

A Guide to Evaluation, Purchase and Implementation of Software Applications for Non-Profit Housing Societies



Published May 2015



FOREWORD

BC Non-Profit Housing Association (BCNPHA) has partnered with BC Housing to build the capacity of the non-profit housing sector around the procurement of appropriate technology to enhance the efficiency and sustainability of affordable housing. This guide comprises one of a range of supports being provided through this partnership.

BCNPHA is a provincial umbrella organization that provides leadership and support to the non-profit housing sector in British Columbia. The association's membership is comprised of non-profit housing societies and organizations concerned with affordable housing. The association provides extensive education and other resources for leadership, operations and asset management.

BC Housing is a Provincial Crown Agency that develops, manages and administers a wide range of subsidized housing options across the province, totaling 7,200 public housing units and more than 300 group homes. It operates in partnership with public and private housing providers, local government, and community agencies to create the best system of housing support for those in greatest need.

With thanks to BC Housing (particularly Mike Klein), the Greater Victoria Housing Society and the New Chelsea Housing Society who kindly gave permission to reproduce documents and processes from their joint software procurement project, and Lily Nimi, independent consultant.

This guide is intended to be an iterative document informed by feedback from non-profit organizations. Any comments and feedback are most welcome, and can be sent to:

BCNPHA

Jacqui Mendes, Director of Member Services and Administration

Phone: (604) 291-2600 Ext. 225

Email: jacqui@bcnpha.ca

BC Housing

Mike Klein, Chief Information Officer

Phone: 604-439-4777

Email: mklein@bchousing.org

Table of Contents

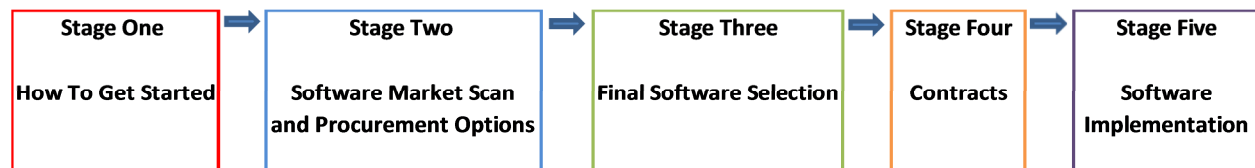
Introduction	4
Stage One - How to Get Started.....	5
Business Needs Analysis	5
Defining the Needs	6
Defining the Future State Needs.....	6
Financial Analysis	7
Capacity and Resources	7
Stage Two - Software Market Scan and Procurement Options	8
Software Market Scan	8
Procurement Options	9
Develop a Request for Proposal (RFP)	10
Stage Three - Final Software Selection	14
Evaluation of Software RFP Responses.....	14
Software Demonstrations.....	16
Final Selection.....	17
STAGE FOUR - Contracts	18
Negotiation.....	18
Detailed Scope of Work (SOW) for Implementation Services	19
Software License and Maintenance Agreement.....	19
STAGE FIVE - Software Implementation	21
Project Management.....	21
Change Management	22
Conclusion	23
<i>Appendices (see separate document)</i>	23

A	<i>Sample Quick Wins</i>
B	<i>Sample Requirements Matrix</i>
C	<i>Sample Consultant Contract</i>
D	<i>Sample Market Scan</i>
E	<i>Sample RFP</i>
F	<i>Sample Demonstration Script</i>

Introduction

BCNPHA and BC Housing held a series of meetings and conversations with non-profit housing providers around British Columbia to talk about information technology –what were the needs and aspirations of providers, and what barriers existed around efficient technology use. The capacity to undertake buying software solutions emerged as a key barrier. This guide is one of the responses generated to remove that barrier.

This guide is structured in five stages:



Stage One – How to Get Started

The first step in any procurement process is an analysis of your business needs. Identify current and future processes and needs, give you a sense of what you can spend, and be a guiding tool throughout your procurement process. Consider external resources for producing a business needs analysis.

Stage Two – Software Market Scan and Procurement Options

The number of available off-the-shelf software options could be as few as three or as many as thirty. An initial scan of what is out there is helpful in getting a better sense of how many options are an approximate fit, will help refine your budget and decide on the appropriate procurement process.

Stage Three – Final Software Selection

How to evaluate vendor responses, approach software demonstrations and acquire client references to make a final software selection.

