



New Rules: Running Your Business From the Cloud

How to get world-class business management tools
without paying (much)

By Richard Minney
(iBE.net founder)

Introduction

You run or help run, or you have aspirations to run, your own small or mid-sized business. You work hard, you are creative, you are passionate about what you do, and yet you feel like you are playing against the world's largest companies on an unbalanced field. Why is this?

Many barriers for smaller companies competing with larger businesses, such as bulk buying power, brand awareness and advertising budgets, have been eroded by this thing called the Internet. Fortune 500 companies may have inefficiencies with overpaid and under-motivated staff buried under layers of management. And yet you find it impossible to compete with larger companies because you don't have the tools to manage your business effectively, and you find it hard to scale. You don't have software designed to run your business the way you work, specialized for your industry, at a price you can afford.

Do you need specialized business management software? If you are a sole proprietor then you can do just fine using a spreadsheet and perhaps a home business finance package from your local stationary store or purchased on-line. Even if you are a handful of employees or contractors, ask yourself how you personally make all the decisions about your business. Do you have all the information you need to make these decisions, in your head, all the time? If the answer is "yes" then you still don't need specialized software to run your business. Put down this book, forget about spending your hard earned cash on some software package you don't need, and spend the time saved with your friends or family.

But if you often rely on looking up information, or talking to employees, business partners, suppliers and customers to make decisions, then you need to collaborate with them in real-time and on-line. In that case you need effective tools to manage your business and to help you compete with the big-boys. In that case this eBook is for you.

You may have read blogs or articles about the recent trends in cloud, social networking and mobile apps for business. No doubt you have a smartphone, which you are using to manage your contacts, appointments and perhaps a personal to-do list. Did you know that these trends along with some other technological advances have revolutionized the business management software industry? Literally turned it on its head in the last five years. This eBook explains what cloud, social and mobile, as well as the other four trends outlined below mean for you and your business, and how to benefit from them.

Technology trends in the last five years

1. Software as a service, or cloud computing to use its trendier modern name, allows you to connect your business to remote, best in class software for a fraction of the cost of downloading, installing and setting up your own systems. This is because the software supplier can share IT resources and costs between thousands of companies like yours sharing the same applications.
2. Social networking has profoundly changed the way people communicate outside work, and now it is having the same impact within the workplace. Employees, customers, suppliers and partners find they can work together without receiving 400 emails a day or attending back to back conference calls from dawn to dusk.

3. Today's smartphones are empowering a new generation of business owners and workers, making business tools not only more accessible from more places, but way easier to use and manage at the same time.
4. Yesterday's business applications with their dull and complicated user interfaces requiring days of specialized training just to navigate are getting consigned to the scrap-heap. A new breed of consumer and mobile apps are being fast adopted by a new breed of business software developers, setting new standards for user interface design, intuitiveness and graphics.
5. Open source applications are tools and software where the core code has been made freely available to anyone, provided that any enhancements they make to it remain open source. Previously for geeks and techies, more business software developers are harnessing the power of open source to cut development costs and compete with established players.
6. With the acceleration towards international business and shrinking of the globe, multi-currency, multi-language and multi-national capabilities are becoming essential for even the smallest companies.
7. IT budgets are under pressure all over the world and all businesses - especially small companies - are being asked to do more with less. This leads to the innovation and creative thinking necessary to drive change, at the same time making established software suppliers weighed down by decades-old technology, just too expensive for all but the world's largest companies.

What software is out there for running your small business? Imagine a boxing ring; in the left corner, the undefeated banter-weight champion for managing small business financials is Quickbooks™. Quickbooks is one of the best finance packages available for small companies operating mostly in one country under one legal entity. Despite trying, other upstarts cannot defeat them in their home ground of accounting software.

In the right corner, the heavyweight champions that between them run most of the Fortune 500 are the twin brothers SAP™ and Oracle™. Bloated, expensive, complicated, and requiring an army of consultants and IT professionals to keep them tuned, configured and customized for each company's needs, there is seemingly no business requirement that SAP and Oracle cannot fulfill. No Fortune 500 CIO is going to get fired for selecting either one of these twins (if they do get fired they might have a lucrative career ahead of them in consulting in either package).

What's in the middle of the ring? If you are too big for Quickbooks and don't have the deep pockets and patience for Oracle or SAP, then what package should you adopt for your business? Well, there's good news and bad news.

- The good news is that there are a lot of tools and packages available.
- The bad news is that none of them stand out; they all have advantages and short-comings which you are going to need to be aware of before making a selection.
- The second bit of good news is that this eBook will take you through the journey to deploying the best tools available - from understanding what you really need, through selection and implementation, to using and improving on the benefits of your hard work.

This eBook is divided into chapters covering the life cycle of a small business (a chain of wedding florists) selecting, implementing and using business management tools to improve the way it runs. We examine the trials and tribulations as Beauty in Bloom's CEO Sarah and IT consultant Jasmine understand their business, select and implement cloud-based software and try to maximize benefits while minimizing cost. Feel free to read the whole book front to back, or dip into chapters as your needs dictate. Enjoy!

Contents

Introduction	1
Contents	3
Chapter 1 – Who Are You? Defining your Business Model	4
Chapter 2 – Defining Requirements: What do you Really Need?	7
Chapter 3 – Creating an IT Roadmap: Potential System Characteristics	10
Chapter 4 – Choosing a Partner: Plenty of Fish in the Sea	14
Chapter 5 – Three is a Magic Number: Pruning Back your Potential Partners	20
Chapter 6 – Implementation Overview: Taking Stock of the Big Picture	25
Chapter 7 – Project Preparation: Signing Contracts and Building a Team	30
Chapter 8 – Design: Teeing up for Long-term Success	34
Chapter 9 – Build & Testing: Guiding the Process	37
Chapter 10 – Go-live Preparation: Steady, Ready and Trained	41
Chapter 11 – Go-live & Support: Weathering the Storm	44
Chapter 12 – The New Solution: Continuous Improvements	49
Chapter 13 – New Rules, a Summary & Why iBE.net	53

Chapter 1 – Who Are You?

Defining your Business Model

Before you rush off headlong to compile checklists of your requirements and explore potential software solutions, it's important that you take the time to understand your own business. You need to identify your business model. To focus on what makes your company a success you need to think carefully about what you do. What sector are you in? Who is your target market? Are you delivering what your market needs?

Most companies evolve over time without ever changing their original business plan.

Adapting to change underpins the success of the human species, and identifying what works and what doesn't is key for business success as well. If you don't have goals and a mechanism to measure your progress then how do you know you are going in the right direction? Start with your KPIs (key performance indicators) and consider where your business is going right and where it is falling short.

Look past a stock answer about what your business does. Try to reduce your business to its core, a 100 word "elevator pitch" on your company. What do you need to get right? Does price, technology or service come first? If you can correctly identify your market's needs and how to turn those needs into profits, then you have a solid foundation. Without a clear and considered business model, you are building on sand.

Let's take a look at how our fictional florist, Beauty in Bloom, worked it out.

Defining Beauty in Bloom's model

The first task for Beauty in Bloom's CEO, Sarah, was to figure out her business model. She had always assumed that flower management was the fundamental key to success. By providing the best choice of fresh flowers in an inviting window display setting she imagined her florist would flourish. In actuality, it began to wilt under high overheads and a lack of business volume.

Like many new companies starting out, the original business plan for Beauty in Bloom was fatally flawed.

Sarah used examples of other successful business models combined with key performance metrics to identify where her business should be focused in order to profit. The process began with correctly identifying what her business model was. Preconceptions can be a real minefield here. She tried to reduce the core business down and set those future ambitions and expansion plans to one side. By identifying the business models behind a number of successful companies, Sarah was able to see Beauty in Bloom's model more clearly.

