mooooch Business Software Advice

Software Package Purchasing

mooooch White Paper

November 2005

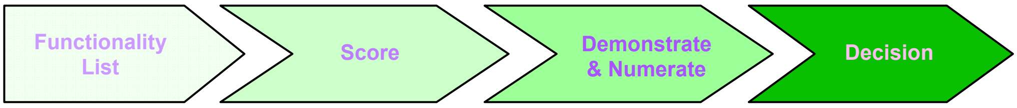
http://www.mooooch.com

Contents

Introduction

This document describes a methodology to determine which software package best fits within an organisation for a particular set of processes. It is not specific to any software genre (ie Finance system, HR system) and explains how to narrow the field from the huge range in the market place down to one.

The methodology produces a quantitative value for each package being evaluated and is based upon how it can deliver time saving benefits to an organisation's business processes.



Software Package Purchasing Methodology

One important but quite often forgotten point is that software is simply a tool to minimise the amount of time spent doing a number of processes. Any software purchased should be delivering its maximum potential to help organisations reduce business process time and ultimately save money.

To ensure a software package can provide the most efficient saving possible, it is essential to purchase one that best fits within a particular organisations structure and business processes.

Common Method of Choosing Software

The points below show how not to choose a software package, but is a common approach to software package purchasing by many organisations. It relies on subjective judgements and is a matter of luck whether the package chosen is the best one for an organisation.

- 3 to 4 software re-sellers are contacted and demonstrations are provided on the organisations premises. The demonstrations are standardised and given to all prospective customers
- Based on how the package looks and feels during the initial demonstration, the selection is reduced to 2 packages and further demonstrations are arranged, this time at the re-sellers premises.
- A more detailed demonstration is given to two or three interested parties from the organisation, where questions will be asked of the kind …
    - Question : We do X this way can the software do that ?
    - Answer : Yes it can, you may have to do Y first and then Z or we can make a modification to the software which will certainly do it.
    - Note : Any modifications will nearly always be an additional charge on top of the package price.
- After all demonstrations are completed for each package, the interested parties re-group for 30 minutes to one hour and decide which package best suits their business. The decision however, is purely subjective and based upon gut feeling and price.
- Re-seller who didn't get the order asks, why not our software?
    - Answer : Package X was cheaper than yours and it suited our needs better.
    - Real Reason : It may have been slightly cheaper but it was a gut feeling decision and the representative who sold it, did a better job than the other one.

- Note : There is rarely a mention of we needed it to do X and Y and your software took longer, or couldn't do it without spending more money.

How To Find The Right Package

Evaluating software packages should be as qualitative as possible; the object is to find a package that best matches an organisation's business processes and reduce the amount of time performing these processes.

Functionality List

First of all build a functionality list the software must do or is capable of doing. Give yourself plenty of time, at least 2-3 weeks; this cannot be done in an afternoon.

- List down the most important business processes the software is expected to interact with.

  Examples

  - Finance System : Void a posted sales invoice
  - Manufacturing System : A product can be stored in more than one bin location
  - HR System : Effectivity dates for grade, salary, appraisal etc.

- Don't include obvious functionality otherwise your list will be too long.

  Examples <u>not</u> to include

  - Finance System : Raise a sales invoice in a finance system
  - Manufacturing System : Transfer stock from one location to another in a manufacturing system.
  - HR System : Inputting a job title

- Talk to employees who will be using the new package and investigate what they are currently doing and how they are doing it. This has two benefits, the first is identifying important business processes and the second discovering unnecessary processes or where the software must reduce the completion time of the process.
- Listen to employees grumbles; what is preventing them from doing their job better.
- Add to the list any items that are peculiar to the business, but are not a process.

  Examples

  - Product codes can be up to 32 characters in length
  - Unit quantities must be to 4 decimal places
  - Customers can have more than one delivery address

Time Consuming Processes

These are processes that are very important to the organisation, where a saving of 10% to 20% will provide you with significant savings. Identify 5 – 7 processes that fall into this category.

- Record the approximate time it is currently taking to complete one task

  Examples

  - If you process many purchase orders, include the data entry of purchase orders. Time for 3 items is 2.5 minutes

- Completing finished works orders, if your manufacturing environment is works order orientated. One works order with 5 items is taking 2 minutes to complete.
- Sending letters to employees. 50 letters takes 25 minutes.

Scoring

A numerical value is placed against each of the items on the list, including the time consuming processes and based upon how important each item is to the organisation.

- Mark those items that are a must have, ie if the software package cannot demonstrate this functionality it will be rejected without further investigation.
- From the remaining items (excluding the time consuming processes) score from 1 to 5 which functionality is important to the organisation, where 5 is very important and 1 is a nice to have.