Stage Four – Contracts

Understand how to structure and negotiate contracts for buying and implementing the software, and the ongoing licensing and support agreements.

Stage Five – Software Implementation

Consider when to undertake implementation, what this may mean for the organization and staff, and what change management looks like for your colleagues.

Stage One - How to Get Started

Business Needs Analysis

The most important way to get started on software buying is to figure out what works for your unique needs. The recommended first step would be to construct a business needs analysis. This should include i) current needs, ii) a look at what your needs may be in the future, iii) a financial analysis and initial budget, and iv) capacity and resources to purchase and implement the software.

A business analysis is important as organizations are different. What works for the society next door may not work for all societies, even though the work you do may be similar. Internal processes, staff capacity, tenant types, unit turnovers, building types – all these and more make a difference to your needs. Don't buy something just because it works for someone else.



Non-profits are also process-driven. They move tenants in and out, they maintain units and buildings, they identify, serve and record transactions with clients, and they recruit, hire and manage employees. A business analysis will identify your processes, and allow you to consider what really needs to be automated, what would be nice to have automated but isn't critical, and what process changes may be needed to automate. Areas to consider in the business analysis include:

- Identifying exactly what the society's requirements and key business drivers are now
- Mapping existing processes
- Reviewing and mapping system output required – reports, management information, property manager needs, tenant needs, funding reports legal requirements etc.
- Forecasting needs for the lifetime of the potential software purchase (5-10 years)
- Establish a project team of key staff from the get-go, to share in decision-making and selection - key to staff buy-in and effective system use
- Ensure that the staff project team allocates necessary and appropriate staff time for optimum effectiveness. Investing in improving capacity at the beginning will pay off in the long-term
- Continue to talk to all staff stakeholders across the organization from the beginning and frequently throughout the process. They have valuable hands-on operational information, and they will be primary system users
- Keep your analysis simple and clear – this will be the basis for a potential Expression of Interest (EOI), Request for Proposal (RFP) or Request for Quotation (RFQ) from potential suppliers
- Think about 'quick wins', i.e. what your needs analysis has shown can be fixed simply and now. Not everything may need an IT solution.

See Appendix A – Sample Quick Wins

Defining the Needs

What are the compelling reasons for buying new software? Think about and write down the answers to a series of questions with key staff across your business areas. The answers can be made part of your business case and be incorporated into an eventual EOI, RFQ, RFP or whatever procurement process you embark on eventually. Some typical questions are:

- What are we trying to achieve?
- What are our expectations?
- Will the software make your operations more efficient?
- Is our aim to improve quality of life for your tenants?
- Will the software be used by many or a few staff?
- Will different departments or locations be using it?
- Will this software integrate with other data systems?
- If this software doesn't work, will it have significant impact on our nonprofit's ability to operate?
- What are the existing process work flows?
- Can processes be improved before your software implementation?
- What are the reports used now across your organization?
- Do you need better or different reports? Who for (think about functions rather than job titles)?
- Do you need to input more data for reports? If so, where will the data come from? Do you have the capacity to maintain the data?
- Do you plan to convert existing data? Does this include all data, or only some (e.g. tenants, buildings/units, assets)?
- If you already have financial software, will the new system replace it, or be able to speak to it?

Defining the Future State Needs

- What is the life expectancy of the software, and how might your organization change during that time?
- What functionalities may you need for future expansion (or contraction), new areas of operation or tenants, or you may want to add as you can afford them?

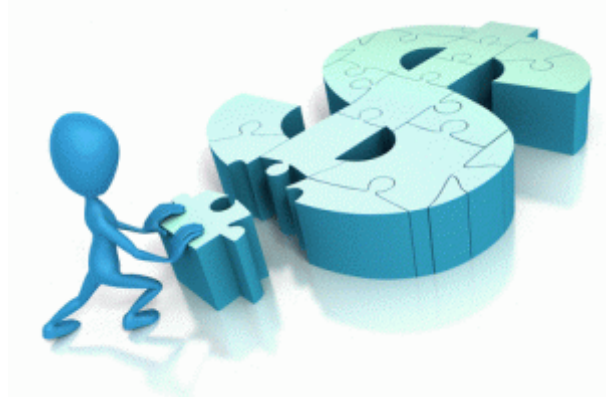
The Defining the Needs and Defining the Future State Needs will comprise the first draft of a needs requirements matrix.