Here are some classic examples of business models Sarah looked at:

- **Walmart.** Let's face it - the world's largest retailer bases its business model on price and volume. Its ability to buy huge quantities of stock from China allows it to offer the lowest prices around.
- **Gillette** relies on a bait and switch model, much like the old Polaroid cameras. You buy a relatively cheap razor set, but that ties you in to purchasing packs of expensive razor blades in the future.
- **Starbucks** offers a premium quality product in an attractive setting that cultivates a strong brand that encourages people to pay more, despite the fact rival cafes are offering a much cheaper cup of coffee.

- With **eBay** the strategy is to provide a network for safe transactions and let sellers and buyers do the work. The network naturally drives down prices and establishes market demand through competition. Offering aggregation and price comparisons have spawned many online businesses because they do the legwork for the consumer and then earn a commission on the sale or referral.
- **Singapore Airlines'** business model is all about service: delivering the best experience for customers keeps them coming back for more.

Sarah realized that her business model was not what she had originally anticipated, but before she could work it out there were other factors to consider.

Sarah had to think about how Beauty in Bloom was organized. It was in the commercial, rather than public sector. It was local, rather than global. She was able to make quick decisions that allowed for more flexibility, but also more risk. As the CEO of a small business, Sarah didn't have a board of fellow directors to reach agreement with before pursuing a course of action.

When her husband first asked about SIC and NAICS codes she was puzzled, but she soon learned that Standard Industrial Classification and North American Industrial Classification System codes are all about classifying businesses and identifying their primary activity. It was a process that proved to be very helpful in understanding the key aims for her business and in identifying potential competition.

She had to check up on regulatory requirements for florists. It's always important to establish whether there are any industry-related or governmental requirements or regulations that your business must adhere to. Even if there are no regulatory requirements now, you must consider whether there might be when your business grows.

The final part of the puzzle was to define Beauty in Bloom's KPIs - or key performance indicators. By establishing the top four measures of success and creating metrics to analyze performance, Sarah was able to understand what was working for her business and what wasn't. As a result her initial business model was superseded by a new plan that focused on the profitable side of Beauty in Bloom.

Sarah's top four performance metrics were as follows:

- Average profitability per job
- Backlog
- Source of leads (referrals versus marketing)
- Revenue compared to fixed costs

She gathered data and measured these four KPI's weekly over a period of two months. The results were unexpected. Sarah realized that the most profitable part of her business, by far, was providing flower arrangements for weddings. She found that Beauty in Bloom was really about providing a premium service for couples getting married, and that the bulk of her new business came from satisfied wedding customers recommending her business to others. She also found that providing online photographs of wedding arrangements was an effective use of technology and more cost efficient than marketing dollars spent on newspaper adverts.

As a result she shifted from spending a lot of cash on fresh flowers for walk-ins, which involved a great deal of waste, and moved towards focusing on premium flower arrangements for weddings (which also gave her volume buying discounts). Her stores became a showcase for couples to visit and discuss their needs and she was able to reduce fixed costs by only buying the flowers she needed in bulk to fulfill her wedding obligations. By specializing and targeting a niche she was able to reduce her field of

competition and focus on providing a better service than her closest rivals in the wedding space.

In short, Sarah realized that her original business plan had been too ambitious and unfocused. She had been trying to do too many things with the business, and overheads had crept up as a result. By focusing on the profitable jobs for Beauty in Bloom and deploying resources for maximum return, she was able to establish a solid foundation.

Something as simple as the question “Who are you?” set Beauty in Bloom on the path to success. Every business should start with the same questions raised in this chapter before considering the part that software and technology tools can play.

Takeaways

There are various lessons that can be gleaned from our first chapter. Don't assume that you already know what your business model should be. You can refer back to these takeaways here or in “chapter 13 – a Summary”, as a quick reference guide.

1. Consider the business model that best fits your company. Your company can only really focus on price, technology or service. All of them are important, but you can only really excel at one. Choose which one and you already know more about your business.
2. Your core market is way smaller than you think; it's where you are a major player. It's better to be a big fish in a small pond. The more focused your market, the more focused – and therefore effective – your messages become. Remember, you can reach a smaller market with less investment.
3. Three to six measures define your entire business. Sure you can report on a bunch of stuff, but most of it won't be the difference between success and failure. For example, time to collect is only important to a cash-intensive business. Know these 3-6 measures by heart, benchmark against your competitors, and measure them objectively every week.
4. Never lose sight of your target customers, your reach, and your primary competitors. Consider the limitations of your company structure and keep regulatory requirements in mind as you scale.

Chapter 2 – Defining Requirements: What do you Really Need?

Now that you have considered what your business is really about, defined your business model, and identified key performance metrics, it's time to think about your software requirements for business management tools. It is very easy to get carried away at this stage. The last thing you want is to end up with a huge requirements document packed with thousands of entries. Creating a useful set of requirements is all about balance.

Carefully consider the level of detail you need to stipulate. Don't concern yourself with standard features that every vendor will provide, and also try to skip the obscure requirements that you don't really need. Establishing what you need versus what you would like is a vital exercise. It is a good idea to weight or score your requirements.

Don't undertake this task in isolation. You need to gather the thoughts of relevant staff, customers, and even suppliers to ensure that you have a firm grasp of the big picture. You may want to consider hiring a consultant, but it's best to use them as a sounding board and a second pair of eyes on your work. You don't want to charge them with creating the requirements document for your approval because you'll probably end up with a heavy report that you and your potential vendors will find too long to read.

Streamline your requirements as much as possible. Ideally you'll end up with 25 or so core requirements and you certainly shouldn't have more than 75. Before you assess potential software vendors it is worth including a couple of spoiler requirements. These are bogus demands designed to test their honesty and find out if they are just saying "yes" to everything. Spoiler requirements can prove invaluable in finding the right partner.

Here's how Sarah defined Beauty in Bloom's requirements.

Scoring the right requirements

To start with Sarah sat down with a cup of coffee and tried to write down all of the requirements she could think of. She thought through the complete process of a typical wedding job for Beauty in Bloom from start to finish and picked out the pertinent data points that any software should be able to track. She was careful to consider the level of detail required for each entry. To say that the business management tools would have to be able to handle financial information was too vague, but stipulating two decimal places for money fields was too detailed.

She also tried hard to avoid including generic software features like "password controlled access to the system". No vendor would offer a software solution without this, so including it in the requirements checklist would be overkill. Once she had a long list in front her Sarah began the process of scoring. By assigning a score of 1 to 5 to each requirement, where 5 denoted an absolutely essential requirement and 1 denoted a wish that would be fulfilled in an ideal world, she was able to see her business' core demands.

Several cups of coffee later, Sarah had a list of Beauty in Bloom's key requirements, but she was still agonizing over specific scores. She was anxious about missing anything that would come back to bite her later. The last thing she wanted was to forget something really important for her business and end up with a management tool that didn't do the job. It was time to get a second opinion.

First of all Sarah shared her list with her staff. She shared the list with her operations manager Rob, with Megan who ran one of her branches, and with her trusted delivery driver, Jennifer. She gave each of them a copy of the list and asked them to

independently evaluate her scores and add any requirements that they felt were missing. She then called a meeting with all three of them to discuss the results and question them about their thoughts. This allowed her to reassess her original requirements and she rescored where appropriate and tweaked the list to reflect their feedback.

Key requirements that everyone agreed upon included:

- Inventory tracking, including spoilage and shelf life
- Inventory attributes including price, cost, photos and seasonality information
- The ability to manage prices by arrangement or individual item
- Automated sales and use tax calculation
- Regional, time-sensitive and customer specific surcharges and discounts
- Secure online credit card and payment integration
- Fixed versus variable cost tracking for the store (including labor, overheads, overtime, inventory, capital etc.)
- Job management workflow
- Payment tracking for customers
- Number of hours and overall cost per job
- Email campaigns and newsletters for new leads and existing customers

The staff discussion also uncovered a few extra requirements and wish list embellishments such as the ability to track SEO (search engine optimization) on their website, the ability to respond to customer queries about job status directly from a smartphone, and the ability to generate reports filtered by different data points.

