A Software Purchase Process

Neville Turbit

Overview

Buying software can be a daunting task. Any software purchase is fraught with danger. Companies have collapsed after the wrong software was purchased. There are many cases of companies that lost their way because stock could not be managed, invoices could not be produced accurately or on time, or financials could not be managed.

This white paper covers, at a high level, the process you should use when starting out to purchase software for your organisation. There is no "one size fits all" but there are some principles and activities that should be undertaken in almost every situation.

Internal Support for Software Purchase

Ensuring people know what is going to be involved, and are fully supportive of the effort to purchase and implement software is critical. I do mean critical in the sense that if it is not in place, you will likely die. Many executives believe purchasing and implementing software is just a touch more complicated than buying and installing a shrink-wrap piece of software on their home computer.

They need to understand that:

- Selection will require a rigorous process involving a number of people over a number of weeks or months
- Implementation will probably cost more than purchase
- Implementation will involve their experienced people being seconded to the project for a period of weeks or months
- Business processes will need to change
- Customisation may turn out to be an extremely expensive exercise and may still not deliver exactly what they want

Resource Impacts

I was contacted by a CIO on one occasion and asked how he could stop a potential ERP purchase. He knew the organisation did not appreciate the disruption an ERP implementation would cause. He was concerned they would be sold a package then find the installation took longer, cost more, and failed to deliver what people wanted.

We hatched a plan which resulted in a presentation to senior management. We identified roles and responsibilities for the implementation. We then identified the skills required to fill those roles. The skills narrowed the candidates down to a group of eight or ten mid to senior managers. We presented a plan to 'kidnap' them for a period of six months to the project. We even identified how they would be replaced. In some cases this was by promoting others, and in some cases it was with external appointments for the period.

As the meeting progressed, there was increasing discomfort at the business impact, and the cost. The CEO wanted us to find other people. We asked the exec team who had the required skills? As an example, we had nominated the Manager Accounts to work on the financial aspects. When the executive team said they could not release him for a period of months, we asked who was authorised to design a new chart of accounts and map existing accounts to the new accounts. The answer kept coming back to the Manager Accounts with the CFO to sign off.

Typically the organisation did not track internal costs of business staff on projects. We did a calculation and presented it to the team. That also caused a sharp inhaling of breath. In the end the company decided to reconsider the ERP implementation in 3 months whilst some more work was carried out to quantify the impact. Three years later, the idea had all but faded away. A couple of years later, the decision was taken but it was approached as one of the most significant changes the organisation would ever make. It was properly resourced, and planned a year in advance.

Go Looking

Assuming your organisation is convinced it will commit to the effort required to implement a new package, the next decision to be made is:

- Do we go looking at solutions first and then work out requirements?
- Do we work out requirements before we go looking for solutions?

Having been down the track a few times, the first question I ask is how much you know about the software you are looking for? For example, if your organisation has no knowledge or experience in web services, and that is where you are going, you probably need to bring some people up to speed. The vendors are a way to do that, but keep in mind they will show you what their software can do. They won't show you what web services can do. If you look at a number of vendors, you will get a better view, but it is not a particularly time effective way to improve your knowledge. You will probably see the same thing over and over and over.

A better solution is to bring in an independent expert. Run a one day workshop with one or more experts to bring people up to speed on the area. Find out what is potentially applicable to your business. You don't even have to mention that you are in the market. Once the word gets out, the phones will start ringing so keep it quiet if you can

Macro Requirements

I have seen people develop extremely detailed requirements before going to market. It usually results in disappointment that there is no perfect solution. A better way is to agree macro requirements. These are a minimum of 5 and a maximum of 20 "must-haves". If the package does not have these attributes, it is not going to make the cut.

Macro requirements are not limited to functionality. They might relate to operating systems, cost, time to implement, compatibility with other software, local support, installations in your industry, access to source code, vendor size or any other aspect of the software. Use a workshop with all key stakeholders to develop a list.

Do a scan of the market and identify all potential packages. Review each package and cull those that do not meet your macro requirements. In many cases, you will not even have to speak with the organisation. If ability to run on Unix is a macro requirement, you can quickly establish from the web if the package will run on Unix.

Initial Demo

Having got down to a shorter list, decide who will form part of the selection team. Also decide what aspects of the software you are particularly interested in seeing. These will typically be functional and may include what people see as their "hot buttons". The things that key stakeholders are particularly interested in having available.

An aside here. I have owned a number of boats over the years. When looking to buy one boat, a wise old owner said to me that he had owned about a dozen boats during his life. He said it took about six or seven before he worked out why he was buying a new boat. His first was a small yacht so he wanted a bigger yacht. His second was bigger but he had not foreseen the maintenance involved in a wooden boat so he bought a fibreglass boat. That boat was too slow so he bought a faster boat. Unfortunately that boat was not particularly stable for pleasure cruising. It was then that he realised the main driver for the next boat was always the biggest problem with the current boat.

Relating this to software, be aware that if your current system takes all night to run a backup, that you are not putting undue emphasis on being able to do a backup in 4 hours. Is this really the dominant feature you are looking for?