See Appendix B – Sample Requirements Matrix

Financial Analysis

Understand the budget for purchase and implementation, as well as ongoing support.

- Calculate total funds available for purchase and implementation (implementation cost may include new hardware, mobile devices, IT infrastructure, internet upgrades for web based software)
- The general guideline is that implementation may be 80% of the total cost, the actual software may only be 20%
- Think about where you will get the money you will need - are there funders, local community opportunities for support, or maybe other societies who want to purchase software, and may share the cost of a consultant and implementation?
- The cost of the software and implementation is important, but it is equally crucial to know what your recurring costs might be, e.g. annual costs per user of a cloud solution, support costs, training, software customization, upgrades
- Implementation consulting/ and customization is about \$90-\$200 per hour depending on the consultant's skills
- Complex requirements and customizations add to the cost of implementation
- Include potential annual licensing fees in your recurring costs
- Include extra room in budget for additional work which may surface during implementation, particularly on change management side



Capacity and Resources

The capacity and resources needed for software procurement depends on the complexity of the system you need. A key success factor for mission critical software is ensuring that internal staff, including both senior management and front line, allocates sufficient time for the procurement and implementation process. Your business analysis should consider staff resources required as well as any external support that may be necessary.

Serious consideration should be given to retaining a consultant for the needs analysis. Upfront costs of a consultant can save a great deal of time and money in the long term, and ensure a more effective solution that fits needs. Organizations should consider joining forces with others undergoing a similar process to share a consultant, which gives great value-add around sharing of best practices as well as lower costs. The consultant's scope of work could include:

- Construction of detailed business needs analysis (needs versus wants)
- Construction of EOI, RFQ or RFP as necessary based on analysis
- Evaluation of EOI, RFQ or RFP software fit to business needs
- Evaluation of infrastructure and hardware upgrades if necessary, and costs
- Assistance with final software/vendor selection and contract

See Appendix C – Sample Consultant Contract

Stage Two - Software Market Scan and Procurement Options

Software Market Scan

Once you are equipped with your business needs analysis and an idea of what you want to spend, you (or your consultant) should be able to embark on a preliminary market scan of off-the-shelf solutions as well as vendors that can provide software design and building services. This will give you a sense of what is out there that fits your budget, or help you to redefine your budget. The scan will be a guide to deciding how to start your procurement process, depending on the number and cost of potential software solutions.

This guide will focus on off-the-shelf solutions, which is the recommended route. Whilst building software programs from scratch is certainly an option, it is suitable only for organizations with excellent long-term technical and staff capacity and significant monetary resources. Software solutions using standard tools such as Access and Excel have been successfully undertaken by small and large organizations; however functionality and integration with other software tends to be limited. Upgrades and product development for new operating systems or new processes with built- for-purpose software can be time-consuming, costly and technically challenging. Off-the-shelf solutions have a large customer base that ensures investment in research and development necessary to keep products up to scratch, fix bugs quickly and provide good support. Off-the-shelf solutions can often be customized to fit your specific needs; however caution is urged when considering extensive customizations as these can be expensive and can create challenges with software upgrades.

You should be able to evaluate and eliminate many products based on lack of features, or high cost, or which may need extensive customization. There are many sites listing available software packages and allowing you to search by features or platform. Many of these sites also include pricing information, making it much easier to obtain a clear sense of potential vendors. It is very useful also to talk to similar organizations, and associations you may belong to.

See Appendix D for a Sample Market Scan

Procurement Options

The market scan should give a clear sense of how many solutions are available that could fit your needs and are in your budget range. If many solutions are available, you may need a formal detailed process to figure out differences. If only two or three fit needs and budget, you may prefer an informal process that takes less internal capacity. However, the size and complexity of the software solution is a factor – large, expensive software will be a high-impact investment and requires a rigorous selection process. Smaller, less critical solutions with may not require such a formal process.