Sarah was pleased with the requirements list and after this exercise she was more confident that it covered all the bases. But she wanted to be thorough, so she asked a trusted supplier and a loyal (twice divorced) customer to take a look.

As a result of their feedback she added suppliers to the payment tracking, along with an email reminder system, and a preferred supplier option for the inventory attributes. She also learned that her supplier had recently been through a similar process and he recommended a good outside consultant.

For peace of mind Sarah hired Jasmine, the consultant her supplier had recommended, to provide a final safety net check. Jasmine suggested that Sarah include some fake, or spoiler, requirements before sending the document out to potential vendors. She explained that this would be a great way of establishing their intentions. By including a couple of requirements that made no business sense at all, Sarah would be able to weed out potential partners who were thoughtlessly saying yes to everything.

She included a requirement that every customer ID field should be at least 20 characters long and another demand that each user in the system should have two passwords – a primary password and a backup password in case they forgot the first one. Any vendor worth their salt would question these requirements and explain why they were a bad idea.

It took Sarah a while before she had a set of requirements for Beauty in Bloom that she really felt confident about, but she knew it was worth putting the time in now to avoid problems down the line.

The end result was a tight, well-defined set of requirements that spoke to the core needs of the business. It would serve as a great starting point to help identify management software that could really enhance the smooth running of Beauty in Bloom. It would also help Sarah find a software partner with integrity.

Takeaways

There are a few key messages to remember when you set out to define the requirements for tools and software that your business needs.

1. Keep it simple – you should be aiming for between 25 and 75 key requirements. A huge checklist is only going to muddy the waters and make it harder to find the right partners and solutions.
2. Make sure you consult with your staff, your suppliers, and even your customers. You can gain valuable insights about your business' requirements this way. You can also reach some consensus on your scoring system to correctly identify vital versus wish list requirements.
3. Get a second opinion from someone without a vested interest. Whether you engage the services of an outside consultant or ask a friend with relevant business experience, the feedback can help you to refine those requirements before you discuss them with software vendors. You could even ask an intern to run through the requirements for a fresh perspective.
4. If you plan on getting vendors to respond to the requirements directly then include a couple of spoiler requirements to test the integrity of your potential software partners. It's important to know that you can trust the vendor you eventually select and eliminate vendors who are simply telling you what they think you want to hear, rather than what is in your best interest.

Chapter 3 – Creating an IT Roadmap: Potential System Characteristics

You have a solid grasp of your business and your software requirements have been defined. Now it's time to consider your IT needs for the next five years. You should create a roadmap for your business' information management and IT tools going forward. In order to do that you'll need to make some tough decisions. Careful analysis of alternative options is vital because this stage of your journey will define what you select and deploy, and therefore how such tools will support your business.

Aim to create a workable plan that provides a clear outline of your company's future IT needs. You need to think about whether you want to deal with chosen tools on-premise or engage a cloud service. You can exert greater control over certain aspects by keeping software in-house, but you'll need IT expertise to cope with server configuration, software updates, back-ups, data security and integrity, and a host of other considerations.

Do you want to be able to pick point solutions off the shelf to target specific needs, or does a software suite offering an end-to-end solution make more sense? It's a classic short term versus long term conundrum as point solutions can be efficient and cost effective to begin with, but as your company grows and you add more and more of them, management and integration becomes tricky and expensive.

You need to think about the opportunities that mobile technology offers. Are smartphones and tablets appropriate for your staff or do they need laptops? Can your business embrace and benefit from the BYOD (bring your own device) trend? Are your employees always connected to the Internet?

Think about what are your business' core processes. Core processes are the ones that you must get right, the line between success and failure in your business. Execute a core process more effectively and you will make more money. On the other hand, consider supporting processes as ones which you and your colleagues simply need to do. It should come as no surprise that the majority of your energy, time and money should be invested in top-class IT tools for your core processes.

Do you like to micro-manage or delegate? How do you incentivize people and manage performance? Do you live week-by-week making sure there is enough cash in the bank, or are you more interested in investing your cash in capitalized assets to reduce tax?

All of these considerations should feed your strategy. Let's take a look at how Sarah worked up an IT roadmap for Beauty in Bloom.

Devising an IT strategy to take Beauty in Bloom forward

After her positive experience with the outside consultant when compiling her requirements list, Sarah's first temptation was to call Jasmine up again and hire her to draft a plan. Sarah considered the costs, but her husband was quick to point out that no one knows her business better than she does. Sarah realized that the strategy she was to create for Beauty in Bloom now would serve as roadmap for the next few years and it was simply too important to outsource from the outset.

She decided to follow the same approach as she had with the requirements. She would draft a plan, discuss it with her staff, then her suppliers, and only when she was satisfied would she call in the consultant to provide that outside perspective.

Her first decision was an easy one to make. Beauty in Bloom would definitely opt for a cloud-based solution. Sarah was happy enough with sending emails and she had used a template design to create Beauty in Bloom's online presence, but she had no appetite

for learning the necessary skills to set up her own servers. She didn't want to have to worry about configuring software, keeping back-ups, and ensuring that data was secure. She had no idea where to start with data center providers and disaster recovery plans. Beauty in Bloom couldn't justify the cost of a dedicated IT professional onsite, and besides there was no room to accommodate more people or computer equipment in any of the stores.

Cloud-based software would free her from worrying about the details. It would provide security, recovery guarantees, easy accessibility, and scalability, not to mention a significantly lower initial outlay in terms of up-front cash.

Her next quandary was whether to opt for individual or niche software, or an end-to-end package that would cover the works. Beauty in Bloom was using QuickBooks for their accounts, Salesforce for customer relationship management, and an antiquated retail point of sale (POS) system. Sarah was already struggling to integrate the three of them and the thought of adding in separate solutions to deal with project tracking, costing, analytics reporting, and inventory management, was too much. She figured a software suite to cover everything would be the best way to go, but resolved to discuss it with Jasmine later.

Sarah had recently gotten an iPad for her birthday and she was keen to be able to use it for work. She also wanted to be able to check up on the progress of individual jobs and inventory levels while on the road via her Android smartphone. She knew the sales team could use the computer in the store to keep things up to date, but she figured her delivery driver Jennifer would really benefit from being able to update job details via her mobile.

She knew the core processes that would dictate whether Beauty in Bloom succeeded or failed were all related to job management. It was vital to be able to get real-time updates on job progress and ensure that customers were never left hanging. Careful inventory management and profitability relied on minimal wastage. Bookkeeping, billing, HR and accounts were all supporting processes.

After discussions with her staff Sarah was pleased with her outline. Megan complained that the computer in her store was slow and temperamental and angled for a new laptop or tablet. She also argued that it would enable them to spend more time with the customer and input updates more quickly if they didn't have to go off to the computer. Sarah could see the sense and it would allow them to ditch the old machines. She also found out that Rob and Jennifer both had iPhones and could potentially use them for work if there was a way to plug into the new system.

She talked to her oldest supplier about the best way to implement a new system and he advised her to maintain close control over the development. She decided to manage it herself and establish a running month-by-month contract with the vendor rather than splash out on a big license purchase up-front. It was important for Beauty in Bloom to maintain a healthy cash reserve in case a number of big orders came in at once, which wasn't uncommon. Sarah was determined to pay based on monthly usage basis rather than investing a large sum upfront.

To finalize her strategy Sarah called Jasmine and arranged a meeting. Jasmine agreed that a cloud-based system made the most sense, but they spent some time discussing the merits of point versus suite solutions. Jasmine explained that point solutions can make a lot of sense in the short term, but that over time companies could grow to have ten or more separate packages all being upgraded at different times to each other, and that would inevitably create a management and integration nightmare.

Jasmine suggested that point solutions could work if managed correctly, but minimizing the downsides would depend on four key points:

1. They should only be considered temporary solutions.

2. The vendor chosen should have the ambition and skills to develop an end-to-end suite solution in the longer term.
3. Deciding and documenting which system has the “master” information from the start is vital to avoid conflicts.
4. Sarah needed to be able to set up or build her own interfaces to extract data and manage the system connections. She must not rely on promises from one vendor to integrate to another vendor’s software.