Demonstration Agenda

It is now time to look at the software. Make sure you drive the agenda of the demo. Take the time to put together an agenda that covers what you want to see. That should take 60% to 70% of the time available. Typically this first demo should run between 1 and 4 hours. If each vendor follows the same agenda, it will make it easier to compare different vendors.

Contact each vendor and ask them to provide a demonstration. It can be at your premises or their premises depending on the product. If you do it at your site, there are potentially issues of connectivity, performance, and configuration. On the other hand, a vendor site is going to run like clockwork which may not be indicative of how easily the software will be to run in your environment.

Make it clear that you have a number of things you want to see and they will need to spend time with you to go through the agenda. I have found they can spend more time with you planning the presentation than actually delivering the presentation.

It is also worth giving them a profile on the attendees and where their interests lie. Also their level of knowledge. For example if most are relatively technologically savvy, then the presentation can take on a more technical flavour. As I said, the agenda should cover 60% to 70% of the time available. This will provide the vendor with the opportunity to cover things that they want to show you.

Also make it clear to the vendor that there are two conditions to the demo.

1. The cover what is in the agenda
2. All future contact with the organisation is through you.

If the vendor is not prepared to meet these requirements, they should not be asked to participate in the initial round of demos.

The Demonstration

Provide attendees with a one or two page sheet to fill in regarding the key points in the agenda. By the mid point of the demonstration, you should check the agenda items already covered, and if the demonstration is not progressing through the agenda, re-direct the vendor to complete the agenda in the remaining time. Failure to cover agenda items may mean there is a weakness in the solution. It is also indicative that the vendor is not prepared to work in a cooperative manner with you.

Another trap with demonstrations is that the vendor will fragment the team and try to attract particular people to another demonstration of an area they are particularly interested in seeing. The vendor is trying to establish as many links into the company as possible, so make it clear to attendees that this is a first look.

If you have an independent consultant working on the project, you might want to use them as a part of the team to raise any questions where the vendor is trying to gloss over a particular area. For example, if you have on your agenda to look at connectivity with another package, the vendor might say “Yes. We can interface.” Perhaps building an interface will be a lengthy process resulting in a clunky exchange of data. Your independent expert should know about this and ask more probing questions.

There is another danger where a member of the team does not attend all demonstrations. Suddenly you have a decision being made by people who were not present. This can cause significant bias. Try to ensure enough lead time to make sure everyone is there.

Post Macro Evaluation Cull

After the demonstrations, gather the attendees and determine which are the best fit, and which you want to proceed with. You should have a goal of anywhere from two to four packages at this stage. If they are very close, you might decide to proceed with two or three and keep the others in the running should one of the selected ones fail to deliver in the next phase. I have had this situation happen in the past when working with clients.

Micro Functional Evaluation

Having established a short list, we now move on to the micro functional evaluation. In order to do this, there needs to be a BRS (Business Requirements Specification) in place. The work on the BRS can begin earlier in the process however it should be kept in mind that the requirements are likely to change slightly when people see the demonstration. For example, the demonstration may illustrate a better way to undertake some functions. It may provide additional functionality that was not considered within the scope of the original requirements.

Requirements should be developed down to the point where they can be evaluated against the package. To take an example, if the function was to maintain a client, the requirements may be for the ability to set up and modify a client and record multiple addresses and phone numbers. The ability to store the following client information:

- Name
- Address
- Age

- DOB
- Etc.

Think of it as a checklist that the vendor will be evaluated against. In this sense, you may use a slightly different format to collect the information than if you were building the application.

The Micro Functional Evaluation

It is likely that a number of business people will be involved in different functional evaluations. For a financial system, your Accounts Receivable people may have a list of functions to check, and your Financial Accounting people another list. Similarly, the vendor may use different people for demonstrating different functions.

Arrange for the vendor to work through the list over a day or more. They should be able to work through the functions with each person. The evaluation should take somewhere between a day for a relatively simple package, and up to a week for an ERP type system.

Each function should be rated as:

- Exceeds requirements
- Good fit with requirements
- Fits with a workaround
- Can be modified to fit (does not include those that just need to be configured)
- Will not fit requirements

Give the vendor an opportunity to review and comment on your findings. There may be some information you have not identified that the vendor can provide. For example if you say a piece of functionality is not available, the vendor might tell you that a simple plug in would solve the problem, or that the function is not required because of the processes embedded in the system.

The Time Between

All this is going to take time to arrange, and carry out. There are other activities that can be undertaken in parallel. For example, you can arrange over this period:

- Review of a standard contract by your legal people
- Vendor to provide indicative costing including all normal charges
- High level review of data compatibility. How clean, and how much of a match is your existing data?
- Review the scope and look at any changes since the project began
- Start to estimate implementation cost and effort
- Develop your negotiating strategy. Who will negotiate? What are your bargaining chips? What is must have and what is nice to have?
- Financial checks on the vendor.