There are a number of ways to buy software solutions depending on your risk appetite, and how much time, money and resources you want to apply to the process:

	<u>Process</u>	<u>Description</u>	<u>Risks</u>	
Low				High
RESOURCES, TIME & COST ↑ ↓ High	1	No formal or informal requirements documented, invite vendor(s) to provide product demo, no formal evaluation, purchase product.	Requirements have not been thoroughly thought out, vendor will make their product look better than it actually is, very little understanding of downstream costs/resourcing past the product purchase.	RISKS ↑ ↓ Low
	2	Informal requirements discussed internally, invite vendor(s) to provide product demo, no formal evaluation, purchase product.	Requirements a little more thought out but risks still exist as outlined in Process 1	
	3	Formally gather requirements, invite vendor(s) to provide product demonstration, formally evaluate based on requirements, purchase product.	Requirements well thought out and software products thoroughly evaluated	
	4	Formally gather requirements, issue RFP, evaluate proposals, invite vendor(s) to provide product demonstration, finalize evaluation, purchase product.	Least risk, thoroughly evaluated, proposal bound to contract for vendor accountability downstream, better sense upfront about what you are getting yourself into.	
High				Low

All of the above processes can be facilitated in-house or through a consultant. Processes for larger complex solutions, or where there is little in-house capacity, can be outsourced to a consultant where budget allows. All processes will require internal staff capacity

Process 1

Not recommended. Highest risk category with no documented requirements or evaluation, and no understanding of subsequent costs and maintenance.

Process 2

Suitable only for low impact, low cost solutions where needs are very standard. Risks are still high as outlined in process 1.

Process 3

Suitable for an organization with a clear understanding of their needs and budget, and where there are not many software options available. Not recommended for high cost, complex software, software that is mission-critical, or where there are many viable options.

Process 4

The software procurement process with the least risk. This may take more time, but it will pay back in terms of financial and staff capacity investment, the best fit to your needs and budget, and relationship with and accountability of the vendor.

Develop a Request for Proposal (RFP)

If you have decided to go through a formal RFP as in Process 4, this must be structured to enable an efficient evaluation process:

- Figure out what is essential, as opposed what is interesting or attractive, but not necessary. List all your requirements, subdivided by categories such as 'Must Have', 'Nice to Have' and ensure responses to these by the vendor are mandatory
- You need information that can be tracked with simple checkboxes. For example, "It can/can't do this," "it can track unit move-in and move-outs" or "it can track maintenance requests by unit, tenant and building"
- An RFP that asks questions that encourage vendors to answer clearly, with answers that can be compared in a simple matrix, will be useful for assessing and documenting the system capabilities
- An RFP can't address all the concerns you're likely to have - subjective questions like "How user-friendly is your system?" or "Please describe your support" are unlikely to be answered meaningfully through an RFP process
- Establish a core staff team to create the RFP, and form the evaluation team. Add external resources that will be helpful, such as your consultant or other third parties with the necessary expertise



Structuring an RFP

The following example of an efficient RFP design comprises four sections:

RFP Section 1 - Introduction

The introduction provides a summary of your organization, mission and the purpose of the RFP. Include your organizations' mandate. It is important for vendors to really understand who you are and what you do.

RFP Section 2 - Background

The background section provides context the vendor will need to truly understand your situation. Include a description of your organization—for instance, number of locations, number of staff and organizational structure, the processes the system should support, and technology infrastructure as network operating system(s), hardware and other core software packages – or where no software exists. Include any upcoming projects that might be relevant.

RFP Section 3 - Questionnaire

The questionnaire is the critical piece - an opportunity for them to understand the project up front and determine their suitability for it. Make sure to ask all of the questions that you need answered. A good vendor will appreciate and want to work with a client that has thought things out well. It's important to be thorough, but don't ask a lot of questions you don't plan to use to actually evaluate the systems. Asking questions "just in case" increases the amount of information you'll need to sift through later, and increases the possibility that vendors might decide your RFP isn't worth the time to respond to.

Consider asking about:

- Functionality - detailed specific questions; use your business requirements to focus on the functions that are critical to you and your specific needs
- Software specifics -
 - Will the software will integrate properly with other applications you use.
 - Can reports can be customized
 - Can ordinary users can customize reports, or does this have to be done by a specialist
 - Support – when available, who by, cost
 - Which formats can data be exported to and imported from
 - Ask for a product roadmap; if the next version is going to be a complete rewrite of the application, you might want to rule out the current version for consideration
- Vendor information -
 - How big are they?
 - How many customers do they have?
 - How many customers do they have similar to you?
 - Do they have an office near you, or in the province?
 - How many people are on their support team?
 - How will they train your staff? Online? In person? Train the trainer?
 - How long have they been in business?
 - Are they public or private?
 - Can they provide some documentation of financial viability?