Jasmine also highlighted the potential virtues of social collaboration and explained how sharing ideas online and allowing input from different sources could boost the business.

After talking it through, Sarah felt more confident in her original gut feeling that an end-to-end solution would be best. She wanted to be able to have a single point of entry that would give her the definitive version of the truth about Beauty in Bloom’s performance.

Now Sarah had a roadmap that suggested a future suite of software for Beauty in Bloom which would manage the entire set of business processes, be based in the cloud, and be accessible from smartphones and tablets. She would manage the development directly and find a vendor to work on a contract basis with a manageable monthly fee rather than a big upfront cost. It was time to start looking at some software options.

Fake vs. real cloud software

If you opted for a cloud-based tool then it’s worth considering whether you are looking at a genuine cloud system or a fake. Genuine cloud systems are designed from inception for consumption via the Internet, as opposed to so-called or hyped cloud systems which are really just on-premise software installed remotely in a data center and sold on a monthly subscription model. How can you tell the difference? Well it is a tough one, but usually price is the biggest indicator. If the software cost is upwards of \$100 per user per month it is probably “cloud-bull” as some industry analysts like to call it, meaning expensive on-premise software hosted by the vendor or some third party in a data center on your behalf. True cloud software is designed to be scalable in a low-cost way. Provided there are enough customers to share the burden, the cost of such software, however sophisticated, should be in the range of \$10-\$75 per user per month.

Other red flags indicating fake cloud software are vendors that offer the same product in both cloud and on-premise options, because software designed for the cloud simply does not work if you try to install it on-premise. Additionally, look out for vendors who offer a range of software hosting options, vendors who have been offering the same software for more than a decade which they now claim is “available in the cloud”, and vendors who claim that being “multi-tenant” is a thing of the past, as these can be red flags for fake-cloud offerings.

Takeaways

When drawing up a roadmap for your own business, make sure that you consider these points:

1. Think about the pros and cons of a cloud solution versus installing on-premise software. Your choice must take the future expansion, ongoing maintenance and upgrades of your system into account.
2. Will social collaboration enable online brainstorming and information sharing that could benefit your business? What does collaboration mean for your business and with who?

3. Can you get by with point solutions as a stepping-stone towards an end-to-end suite down the line? You must choose your vendors carefully and think about your expectations from them long-term.
4. You could leverage mobile technology for use in your business, but it requires planning to ensure proper integration and security. It's always worth asking how mobile devices will enable your staff to perform at a higher level. Don't buy a bunch of tablets simply because they are in vogue right now.
5. Consider your preferred management style and the importance of cash flow in your business. The contract you put in place has to work for you.

Chapter 4 – Choosing a Partner: Plenty of Fish in the Sea

Armed with a solid business plan, a set of requirements, and an IT roadmap, it is time to take a look at some of the software products (or if you decided to build your own system, the software tools) on the market. The sheer variety may appear overwhelming at first glance. But, you already have the information you need to filter options and find the best solutions for your business.

In the last chapter you decided on the kind of system you want. The market can be divided into two broad categories to begin with: point solutions and end-to-end software.

If you intend to pick off-the-shelf packages that address niche aspects of your business on an ad hoc basis, then you might be looking more at point solutions. If you opted for a long-term view and prefer an all-encompassing system that's fully scalable, then you'll most likely be looking for an end-to-end solution long-term. Depending on how strongly you feel, what's available could change your views on this, so keep an open mind.

The first challenge is where to look for available solutions. There are many resources, both on-line and offline, to help you seek out available systems:

1. What your friends, business partners and competitors are using or recommending.
2. Researching independent sources such as Wikipedia, or software evaluation websites.
3. What you find online with suitable search terms such as "business management software" or "cloud enterprise software" (the natural search results, not the paid ads).
4. Stuff you heard about in the news or read about on blogs from apparently independent sources.
5. Sponsored lists such as industry group or association "recommended software lists". Software directories of one sort or another.
6. What your consultant or employees are recommending.
7. Adverts you have seen, including the ads that come up when you search for software.
8. Folks who call you up (telesales).

This list, which is not exhaustive, is in order from the least biased at the top, to the most biased at the bottom. That does not mean that you should only go with recommendations from friends and what you find in online searches or on comparison websites, but it does mean that you should keep your eyes open. Many articles or blogs that appear to be independent are actually authored by people from software companies, and employees and independent consultants can be incredibly biased based on their own good or bad experiences. A sensible yardstick is to ensure that you are checking at least four out of the eight sources above, plus one or two of your own choosing. The more ways you can find to look for available options, the more options you will identify (and provided that you manage the down-selection properly as explained in the next chapter), the better the outcome.

If you are looking for end-to-end solutions then try to find around 15-25 potential systems before you start to narrow down your search. If you are looking for point solutions you should find 10-20 options for each module or niche-requirement that you

have. So, if you are looking for 5-6 point solutions from the outset, you will need to get used to juggling 50-100 potential options at the same time!

Once you have a list of suitable software to peruse you can think about signing up for demonstrations, free trials, and white papers. You should set up a separate email address specifically for this purpose so that it doesn't distract you from your daily business inbox. More on how to manage all your new "best friends" from software companies you contacted is covered in the next chapter...but first let's look at how Sarah, our Beauty in Bloom CEO, worked out a list of potential solutions for her business.

Identifying suitable software solutions

As soon as Sarah ventured on-line to seek out software systems that might suit the needs of Beauty in Bloom going forward she felt overwhelmed in a sea of acronyms, many of which appeared to amount to the same thing. From PSA to ERP to CRM, she was disappointed to find that many vendors just assumed potential customers would understand what their software was about without giving a proper explanation.

She started by seeking out a list to explain the basic terminology relevant to her needs, which included CRM, FAS, SCM, TPM, PSA, HRMS, GRC, BI and a retail industry-specific term she had never really bothered to understand – Point of Sale or POS tools. An explanation of these and other key industry terms is provided later in this chapter.

Sarah started searching online, first for her chosen keywords such as "Florist software" which gave just one hit. When she understood that her business is really a mixture of retail, basic supply chain and customer service, she started looking for small business CRM, finance and retail point of sale systems, which gave far better results.

Anything that wasn't clear, Sarah would ask Jasmine or search for via Google and find relevant definitions and related articles online at trusted sources. By checking multiple sources she quickly built up a picture of what software would meet her requirements and fit in with her IT roadmap. Most of the systems dealt with specific pain points and so Sarah was able to discount them because she knew that she wanted an end-to-end solution and that it had to be cloud-based. She wasn't quite sure which systems were cloud-based and which were 'fake cloud', but she figured the truth would come out when she started to whittle down the shortlist.

Sarah had no problems finding potential solutions that might fit Beauty in Bloom via online searches. Her initial list of potential packages included SAP Business One, Brightpearl, Epicor, SAGE and Infor. She figured that this was rather too short a list, especially if two thirds of her potentials were outside her price range, so she decided to talk to her employees and Jasmine, and find independent recommendations and feedback on the available software in order to make a more informed choice.

Her husband suggested searching for some good comparison reports from industry analysts and Sarah discovered Gartner's Magic Quadrants methodology and reports for discovering potential vendors. A quick follow-up phone call from her consultant Jasmine also led her to software advice (www.softwareadvice.com), Technology Evaluation (www.technologyevaluation.com) and ERP Choice (www.erpchoice.com) - all websites where she was able to get lists of potential business management solutions.

She was able to generate a broader list of options based on specific criteria and taking into account the size, location and industry in which Beauty in Bloom operated. From this list she added six more packages: iBE.net, Zoho, Microsoft Dynamics Retail Management Software, NetSuite, AmberPOS (point of sale) and Plex. One of her employees also recommended Freshbooks.