  Example : Void a posted sales invoice

  - Score 5 if large numbers of invoices are raised, where mistakes are easily made.
  - Score 2 if this would be useful but not essential, as the organisation rarely voids posted invoices.

- From the time consuming processes in the list, score what a good time reduction would be. Should the package not meet this target assign a score if it could achieve a below target value, as long as it is an improvement on the current times.

  Example : For an organisation buying many materials, based on the current time of 2.5 minutes to input a purchase order of 3 items.

  - Score of 5 if a purchase order of 3 items can be input in 1 minute
  - Score of 3 for 1.5 minutes.
  - Score of 1 for 2 minutes.

- There should now be a list of must haves and a score against each of the other items in the list. Review it with employees to ensure nothing has been missed and make it no larger than 30 items (excluding the time consuming processes).

Demonstrate & Numerate

Contact re-sellers of 3 to 5 packages and arrange for them to visit and demonstrate the software. This demonstration is designed to eliminate those packages which cannot meet your essential tasks. There is no point wasting time travelling to a re-seller’s premises if the package cannot comply with the “must have” list.

If a package ‘passes’ this first test, arrange a subsequent demonstration at the re-sellers premises, based upon the following points.

- Send the re-seller the “functional list”, however without any associated scores or target scores. Inform them of the expectation to see in the demonstration the 5 to 7 time consuming tasks and give details of what each is to do, e.g. create a purchase order of 3 items.
- If the software cannot do a task ‘out of the box’, but a bespoke modification can, make a note and obtain a ‘ball park’ value for the cost to code the modification.

  Any bespoke work costs should be added to the price of the package, however keep the amount of bespoke work to an absolute minimum (see Software Package Implementation white paper), even better none at all if possible.

- Throughout the demonstration tick off a task from the functionality list that has been demonstrated successfully and add its score to an overall total.

Decision

Once all demonstrations have been given and there is an overall score for each package, choose the top two if they are relatively close together or it may be one is a clear winner.

Taking into account the organisation’s budget for the software and the price of the packages with the highest scores, an objective decision on which best matches your business can now be made.

For example, trading off some functionality for a package with a lower price.

Additional Modules

Re-sellers will quite often try to sell additional modules at the same time as the main core package. Here are some points to consider.

- Do not purchase modules that are not going to be used immediately; if they are needed they can be bought later. Everyone involved with the new package will be too busy familiarising with the basic functionality and not have spare time for learning any specialist modules.
- Review the situation again after 6 months to see if the specialist modules would be beneficial to the business, but only after all the systems have bedded down and everyone is comfortable working with the package.

Changes To The Software

Software packages are written to encompass a broad spectrum of market sectors, as such screens are over complicated, there are a huge number of set up parameters and reports contain a morass of data.

The temptation is to modify the package to exactly fit the business process. Try to resist this if possible.

- Initially the package will be unfamiliar and there will be many different ways of doing things. Try and get to know the package first before commissioning any modifications.
- Modifications can be an upgrade nightmare, quite often on a new release they will need to be re-applied. Although the software developers will do this, it can be more trouble than its worth. For a heavily modified system it can sometimes be too costly to do any upgrades at all.
- If your package cannot handle a specific process how the organisation would like, then by all means investigate if a modification will help, but make sure there will be real business benefits from it and it is not a “nice to have”.

Parent Companies

Many parent organisations have global licenses that encompass the use of the software by subsidiary companies. This gives the impression it is free software; however before embarking down this route, consider some of the following points.

- The software is very likely to be an enterprise solution. These are extremely complex systems and normally require a dedicated server and on hand IT expertise. Investigate what resources are needed to run the system (e.g. server, database, support).

- The functionality of the system may be much more than the organisation requires, it may make the processes less efficient rather than more efficient.

  Example : Additional data required to be completed in a form, most of which is irrelevant to the subsidiary company.

- Enterprise solutions take a lot of setting up and training, find out how much consultancy costs are and the number of days set up and training required. (By saving the consultancy charges alone, it may be cheaper to purchase a separate package that provides better functionality and process time reductions for the subsidiary company).

- Check on licensing implications if the parent organisation sells the subsidiary company. Would there be a need to purchase licenses if this occurs and how much.

  Note, this has the potential to expose the company to a significant unbudgeted expenditure.

Conclusion

By comparing software packages using this methodology an organisation will not only have re-acquainted itself with some of its time consuming processes, but will have identified a package that will deliver tangible time saving benefits and make a contribution to taking the business forward in the oncoming years.

The information contained in this document represents the current view of mooooch Ltd at the date of publication.

This white paper is for informational purposes only, mooooch makes no warranties express or implied in this document.