- Who are the staff members that will be assigned to your project? What is their experience in this type of implementation, how long have they been with the vendor?
- Ask for references from similar clients with similar types of software needs
- Pricing and availability:
 - Cost of software?
 - Cost of implementation?
 - Total estimate for the project as described?
 - What are hourly rates for implementation, broken down by role, if applicable?
 - Payment terms?
 - What hosting choices are available? a) putting the software on your own server, or b) using Software as a Service model (SAAS) in the cloud (online access with an external software host)?
 - What are the software and implementation costs over 5 years of hosting choices?
 - How do they handle changes in project scope that might arise during implementation?
 - What are their incidental rates and policies (travel, meals)?
 - Do they discount their services or software costs for non-profits?
 - How long do they estimate this project will take?
 - When are they available to start?
 - What are their standard support hours and costs after implementation?
 - What are their costs for adding more users?
- Privacy and Security -
 - If they are offering Software As A Service (SAAS), is the data held in Canada?
 - Who ultimately owns the data?
 - Who has access to data?
 - What happens to the data if they go out of business?
 - Is there planned redundancy for down time or software/hardware errors which could damage your database?
- Software Licensing:
 - How is the license pricing structured? Is it by user, by module, or by number of units?
 - What is the annual pricing?
 - How will annual license price increases be negotiated?
 - How many concurrent users will be able to access the software?
 - What does it cost to add user licenses? Is there a minimum amount of user licenses per purchase?
 - Are there different types of user licenses with different costs?
 - What does it cost to add additional modules?
 - What is the cost if unit numbers increase?
 - Will there be an additional license fee for software upgrades?

RFP Section 4 - Instructions

It is important that vendors know exactly how you want to receive submissions, and when by:

- Close with a deadline and details about how to submit replies
- For a large RFP, allow a minimum of four to six weeks for a response
- Giving ample time to provide a detailed response will ensure that you receive a good clear submission
- Indicate how additional questions will be handled
- Usually, if one vendor asks for clarification or details, your answers should be shared with all of the RFP participants. You want to keep things on a level playing field, and not give one vendor an advantage over the rest
- You might do this via a group Q&A, with all the vendors invited to participate in a meeting or conference call after the RFP has been sent to them but well before they are due to respond. With all vendors asking their questions in the same room, you keep them all equally informed
- Alternatively, you can specify a deadline by which written questions must be submitted - all participants would then receive the questions and answers, which can be posted as an addendum to your RFP

See Appendix E - Sample RFP

Stage Three - Final Software Selection

Evaluation of Software RFP Responses

Procurement process options 3 and 4 require formal vendor evaluations. Have each member of the RFP implementation team evaluate the responses independently. Then work through the evaluations together to arrive at a consensus.

As long as you understand your own critical needs, the RFP responses will help pare down the list by eliminating candidates:

- Set up a comparative matrix. Create a table or spreadsheet with columns for each vendor and rows for each question, summarizing the responses as much as possible in order to have a readable chart. You might add columns that weigh the responses, both on the suitability of the vendor's response (e.g. 1, unacceptable; 2, fair; 3, excellent) and/or on the importance of the question (for instance, some features are going to be much more important to you than others)
- Which solutions offer the most features on your Must Have Wants list? Which packages offer the most features from your Nice to Have Wants list? Even if they are not within your budget now, you may be able to purchase add-on modules down the road. You'll see the strong and weak points of the applications by comparing and contrasting the features and technology sections for each solution
- User licenses – these can differ quite significantly between software. Think carefully about the model that suits the organization
- Pay careful attention to pricing and make sure you are comparing apples to apples. As stated, as a general rule of thumb, software cost is usually 20% of total project cost, implementation usually 80% - this also depends on the size and complexity of the software. Smaller straightforward off-the-shelf non-critical solutions that don't need customization may have much lower implementation costs.
- What product enhancements are planned? Are they features that will be useful to you and how quickly will they become available? Often you can negotiate a discount on additional modules prior to their official release
- In determining which solution fits your needs, there will likely be some trade-offs—perhaps one application has a stronger model for handling processes, but another has more flexible reporting. It's unlikely that any will jump out as the perfect application, but you'll be able to determine which are generally suitable, and which aren't



