know have financial software, getting information from your hardware vendor and talking to other users at a hardware user conference such as this one. Ask for brochures and as much information as you can from the vendors and have the team review the materials. After you feel comfortable that you have reviewed enough materials from enough vendors, you should narrow your focus to the best three to five vendors; this will become your short list. One more step to deal with before going forward with your short list is to project where your company will be heading in the next five to ten years. Try to forecast what your accounting needs will be during that period so that you can select a solution that meets your needs now and in the future. This would include examining your hardware plans, usage of PCs and any other needs that you can envision. After you have completed this, you will be ready to proceed to the next step of the decision process.

Application Software as a Long Term Investment



IV. Deciding on the Right Investment

Now that you have narrowed your potential solutions down to a short list, it is time to get down to the serious business of deciding on the right investment closely examining each of the options. The first step to take with your short list is to assemble the list of your needs and to produce a document generally referred to as a Request for Product (RFP). Next you should meet with the vendor's sales representative and have a discussion that covers such topics as an overview of the vendor's company, the vendor's goals and philosophy as well as a discussion on the user's goals and objectives. It is also the point in time where the RFP should be given to the software vendor's sales representative. You should give the vendor a reasonable amount of time to review the RFP and return a written response to you for your review and you should also schedule a follow up meeting for a detailed presentation. After sitting down and evaluating the responses and comparing them to the priorities that you have determined for each issue, you should determine what follow-up questions must be asked in the detailed presentations. It is important that the detailed presentations be conducted in two parts; the first part being a detailed overview of the system with the second being a demo. The detailed presentation should be scheduled at a time when the entire team can attend and it should be uninterrupted time so that all team members can attend all aspects of the presentation. Questions from the RFP that needed further clarification should be asked at this time.

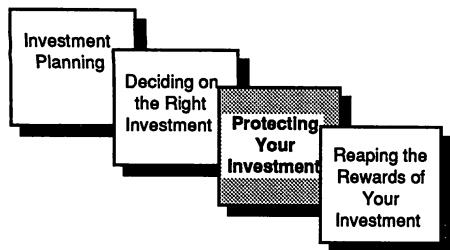
This are also two critical pieces of information that should be addressed at the detailed review.

First, obtain a list of some of the vendor's users that you can talk to and find out their impressions of the software, services and the performance of the software vendor. It is usually a good idea to talk to the chairperson of the vendor's user group. This will tell you whether there is a strong relationship between the vendor and the user community and if there is a user group. A good sign of an investment that will grow with you is if there is already a mechanism in place for the users to work with the vendor on ideas. The chairperson is also a good sounding board for the mood of the entire user base as opposed to the narrower focus you might get from one user. You also will usually get a less biased picture from the chairperson than you would get from a vendor selected reference.

Application Software as a Long Term Investment

Secondly, review the future plans of the company for its product line. This will help you determine if the vendor has taken the time to sit down to plan for the long term or whether they are still focussing in on the short term. When I say long term, I am talking three to five years down the road, not just plans for the next release. This will give you a chance to see if the vendor is a good *partner* for the long term and whether the vendor's plans and your long term plans are along the same line.

After completing the detailed review it is time to make the final decision. Using the information from the RFP, the detailed review and the users' references, it is time to come to a decision and negotiate a final contract. If you have followed the above process, it should become quite evident who the final choice should be. If there is any question at all and you are at a toss-up between two vendors, you should go with the vendor that appears most willing to work with all of its users and who appears to be positioned with the best future plan. This is not to say that you should buy "vaporware". In fact, you should be able to predict which vendor will have the least likelihood for "vaporware" selling by who has spent the most time determining where the future of the industry lies. Usually, the vendor who has enough confidence in what they are doing that they does not have to sell future products because they have committed to a long term goal. That goal would be damaged if they began promising pieces of it before they could deliver them. Also beware the vendor that promises to complete all of the development projects that you are interested in this year and is willing to shift their development plans just for you. This vendor probably makes the same promises to all of its prospective users and never delivers once the contract is signed. Look for the vendor that is willing to give an honest answer even if it might not be what you want to hear because that vendor is more likely to deliver in a timely manner. The bottom line is that you are making a long term investment and it is important to take the time to make sure that you leave no stone unturned and that you have all the information that you need to make the right decision. The main difference between a short term solution and a long term solution is in what you choose to do after the purchase.