She set up a separate Gmail account at Jasmine's suggestion, which she used to sign up for reports, white papers, comparisons and related research. This would help to keep the flow of promotional material and sales pitches out of her everyday inbox, plus she could give Jasmine and other members of her evaluation team access to the same email account. She would later use the same email address to enroll in trials and check out demonstrations with her short list of vendors. She also quickly learned to recognize and avoid downloading impartial reports sponsored by one or other vendor or designed to get her contact details.

After a few hours of careful research and comparison, Sarah had a list of 12 end-to-end software solutions that she felt might work for Beauty in Bloom. She thought that despite opting for end-to-end solutions, they should consider a couple of point solutions. So Quickbooks online (her current finance application was Quickbooks installed locally), Basecamp and Salesforce.com were added to the mix. She was looking for a system that was cloud-based, accessible and intuitive to use, easily scalable, and with support for mobile devices. Based on her initial reading of the various vendor blurbs on their web-sites, most of the list of 15 packages seemed to fit the bill nicely.

IT Business Software Jargon

One thing that will quickly hit home is the sheer volume of industry jargon. Some software companies are good at avoiding unnecessary jargon, but inevitably it will creep in from time to time. If you can't move Mohammed to the mountain, then move the mountain; Sarah decided it was worth taking some time to learn the basic terms. Understanding the jargon below helped her focus on the right software system for Beauty in Bloom. Here are a few explanations of the most common business or non-technical terms:

- Enterprise Resource Planning or **ERP** is all about automating and facilitating the flow of data from all the facets of your business. ERP was the original term for an end-to-end solution. As with any end-to-end solution, the aim is to streamline processes and increase efficiency by giving you real-time information on sales, distribution, stock and everything else that makes your business tick. ERP has, in the last ten years or so, also become associated with old, complicated software systems which need armies of consultants to implement, and with systems that are primarily focused on your "back-office" workers (like accountants, buyers, order entry reps etc.) as opposed to your "front-office" staff like shop-floor, sales or service people in the field. This is partially true - ERP systems support many front-office business processes, but they do so with antiquated non-user friendly tools, so in effect they tend not to support front-office workers very effectively.
- Professional Services Automation or **PSA** is designed to assist with project resource management and billing for service-focused companies and professionals like lawyers, engineers or management/IT consultants. They are like ERP systems for service-orientated organizations, skipping over modules like inventory management, purchasing or manufacturing and instead focusing on time and expenses, employee and contractor management and sophisticated time and expense based billing.
- Business Process Management or **BPM** tools map and workflow your organization's internal business processes, allowing you to keep track of where your documents are, what their status is, and who is the next person due to action the document. BPM tools are ideal in organizations that perform repetitive, static processes, such as claims or checks, but tend to fall down in companies where no two days in the life of a worker are remotely alike.

- Customer Relationship Management or **CRM** is another vital consideration, particularly for organizations dealing with a lot of leads and customers, i.e. a large volume of small value sales. CRM is a system for managing all of your company's interactions with current or future customers, from lead generation and marketing through to order capture and delivery fulfillment. Specific aspects of CRM systems can include the management of marketing costs, campaigns and resources or Marketing Resource Management (**MRM**), online or electronic commerce via websites between two businesses or between a business and an end consumer (**B2B** and **B2C**), and the electronic transfer of data using specific Electronic Data Interchange or **EDI** protocols.
- Supply Chain Management or **SCM** is about tracking goods from point of origin all the way through to the point of consumption. Often SCM systems encompass manufacturing, though more and more organizations outsource this to lower-labor cost countries. As a consequence, managing the supply chain through external suppliers or vendors is more critical nowadays. For this reason SCM can span multiple businesses, and the aim is to meet customer expectations as efficiently as possible, maximizing delivery performance while minimizing cost. Offshoots of SCM systems include systems focused specifically on Manufacturing Execution (**MES**) or shop floor information and machine interfaces, Advanced Planning and Scheduling (**APS**) systems that manage capacity and schedules, and Material Requirements Planning (**MRP**) systems that forecast when critical sub-assemblies and raw materials need to be bought or made to support the order fulfillment schedule.
- In addition to the various sub-sets of **SCM**, there are tools specifically focused on managing your inventory, such as Warehouse Management Systems (**WMS**), managing the quality of your product, including planned testing and quality defects or Total Quality Management (**TQM**) systems, Transportation Planning Management or **TPM** tools, and last, but not least, tools focused on the management of suppliers and purchasing cycle, Supplier Relationship Management or **SRM** (the opposite of Customer Relationship Management). These systems are essentially sub-sets of SCM in general which can be quite confusing.
- Enterprise Asset Management or Asset Life-Cycle Management (**EAM** or **ALM**) systems are concerned with the life of a part from when it is first designed and produced to how it is managed and supported in the field, typically at a customer's site, and finally how it is disposed of. Such tools are normally focused on post-delivery support, repair, and maintenance of complex products, which by their nature are cheaper to repair than replace.
- Business Intelligence or **BI** (sometimes called Business Intelligence and Analytics or **BIA**) is an over-arching term for storing, analyzing and reporting on large volumes of data, normally in an offline mode or not quite in real time. BI systems are often touted as strategic decision support tools as opposed to providing operational reports. The recent trend is for BI tools that savvy end users can create reports and dashboards on, without needing IT support.
- Governance, Risk and Compliance or **GRC** solutions relate to how senior executive decisions are supported with information, risk management and compliance with regulations in a company. GRC solutions are typically focused on access controls, especially since the U.S. Sarbanes-Oxley or SOX Compliance Act. Sometimes GRC systems include strategic forecasting and planning for both financial and operations metrics, such as sales and operations planning (**S&OP**).

- Project and Portfolio Management or **PPM** covers the management of one or many projects including performance monitoring, cost vs. benefits, and assessing where to focus investments between competing projects.
- Product Life-cycle Management or **PLM** is the process of managing a product (or service) from initial concept through to design, production and post-sales support. It is naturally focused on the design and development stage of a product, and has strong overlap with project management tools that are also part of PPM, and often coupled with connections to Computer Aided Design (**CAD**) software.
- Human capital or human resources management (**HCM, HR, HRM, HRMS, HRIS** or Employee Relationship Management - **ERMS**) systems help you to manage your employees and external workers, from recruitment through to termination. This includes all stages in between, such as employee records, career development, performance feedback, learning, organizational planning, payroll, benefits, life events, vacations and sometimes even time and expenses.
- Accounting software concerns keep up a company's books or general and sub-ledgers (receivables, payables, assets etc.) so that the company has a good handle on its cash, profit and loss, and balance sheet, and can accurately provide tax and other regulatory financial reports. Reconciliation with the company's bank accounts and integration to sales, purchasing, inventory and production systems are often included in accounting systems that are part of end-to-end solutions. For some reason accountants never came up with a three-letter acronym for their software module other than **FAS** or Finance and Accounting Systems.
- Enterprise Content Management or **ECM** is nothing more than a store for structured and unstructured information held by the business, typically with some way of organizing this data into folders, pages, documents or sites and some way to search the data. ECM systems often sit alongside ERP or BI systems to provide an alternative way to search data such as emails, web-activity logs or other less structured data that does not fit a rigorous database schema.
- Independent Software Vendor or **ISV** is a term often used by end-to-end solution providers such as ERP software vendors for other software suppliers who provide "plug-ins" to complete gaps in the end-to-end software provider's portfolio.
- Companies that are focused on implementing the software (for a fee) are often called Systems Integrators (**SI's**) and companies that resell a software vendor's product with their mark-up or commission, sometimes bundling in implementation or front-line support services, are often called Value Added Resellers or **VARs**.
- **Configuration** is where you enter preferences or settings in screens and tables specifically provided by the software vendor for the purpose of controlling how the software will behave, defining customer-specific 'rules'.
- **Customization** is where you write scripts, connect to outside or third-party software tools, or add on customer-specific programming logic to extension blocks or **API's** (application programming interfaces) provided by the software vendor for that purpose.
- **SaaS** or Software as a Service relates to the provision of software over the internet or from the cloud as explained in chapter three.