- The 'Best of breed' approach often works well – which software has the best reputation for the core functionality you need most?
- Be wary of a vendor response that outlines a custom approach, or extensive customizations that may not work with software upgrades
- Consider software upgrade costs over time, and complexity of supporting these
- Consider user security and data issues around privacy.
- Vendor evaluations:
 - The RFP responses will tell you a lot about the vendors. You're asking questions that are important to your ability to operate. Their ability to read, comprehend and reasonably reply to those questions will offer a strong indication as to how important your organization is to them, and whether they'll consider your needs as the software is implemented and into the future. If they respond (as many will) to your critical questions with incomplete answers, or with stacks of pre-printed literature—saying, in effect, "the answers are in here"—then they're telling you they won't take a lot of time to address your concerns. Look for a collaborative vendor who will work well with your organization. Choose not just a product, but a partner who is willing to help before, during, and after the actual sale
 - You might rule out a vendor or two based on what the RFP response tells you about their availability or company stability
 - Consider vendors long-term support capabilities
 - Take care, though, in eliminating vendors based on their RFP pricing information. RFP responses can be very subjective. Before determining that a vendor is too pricy based on their project estimate, dig deeper—other vendors might be underestimating the actual cost. If you feel you have a solid grasp on the project timeline, use the hourly rates as a more significant measurement
 - Remember that a weak sales representative might not mean a weak vendor, particularly if they're representing a product that comes recommended or looks particularly suitable on all other fronts. It's acceptable to reject the response and ask the vendor to resubmit if you really feel they have done you, and themselves, a disservice—but temper this with the knowledge that they blew it the first time
 - Focus on companies that are clearly designing products with one eye on the future. Does their product easily integrate with others? Is the software platforms supported by industry leaders? Choose a platform that has a broad base of users so you won't have trouble finding resources to keep it running down the road
 - Avoid companies that don't have at least 5 years of experience working with customers. If they can't point to a history of working with the type of software you're shopping for, they probably don't have the experience to help you

Software Demonstrations

The next step is to schedule software demos by the software vendors.

- Ask vendors to use a demo script provided by you (your consultant or a business analyst can assist with building this). This will be a great help in comparing software. Otherwise you won't know exactly what each vendor can or can't do, compared to their competitors. A demo script prevents a vendor from being able to highlight product strengths while glossing over weaknesses.
- Sample data - it can be useful to provide them with sample data, if they are willing. Evaluating a program with data similar to your own data will be less distracting. Be careful not to provide them with actual data that might compromise your clients' or tenants' privacy and security. The goal is to provide a level and familiar experience that unifies the demos and puts you in the driver's seat, not the vendor
- Evaluate the look and feel of the software – as good as the specs might look, you'll know quickly in a demo if an application is really unusable
- Dig deeper into your specific needs - the demo is another opportunity for the vendor to show the operating assumptions of the software, and for you to provide them with more insight into your needs. In less standardized areas the model used by the software application can differ dramatically from your internal processes, making it difficult for your organization to use. Use the demo to learn how the software will address your own process and specific needs
- Internal training and staff buy-in - use demos to show internal staff what they'll be able to do with the software. Demos are an excellent opportunity to get staff thinking about the application of technology that you should pack the room with as many people as you can. Get a good mix of key decision-makers and application end-users—the people who design and perform the processes the software facilitates. The people who will actually use the software are the ones who can really tell if the package will work for them



Refer to Appendix F for a Sample Demonstration Script

Final Selection

- Vendor references:
Before making a final decision, always check vendor references, preferably from organizations in your business area. However don't completely rely on these. Satisfaction with software depends not only on how it meets their needs, but how familiar they are with their options. There are a lot of people who are happy using difficult, labor-heavy, limited applications simply because they don't know there are better alternatives
- If you still have a tie after RFPs, demos and reference checks, the best next step is to conduct on-site visits to existing customers for each software package. As with demos, bring a representative group of staff. The visit will highlight how the software meets their needs, and will give you a good, actual look at its strengths and weaknesses. You'll also obtain new ideas how you might use the software
- Consider costs and feasibility of hosting software in-house versus Software as a Service (SaaS) over 5 years. Include costs for comparison for support, cloud hosting, hardware/software costs and internal technical resources
- If none of the applications you evaluated completely meets your needs, but one comes close, consider customizations or software modifications to address the missing areas. However - any alterations of the basic software package will likely be costly, will not be covered in the packaged documentation and help files, and might break if and when you upgrade the software. Be very sure there isn't an alternative, built-in way to accomplish your goal. If the modification is justified, make sure it's done in such a way that it won't be too difficult to support as the software is developed.