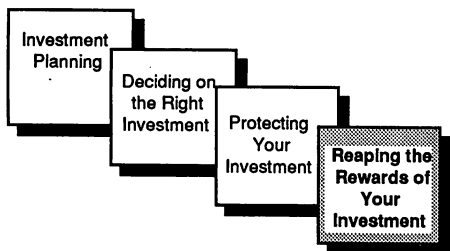


V. Protecting Your Investment

The investment cycle does not end with the purchase of the software. If you were buying a car, you would take the time to tune it up regularly, change the oil and performed other maintenance tasks designed to make your car last longer. This is the same idea that you should follow for your financial software. It is especially important to get started off on the right foot. This means taking the time to sit down with the vendor to plan an implementation schedule that includes training on the systems for all the individuals who will be involved with the system. This would include accountants, data entry clerks and even the management staff that will be requesting certain financial reports. The more everyone understands how the system works, the easier it will be to get things done. The implementation schedule should be spread out in such a way that different systems are being installed at different times because chances are that many of the same people will be involved in each system. As I said before, each application is merely one step in the financial function, not a totally different function that would be handled by totally different people. It is also helpful if you set up a procedure for calling the vendor's support. It is generally a good idea to have the same one or two people making the phone calls so that both parties get a chance to build a strong working rapport. Once your applications have been implemented smoothly and you have established a procedure for calling the vendor, it is tempting to just sit back and take the changes as they come. You must become an *active* rather than *passive* participant in the process if you want to protect your investment. You should subscribe to vendor or user newsletters to keep on top of what is happening. You should immediately become an active participant in the user group of that vendor's users just as you have taken the time to come to this conference. Many valuable pieces of information are learned and shared at conferences such as this. First and foremost, you get to meet other users from other companies and learn how they use their system. It is a great opportunity for discussing "what if?" ideas with your peers. It is also an opportunity to influence the future direction of the vendor's product line. Software vendors should welcome the chance to work *with* their users. It is to everyone's benefit, especially the vendor's, if everyone is pulling together in the same direction. These user groups generally have some form of committee that acts as a planning staff with the vendor as the vendor begins planning new releases. If you want to protect your investment, then you

Application Software as a Long Term Investment

must be a participant in this process. As I stated earlier, the software industry is evolving to the point where there is an intense sharing between the vendor and the user. If you do not make the effort to share ideas, then you are taking a risk that your investment may become outdated because your needs were not addressed.



VI. Reaping the Rewards of Your Investment

As is the case with any investment, the more you put into it, the greater the return. If you take the time to sit down and analyze both your current and future needs, review all available options prior to making the purchase, and become an active participant after the purchase, the rewards can be great. You will reap a system that will meet the demands of your company and will build a solid relationship with your vendor. More importantly, you will make an investment that can grow with you and change to meet your needs as you change. You will end the dreaded cycle of having to trade in your software every few years because it no longer meets your needs. Granted, it takes a commitment of time and people up front, and it has to involve many resources. However, if you take the time to do a thorough job up front, there is no reason why you will have to trade in the software again.

VII. Conclusion

In summary, it is important that you look at the long term ramifications of your investment decision. Take a look at where the industry and your company are headed and determine if the vendor is prepared to handle any new changes. . Verify the fact that the vendor is willing to work *together* with their client base. Explore every option prior to the purchase, but don't let it end there. Like any other worthwhile relationship, you must constantly work at it to keep it strong. If you sit back and let the events happen without providing any direction or input, you will only have yourself to blame if the software does not meet your future needs. Take the time to make an investment of your time in the software investment process and you will have built a solid "Foundation for the Future".