As you can imagine, you can waste countless hours learning about acronyms and industry terms. Sarah did not want to become an expert, but she needed to be able to converse with consultants and software companies. She studiously ignored any technical jargon or vendor specific jargon, and most retail industry-specific terms she already knew.

Takeaways

When you begin to explore the market you'll find a wealth of software solutions and vendors vying for your attention. Here's what you need to know to build up your initial shortlist:

1. Use multiple sources to look for systems on-line and offline, formal and informal. The more sources you use for information, the better overall picture you will get.
2. Get to grips with some basic industry jargon to help you gain a better understanding of what is on offer. See above for common explanations.
3. Refer back to your requirements and your IT roadmap as you go.
4. Research systems that might tick all of your boxes by checking out independent software comparisons, white papers and customer reviews.
5. Always seek out independent opinions that are divorced from vested interests. Analyst reports and independent comparison websites can really help with this. Don't be fooled by so-called independent reports and blogs written or sponsored by one or other software vendor.
6. Aim for 15-25 end-to-end solutions or 10-20 point solutions (per module or feature) in this initial round. There is plenty of time for down-selection later.
7. Set up an email address specifically for the research and demo phase of your journey so it doesn't distract you when you need to focus on the day-to-day running of your business.

Chapter 5 – Three is a Magic Number: Pruning Back your Potential Partners

You've come a long way since this book began. Over the first four chapters you defined your business, identified your requirements, and created a clear roadmap. Armed with that knowledge you have cut through the jargon minefield and found up to 20 end-to-end packages with the potential to meet your needs (if you're looking at point solutions that figure will be higher). Now it's time to be ruthless.

You might have 10-20 solutions in mind, but it's a safe bet that twice as many software vendors are now courting your attention with regular follow-up calls. This can't go on. You have to down-select and three is the magic number for negotiating. You're ideally looking for three end-to-end solutions, or two to three options for each point solution. You don't want to spend too long negotiating, but it's also important not to lock yourself in to a specific solution too early. Unforeseen limitations in the software or a vendor playing hardball at the negotiating table can run the process off the rails.

Use your requirements and your roadmap to determine your demands and establish which of your potential partners can meet them. It's important to distinguish between what is critical for your business and the things that would simply be nice to have. This distinction dictates your demands versus your wish list. If three or more of the solutions meet your demands then cut the rest immediately. Don't be tempted to keep something on the list because of a single cool feature, and consider whether persuasive or aggressive sales staff may have unduly swayed your decision. Objectivity is the aim.

If you're struggling to find three options that meet your demands then you need to go back a step and ensure that you didn't miss any good solutions. You could also scrutinize those demands to make sure that no wishes snuck in there. If you can't identify a trio that meet all of your demands then you were too demanding, or you did not look hard enough for solution options.

Still can't find three good options? Your last recourse is to consider custom development or a hybrid solution to plug the gaps, but this will be expensive and fraught with difficulty. Be careful to calculate workaround costs. Viable combinations of multiple packages working in tandem will have been tried before, so you can learn from other companies that have encountered the same problems. If you are doing some novel and unique combination of packages you will probably get your fingers burnt.

When you have more than three potential options it is time to compare them and there are two solid methods to do it.

1. Weighted scoring – whereby you create a list of criteria and every entry is measured and weighted to compute a total score, typically on an Excel spreadsheet.
2. Hands-on evaluation – where you and your team try the software out and see how it delivers in reality, assessing functionality, usability, vendor support and performance.

The smart play is to do both. You need to consider factors like cost, vendor flexibility, quality of documentation or training, ongoing support, and even vendor financial stability. Non-product related criteria are important, but shouldn't ever contribute more than a third of your overall score. You also need to try the system out for yourself. PowerPoint presentations and vendor demos are designed to sell – you can't rely on them alone.

For a weighted scoring exercise to be successful you must come up with your own criteria and make sure they're not too broad or too narrow. You list the criteria in a

spreadsheet with a consistent scoring mechanism such as High, Medium, Low or 1-5 or 1-10, being careful to write down and have everyone in your team agree exactly what is meant by “High” or “6 out of 10.” Then you allocate a percentage weight against each criteria and compute the sum of each criteria score times its weight. Excel can do this in a jiffy.

You should never allow a vendor to dictate the criteria or score themselves. It is vital to engage an independent party free of vested interest; with consultants remember to check up on their track record in case there’s a bias in how they score. If you are going to take scores from vendors then make sure you validate them by getting independent references from their customers, checking online feedback sites, or reading detailed independent assessment reports. “We can’t do that” is simply not in most vendor vocabularies!

The more people who have input to the scoring the better, so use your team to get a balanced view. You and your most valued staff, along with key customers or partners, and your consultant, should give you a big picture view. This will help you to avoid personal bias and it will foster broader buy-in from your workforce. Everyone likes to be consulted about major changes and sometimes worker buy-in to a decision is more important than making the best decision.

When you go hands-on with the software make sure you are forearmed with objective measures so you can balance your gut feelings with predefined criteria to arrive at a fair assessment. Write down a script of what you want to achieve based on real business scenarios. Don’t let the vendors give you their pre-canned scripts, but equally don’t be too ambitious in your script. You need a script that you would normally get through keying into a user-friendly tool in about ten minutes, not a complete month’s worth of business transactions. Focus on your team’s hands-on experience with the software, without factoring in demonstrations and presentations where you aren’t in the driving seat.

The best software vendors will try to change the rules and move the goalposts. They will try to positively influence your assessment in whatever way they can. Watch out for attempts to bypass you and influence other decision makers in your organization, or stalling tactics designed to alter your schedule.

It’s important to be vigilant and gather as much independent information as you can. Looking beyond your direct competitors you should find plenty of other customers in a position to talk about the same software you are considering. The more unbiased views you can gather, the easier your evaluation becomes. Just remember to “pay it forward” by sharing your experiences with other companies undertaking this journey in future.

You might find this process daunting, but remember that you are closing in on the solution that will transform your business and lay the groundwork for increased efficiency for years to come. It proved to be a rocky path for Sarah and Beauty in Bloom, but the final destination made the journey worthwhile.

Pruning back the options

Sarah was raring to get her cutters out and start the work of trimming back her software options. She imagined that cutting out software packages that didn’t fit would be easy, but reaching a streamlined bouquet of options proved to be much harder than first thought. Jasmine had advised her to reduce the line-up to between two and five options, but Sarah’s husband was adamant that three was the magic number to aim for.

She started by listing out demands. Using her requirements and the roadmap she had drawn up, it was easy to create a filter that could be applied to the potential solutions, though she had some difficulty distinguishing between definite requirements and wish list items that she could probably live without. A few of her demands jumped back and forth between the two columns.

Ever since Sarah had begun to identify software packages that might work for Beauty in Bloom she had been inundated with sales calls. She was relieved that she had set up a separate email address to handle the process because there had been a steady flow into the inbox. She was sure some of them weren't even packages that she had looked at; it felt as though every developer in the country had been alerted to her search and by a massive coincidence they were all offering exactly what she needed.

By cross referencing her demands and working through the list Sarah cut the contenders down to five and did her best to dissuade the others from wasting any more time trying to convince her. In the end she had to tell one particularly persistent vendor that she had already signed up to a competitor to get him to stop calling.

Sarah was keen to get a hands-on look at her top choices as soon as possible, but Jasmine advised her to establish criteria before meeting with them. She explained that a list of objective measures would be the best starting point and sent through an example Excel document. Sarah was able to fill in her criteria, each to be scored after assessment on a scale of one to five, and she also assigned a percentage allocation to weight them. This would give her a total weighted score for each solution.