Your vendor selection process will now be complete, with one package clearly identified as the best option. If key users are torn between two options or unimpressed with the lot, senior decision-makers might have to make the call. Be careful, however, not to alienate a group of people whose commitment and enthusiasm for the project might be needed.

STAGE FOUR - Contracts

Negotiation

Start reviewing the contract when you are close to the final decision and the vendor is doing everything they can to win your business. Don't wait until the very end, because by then you will have given up most of your negotiating power. Draft an agreement that will cover you in case of problems - don't just accept the vendor's contract. The goal is a contract that equally defends the rights of both parties. A good contract clearly articulates and sets out everything that has been discussed to date into a legally binding agreement. If, down the road, the vendor isn't living up to their promises, or the software can't do what you were told it would do, then this is your recourse for getting out of an expensive project.

- Understand contract types and ensure this is clearly defined:
 - Fixed price contract - a price is agreed on that cannot be changed except by mutual agreement. The price does not change according to the amount of resources or time expended by the vendor. Most of the risk is on the vendor side
 - Time and materials - the buyer agrees to pay the vendor for all work performed, and for materials, no matter how much work is required to complete the project. If unexpected issues arise, or the project takes longer, the buyer pays more. Most of the risk is on the buyer side. Consider risk assumptions for the implementation contract on both sides. Often vendor contracts are one-sided and the client takes all the risk.
- Negotiate fees and terms as well as prices: for example, lock-in vendor and/or implementation consultant hourly rates for the duration of the implementation and post-implementation trouble-shooting phase. Vendors are often flexible with purchase prices but less so with maintenance fees. Ideally, those fees should be based on a discounted price rather than the list price and they should be capped to prevent unlimited escalation.
- Don't rush into signing: contract negotiations can take time. It's more dangerous to sign a bad contract in the interest of time than it is to delay a project while you ensure that both parties completely understand each other's requirements.
- Get additional advice: talk to your consultant or a third party that has experience in this area. For purchases that are large for your organization, it is well worth having a lawyer or your consultant review or assist in negotiations.

Detailed Scope of Work (SOW) for Implementation Services

A SOW is usually negotiated with the vendor prior to the actual contract and should be attached as a governing exhibit to the contract. By having it attached to the contract, the vendor is now legally obligated to do what they said they would do. It describes exactly what the project will consist of, what will happen, when, and how long it will take. Good scopes include:

- Project charter (vision, stakeholders, definition of successful outcomes)
- Estimates of hours and costs by task and/or stage of the project
- Detailed project deliverables
- Payment milestones linked to project deliverables, with a clear process for agreement by both sides that the deliverable has been met
- Full implementation details in a project plan with timelines and responsibilities
- Clear timelines for each project item. Include flexibility for unanticipated delays
- Roles and responsibilities of buyer and vendor
- Names and bios of vendor technical staff
- Training details and costs
- Hourly rates for technical support for implementation (for duration of project)
- Process for mutually agreed change orders
- Dispute resolution mechanism, clearly stating that provincial law applies

Software License and Maintenance Agreement

It is critical to ensure that all software, modules, license structure and number of users are explicitly listed in the agreement. This should reflect exactly what has been discussed, suit your structure and users, and allow for affordable expansion. The license specifies the allowed uses of the software you're purchasing and can contain some unacceptable conditions. Consider:

- Support details and costs for face-to-face support, remote implementation or both
- Hourly rates for on-going technical support after implementation
- Details on what happens if the vendor goes out of business, e.g. access to software codes/data
- Full software details including version number and details of all modules to be installed/maintained
- Annual licensing details and costs
- Number of concurrent users and cost per user or user block
- Annual maintenance and support details and cost
- Software upgrade costs
- Details on use and access to your data
 - software licenses should not restrict your rights to access or work with your data in any way



- The license agreement should not have conditions under which the software warranty would be voided. It's ok for a vendor to bar re-engineering their product, but not ok for them to void the warranty if you are only modifying data
- Conditions that prevent exporting, importing, archiving or mass updating of data should be removed
- If the system is hosted, contract language must ensure that you own all of your data. There should be no language in the contract implying that the vendor owns your data, or that they can use it for any additional purpose
- The vendor should provide full access to your data
- The license should include language ensuring that buyers have reasonable access for using, copying and backing up all customer information in the database.
- If the system is hosted, the data must be on a server in Canada