On one selection I was running we did a financial check on vendors whilst arranging the Micro Functional Evaluation. To our surprise we saw a 50% drop in their parent company share price over the previous 12 months. That started to ring warning bells.

Further investigation turned up a dramatic drop in R&D expenditure and key staff departing for a competitor. Although the company assured us that they had turned the corner after some major internal restructuring, we were not convinced. We backed out of the evaluation and dropped them from our list. Two years later they did not exist.

Post Micro Evaluation Cull

By the end of the Micro Functional Evaluation, you should be down to one or two packages. Avoid getting down to one or you loose your bargaining power. Always keep another in the wings. One key consideration at this point is the implementation effort. Typically implementation is going to cost you more than the software. If it is becoming evident that one package is going to be a relatively quick and easy implementation, and another difficult, make sure the proper weighting is applied. A difference in implementation effort is not a trivial matter.

Technical Review

It is time to set loose the technologists. The next stage is to do a technical evaluation – hopefully with only one package. Whilst the Micro Functional Evaluation is under way, a technical requirement should have been prepared. There may even be some technical functional requirements that need to be evaluated. For example, the ability to create custom screens for different user groups.

My preference is to have the vendor do an install in the local environment. Obviously it depends on the type of package, but for many packages it is feasible to set it up within your environment. It gives you a feel as to how easy or difficult it is to install, operate and configure. It also gives you an indication of performance. For example, will it run successfully over a WAN if that is a requirement?

An issue may arise at this time regarding payment for the setup. It is not uncommon for a vendor to request payment for the technical support. It is one thing for the sales people to run a demo off a laptop. It is a completely different thing to have a number of technicians setting up the package on your network. In my view a payment is justified.

If you set a week aside to do a technical evaluation, you may want to negotiate a payment for a week technical support. In any case, if you do go ahead and purchase, the investment would likely be made as part of the implementation. If you don't, you have paid a few thousand dollars to avoid a big mistake.

The technical review should cover such areas as:

- Does it play comfortably with other software?
- How easy is it to build an interface?
- How customisable are screens and reports?
- What are bandwidth requirements?
- How much effort is required to install and configure the software?
- What happens under load?
- How easy is backup and restore?

Technicians tend to be more open than some sales people so it is a good chance to find out some facts that may have been skipped earlier.

Final Decision

All the factors discovered to date start to come together into the almost final decision. You still need to:

- Check reference sites
- Negotiate the price
- Iron out any contractual issues
- Review implementation effort and cost

Most of these can be organised during the technical evaluation. Once the technical evaluation is complete, the activities above can be completed. By asking for indicative pricing, and reviewing standard contracts earlier you are in a better position to move quickly.

Timing is also important. Vendors get more generous when their sales periods are coming to an end. Usually they are chasing budgets and if you know their financial year ends on 30 June, and you are nearing the final decision at that time, use the date as a negotiation point. "What will you give me if we close this deal before 30 June?"

A Few Final Points

These are in no particular order:

- Tell the vendors at the start what the process for selection will be. They will appreciate knowing what the path is, and how you will reach a decision. It assists their planning as well. You will get far better service if they know when you are going to make a decision.
- Consider an external person to do the negotiation. If negotiations are tough and the vendor feels they have been squeezed dry, it is better for the person who did the squeezing to be gone so that there are less lingering traces of animosity.
- It is a bit like herding cats but you need to keep a single point of contact with each vendor. If vendors can see a chance to influence people at different levels you will soon find all sorts of pressures being applied to you.
- Treat it as a team selection decision, and make sure your team know the process. If you set the path and the criteria from the start, it will help manage expectations internally.
- It is useful to get written agreement from each vendor to comply with the process as a condition to being considered.
- Beware the trap of having their top technical person available all through the sales cycle only to be replaced at the point of implementation by someone who has been with the company for a week. If the vendor is also involved as an implementer, have the implementation team named in the contract.
- The purchase process should be costed into the budget just as the cost of the software and implementation are costed. There can often be many man months of effort required to manage the process and, if not done effectively, will cause major cost blowouts when the wrong software is purchased.

- Another factor touched on a number of times is that much of the work is sequential. There will be gaps in the process – for example between requesting a demonstration, and organising the people to participate. During that lull in proceedings, other work can be done that will contribute to later activities. A software purchase can result in a very complicated schedule.

Summary

I once put together a package selection process that ran to over 3000 screens of information and around 350 templates. What I have covered in this white paper is the tip of the iceberg. Clients are regularly burnt because they do not have a proper process to purchase software. They end up being "sold" software rather than "buying" software.

Finally, ask yourself:

- Who knows the most about a software purchase. The vendor who does it all day every day, or you?
- Who knows more about the software? The vendor who built it, or you who are being shown selected parts of the software?
- Who has the incentive to do the right thing for your business? You, who work there, or someone you are about to buy a product from?

If you remember one thing, it should be "buyer beware".

Neville Turbit has had over 15 years experience as an IT consultant and almost an equal time working in Business. He is the principal of Project Perfect. Neville can be contacted at turbit@projectperfect.com.au

Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell "Project Administrator" software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called "Method H"™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au