Proud of her organization and forethought, albeit at the behest of her consultant, Sarah excitedly showed off the criteria document to her husband. He soon brought her down to earth by picking at her list and criticizing the broader criteria. One entry, "Vendor's company" was eventually split into: stability, flexibility, support level, and quality of consulting staff. There were other entries where he identified the opposite problem, pointing out that "quick adoption" and "having an intuitive user interface" were essentially the same thing and could be merged.

After an hour or so of tweaking Sarah was silently fuming, but her criteria list looked solid. She banished her husband to the kitchen to make her some dinner while she called Jasmine to run through the list with her. They agreed that the next step was to get a hands-on look at the software and she would rope in her employees Jennifer and Rob to assess for themselves. That meant persuading them to take time out of their day-jobs and Sarah resolved to arm herself with a tasty cake incentive before asking.

The first half day of "hands-on testing" was frustrating for everyone concerned. Sarah, Rob, Jennifer, and Jasmine turned up at the vendor's offices and sat through a lengthy presentation. The sales force seemed to be taking it in turns to present the same basic information in different ways and by mid-morning Sarah was fed up of PowerPoint slides.

She insisted on some hands-on time for everyone after they had coffee, but they ended up with two computers between them and the vendor's staff seemed intent on hand-holding. At one point the lead solution engineer wrestled the mouse from Sarah and bypassing her questions clicked through the interface at light speed to show off an unrelated feature. The whole experience left a bad taste in Sarah's mouth and she felt she'd asked her staff to give up a morning for nothing.

Back in the office the following day the questionnaires based on Sarah's criteria list had arrived back from all five vendors (she had set a fixed deadline for them at Jasmine's suggestion). The vendor they'd visited the day before had scored themselves 10 for every category but one, and the scale only went up to five. Cross referencing with online reviews and customer feedback Sarah soon found plenty of evidence that a 10 for customer service was not merited. Combined with her experience at their offices it was enough to knock them down to the bottom of the list.

In the days that followed Sarah and her team were able to get a hands-on look at the other software packages and she was bullish about preventing vendor input from disrupting the process. At her husband's suggestion she insisted on them coming into her office and luckily this meant fewer people to present slides; a couple of vendors actually did the whole evaluation over web conference and phone. The last thing she

needed was another hijack situation turning the testing into a demo, or worse, a presentation, so she sent out a small list of rules to the remaining four vendors: Maximum 10 slides; demo's won't count toward evaluation, only hands-on; if anything needs to be installed then bring four extra laptops with everything ready.

One of the vendors made a point of taking Sarah aside and started to ask a lot of detailed questions about her time scale and budget, but she refused to give away anything. Truth was she didn't know. Jasmine had already warned her against sharing too much information and made it clear that it was important to create a level playing field to achieve true objectivity in assessing the possibilities. After the testing, each member of Sarah's team was asked to give their impressions in isolation based on their listed criteria.

Meanwhile Sarah called other companies that had used the software she was assessing and trawled the Internet for customer reviews and feedback. She was even able to find an independent report covering two of the packages. All of the data she uncovered was cross referenced with the scores that the vendors had assigned themselves and what Sarah had gleaned from her own hands-on testing. The final step was to collate the feedback from the rest of her team.

There was a general consensus on the right choice, but Rob seemed to have a different take on one of the vendors than the rest of the team and Sarah wanted to know why. It was as if he'd been given information that wasn't shared with everyone else. She scheduled a meeting to discuss it with him and found out that the sales lead for that vendor had contacted Rob directly through the store and taken him for lunch. That was enough for Sarah to discount the variance and go with the consensus, putting that particular software vendor in third place, excluding price.

She had decided at the start of the process not to include cost as one of the weighted criteria. Having arrived at her shortlist of five vendors and assessed them thoroughly with weighted scoring and hands-on evaluation of the software, it was finally time to factor in the cost. There were only three packages that had come through the assessment positively and one of them was too expensive. Sarah had successfully narrowed it down to two. She was confident that either of them could boost her business and help catapult it into the big leagues. According to her research there were more than 16,000 florist stores in the U.S. and Sarah was determined to make Beauty in Bloom the best chain of florists there was.

The next challenge on the horizon would involve putting a team together and negotiating the final contract before implementation, but for now Sarah felt she had earned a nice glass of red wine.

Takeaways

Cutting your selection down can be a grueling process, but it should also be exciting as you narrow in on the right choice. Here are some key takeaways to help you in your task:

1. Always dictate your own criteria and weighting system to assess potential partners, don't be tempted to adopt criteria from a software vendor or biased consultant.
2. Independent advice is valuable so seek it out from customer reviews and online feedback, third-party reports, independent consultants, and by calling other companies that have used the same software. Always try to check for potential bias. Remember the more sources you gather information from, the more reliable it is.
3. Empower your staff and enlist their help to narrow your selection. It gives you a big picture overview and it secures their buy-in.

4. Insist on hands-on testing time with the system where you are free to explore it for yourself. Don't settle for presentations, demonstrations, or guided tours.
5. Don't allow vendors to influence the process by going round you to other staff or stalling for time. Make sure they all have exactly the same information to create a truly level playing field and show that you are in control.
6. If you give dates to vendors when decisions will be made try to stick with them.
7. Always keep the bottom line in sight and be careful about the impact of lost time on the day-to-day running of your business.

Chapter 6 – Implementation Overview: Taking Stock of the Big Picture

Now that you have selected a package you are no doubt itching to just start using it. All you need to do is to send a memo to your employees, partners and key clients about the fantastic tools you found and the efficiency improvements and cost savings will start rolling in, right?

Not quite, you've got to go through an implementation process, even for a point solution.

Implementing the software involves all the steps from initial planning and selection (chapters 1-5) up to the point when you turn on the system for real and stop using your existing tools. This final step is called the initial "go-live." The implementation includes setting up the tools to behave as you would like, organizing your business better and getting ready for the go-live. Let's consider each one of these concepts in turn.

- **Setting up the software** can involve anything from selecting your industry and putting in basic data like users, employees, projects and orders, to custom developing add-ons or integrating third party plug-in solutions. Most enterprise business applications have a range of settings, configuration or preference screens where you can indicate how you want the system to behave. This ranges from how documents should be numbered to workflows or report layouts. Keying in options, preferences or settings in screens and tables specifically provided by the software vendor for this purpose is often called 'configuration'. Connecting outside or third-party software, or adding on scripts or programming logic to extension blocks, API's or user-exits provided by the software vendor is often called 'customization'.
 - Configuration vs. Customization loosely addresses the way people tweak the software settings, scripts or code to behave in a certain way. Configuration is more about changing settings and simple scripts and customization is more about complex scripts, HTML layouts or code. As mentioned in chapter four – never modify the underlying source code and avoid software vendors who recommend this as a strategy.
 - A key and often under-estimated part of setting up any new system is mapping your data - particularly static information that describes your business such as products, partners, resources, sites and employees (i.e., your master data) - to the new system.
 - If you didn't go for a cloud-based system then a significant part of the system set-up effort will involve provisioning and setting up hardware, installing the software and getting all the operational procedures like back-up, disaster recovery, security, monitoring, load balancing and system administration established.
- **Organizing your business** is needed at four levels: covering people, processes, technology and information.
 1. Organizing your people ranges from reviewing job descriptions and policies and seeing what needs to be revised in light of the new tools and processes you are implementing, to making changes in reporting lines and organizational structure, to retraining or even replacing certain staff because you can automate some tasks while introducing new skill-needs into your organization.