The agreement should not contain:

- Responsibility avoidance - software warranties should not include blanket "software provider is not responsible if nothing works" statements
- Licensing changes - the license should not allow for any post-sale reversals of licensing, such as language stating that the contract will be void if the customer uses the software in perfectly reasonable ways they don't anticipate
- Back doors - there should not be any "back doors" programmed into the application that the vendor can maintain for purposes of disabling the software

STAGE FIVE - Software Implementation

Once software is selected, you can move forward to software installation. This should be treated as a management project, with defined start and end dates, and assigned staff and budget resources. Your contract with the vendor should have these already outlined in the contract scope of work. Success factors include a good relationship and communication flow with the vendor, allocating staff time (usually more than you may think), involving key staff and top management as appropriate, and being very clear on your goals and processes.

Software implementation where there is existing software and data conversion is needed tends to be more complex processes. If the new software is replacing the old software, think about running the two systems parallel for a defined period, if resources allow. Work closely with your vendor for a seamless transition. Data conversion becomes very important to plan carefully, as the way your data was structured in the old system may not work in the new system, and may need work and time to be imported correctly. Complexity also exists where the new software has to talk to other software.

If you have no existing software data conversion is not an issue. However you may need to adjust and improve processes to ensure the right data is captured and input correctly and in a timely manner. It is an excellent opportunity to have a fresh look at work flows and staff responsibilities.

Project Management

- Establish timelines with soft and hard deadlines
- Plan your implementation carefully to avoid your peak busy times and to allow for system redundancy to ensure that none of your data is lost during the transition period. Many organizations feel compelled to change systems at the beginning of the fiscal year, but often this means implementing your new system at the end of a fiscal year when most organizations are focused on end of year audits and reporting - most systems will allow you to begin implementation at any time without any loss of data or additional work
- Understand the impact on your organization and the project if timelines shift, so the project plan can be readjusted
- Determine a project lead
- Determine clear and detailed staff and vendor roles and tasks – this will ensure that the project isn't held up by misunderstandings over who is doing what work
- Have a written project plan that all stakeholders can access
- Have a written internal and external communications plan
- Schedule project progress meetings regularly, and around key deliverables and milestones
- Determine a process for troubleshooting, overcoming implementation obstacles and possible disagreements with the vendor
- Keep calm and carry on!

Change Management

Change management is one of biggest challenges when investing in new software. Employee resistance, communication breakdown, insufficient training, staff turnover and budget overruns can all be obstacles during major change. Key strategies to address these include:

- Involve staff in the process from the beginning
- Ensure continued support and buy-in from organization top management – staff will quickly pick up ambivalence or disinterest from senior management, and can become discouraged at a time when their commitment is needed
- Have a clear articulated vision of why your organization is doing this
- Communicate what the benefit is for staff, particularly individuals resistant to change - evaluate what is in it for individuals, articulate and promote these to boost staff buy-in
- Select key informants from staff who can keep others informed and motivated
- Develop in-house communication tools, such as newsletters, email updates, regular meetings
- Gather internal long-term subject matter experts
- Recruit internal champions
- Consider discrete training strategies for different groups, e.g. finance, property managers will have different training needs
- Increase internal IT competencies and support capacity where necessary
- Consider capacity and upgrades for new or existing infrastructure if necessary, including other software, hardware and internet
- Invest in staff success by ensuring you have an adequate training budget. This should be both for the initial implementation, and as an ongoing resource for new staff, software upgrades, and refresher courses. The 'Train the Trainer' model can be cost effective, whereby selected staff become in-house experts and take on training other staff.
- Celebrate when the project is finished



Conclusion

Software procurement is well worth the investment in cost, time and energy, when done correctly. It can transform your organization by giving staff the right tools to do their jobs better, manage and share information. It hardly ever means less work, but can greatly enhance the efficiency of your organization.

Appendices (see separate document)

- A Sample Quick Wins*
- B Sample Requirements Matrix*
- C Sample Consultant Contract*
- D Sample Market Scan*
- E Sample RFP*
- F Sample Demonstration Script*