2. Organizing your processes involves understanding what you do and how you want to run your business differently in the future, getting a standard and best-practice way of doing repetitive tasks so that you don't keep on re-inventing the wheel, and writing it all down. This can become a science project; vast consulting businesses have been built up on the merits of "process re-engineering". We are not recommending that you bring in a bunch of Oliver Wight consultants and start plastering your walls with brown parcel paper, post-it notes and string. But we do recommend you look at your core business processes, assess what is unique to your business vs. something you just need to take as a best practice from somewhere else, and informally map out what you will do. Then get everyone to buy into it and make someone in your company accountable for policing the processes going forward while improving them at the same time.
 3. Organizing your technology is really covered by the software set-up tasks outlined above. However, you might have to make other infrastructure or technology changes to suit the new system such as upgrading browsers, getting people some tablets, or a faster or more secure Internet link.
 4. Organizing your information involves finding your data as it currently resides on legacy systems, spreadsheets, paper and other databases in your company, getting it out and cleaning it up ready to upload to your new tool. This can be complicated by the fact that the new tool requires data in a specific format including a host of fields or attributes you did not even know you needed such as "valuation method," "product type," "posting key" or "planning group."
- **Getting ready for the go-live** is something you do once per go-live, helping you smooth over the bumpy transition from your current ways to the new way. Are you thinking that the new software will work the same as your old tool, so there is no effort in the transition? This is unlikely. If your old tools were fine then why go to all this trouble to replace them? More likely the new solution and resulting tweaks to your processes, policies, organization and people mentioned above involves quite a transition. Getting ready for the go-live includes your people (aka, training and education), change readiness, as well as communication to external partners, customers and suppliers. It involves ensuring that you are prepared in case Murphy's Law strikes at the moment you turn on the new system. This preparation includes anything from printing out paper copies of key documents to asking your suppliers to delay their shipments to you for two weeks around the go-live to let you ramp up your warehouse department slowly. One company gave all their employees an extended Christmas holiday and asked them to come back department by department one day after the next. Another made every single employee's bonus for that quarter dependent on success and acceptance of the new system, even if they had no involvement in the project. Both go-lives were remarkably smooth. Less than smooth go-lives are discussed in chapter 11.

At this point you might be wondering what you have let yourself in for and wishing you had read this chapter before embarking on this journey. If you picked the right tool and the right team you don't have to spend months or years paying a busload of consultants to pick over your business and leave you with a system that no one can use.

Divide the Implementation into Five Phases

You are about to create your plan and lock yourself into a process, so it's important to pause for breath and take in the big picture. Like all problems the easiest way to solve it is to break it down. Let's break down the implementation process into five phases:

1. **Project Preparation & Benefits Case** is about signing contracts, making sure you have a solid team in place with any needed temporary work re-assignments or back-fills, and a solid supply of coffee for the late nights ahead.
2. The first real phase is the **Design** – business design (your processes), organization design (your people) and software design (what tools and technologies you will use). For commercial off-the-shelf software you are not embarking on a custom development so the design phase should include hands-on configuration or toying with the settings provided by your software vendor, at least in trial or test mode. But any complicated scripts, forms, reports and interfaces you will want to map, scope and specify as ready to build and test in the next phase.
3. The next phase is **Build & Test** which is where you, your consultants and/or the software vendor completes the customization and configuration required, including any third-party plug-ins and custom forms or reports you deemed necessary during the design phase. In addition, this phase must include a thorough end-to-end system test, including aspects such as unit or module functionality, sunny-day vs. what-can-go-wrong, day-in-the-life testing and sometimes performance or stress testing. User acceptance testing is often part of the Go-live Preparation phase, but some companies include it here. Including the user acceptance testing early is a neat trick to get more users familiar with the new system in advance of the go-live stage. In parallel to Build & Test you need to plan your go-live, deciding what is being transitioned at what point and ensuring everyone and everything will be ready and trained at the right time.
4. The next phase is typically called **Go-live Preparation**, Final Preparation, or Acceptance & Training. This phase is largely focused on your end users (training, education and acceptance testing) and getting your data into the new system. This phase completes the planning, rehearsal and contingency measures around the go-live, at the same time as making final tweaks to the tools, and finalizing your processes and policies.
5. The last phase starts with the actual go-live or cutover. This includes immediate post go-live **Support** as you ramp up business operations, assess and measure any benefits and transition to a long term mode. Long-term support is where the core team can go back to their jobs and support is provided by permanent internal staff or by your software vendor.

There are many available online resources featuring case studies and lessons learned from other implementations, key success factors, ERP-pitfalls, best-practice system implementation approaches, agile or six sigma, consulting company-specific methodologies that allegedly never fail, and so on. We encourage you to look at these but don't buy into anything you don't understand (it does not have to be that complicated). Don't go with something proprietary to one vendor unless the value of it is really compelling.

Above all else, talk to other businesses. Even if they did not implement the same software for the same industry in the same sized organization, talking to someone else who did this journey before you with a similar product in a similar industry will be more beneficial than anything else you can do to prepare yourself. Talk to three or four and you will become a true expert. Talk to ten or more and you can write your own eBook.

We'll deal with these five phases in the next five chapters, but first let's take a look at how Sarah understood the overall implementation process for Beauty in Bloom.

Blooming or Blooming Heck! Sarah's Implementation

Sarah already had a clear idea of who she wanted to include in her team for implementation, but before calling them all together she wanted to pause and take stock. Her potential vendors had been pruned back to two and thoroughly vetted. Sarah knew that the contract negotiation phase would have to take place soon and she was nervous about it. She had no problem securing a great deal with her Californian flower suppliers, but these contracts would be a whole different level of complexity.

In order to get a little more perspective on the process Sarah had lunch with a local CEO she had met at a networking event. Although he led a catering company, she knew that they had recently implemented one of the two chosen end-to-end systems, so she thought he might offer some insight or advice. He painted a picture of drawn out legal wrangling over the contract, a project team with disparate ideas about how to move forward, and consultants determined to squeeze every last penny from the budget and make the project last as long as possible.

He advised caution when hiring consultants and told Sarah she should opt for a fixed price deal to keep the timescale and cost under control. He also outlined the importance of incentivizing everyone for success, but making sure that success meant the same thing to everyone on the team. Despite the negativity when Sarah asked him if he would do it all again he said yes without hesitation and explained that business performance had improved measurably since the new system came online. Sarah was grateful for the insight, but the task ahead appeared more daunting than ever.

With stress levels soaring, her husband insisted that she take a break. She reluctantly allowed herself to be driven to a local seafood restaurant for dinner. As was so often the case, they didn't get past the starter before talk turned from family to business. After a resigned sigh, her husband told her to take a deep breath and just work through the whole process step by step before considering each actual decision. It was time to look at the big picture.

Beauty in Bloom's implementation plan was sketched out in skeleton form on a napkin. The key deliverables were clearly identified and the necessary milestones were outlined. Part of the problem was that some of the phases had to run in parallel. Having talked it through, Sarah felt more relaxed, but she was itching to get it all down.

By the time they got home Sarah felt confident about going forward. She left her husband nursing a whisky in front of the TV and retreated to the office to get her thoughts down on her computer before they melted away. She knew that understanding the implementation phases beforehand would help her through the process smoothly.

With the light of Sunday morning shining through the windows, she returned to find her husband asleep on the couch. She gently roused him and led him to bed. It was time for a good night's sleep. The week ahead would be challenging, but Sarah felt excited and energized. She was ready to prepare a detailed implementation plan and she felt sure it would go well.

Takeaways

Think of the implementation as three discrete and parallel work-streams running across five sequential project phases, as outlined in the table below:

Project Phase	Software & Data Set-up	Process, People, Technology & Information Organization	Readiness for Go-live
1. Project Prep. & Benefits	Scope, bolt-on or plug-ins required, vendor contracts	Benefits case, project team (internal & external), key drivers for change	Go-live or deployment strategy (phased vs. big-bang etc.)
2. Design	Configuration settings. Specify forms, reports, enhancements or scripts.	Organization design (people, job changes) Process design. Supporting technology or hardware required. Analyze data sources & clean-up required	Training / education effort estimate
3. Build & Test	Build forms, reports, enhancements, workflows or scripts. System testing, end-to-end scenario testing, sunny day testing, performance or stress testing	Job descriptions, policy updates, data clean-up and test-mode manual data entry	User acceptance testing to get more people using the tool (optional). Initial cut-over planning
4. Go-live Preparation	Final system tweaks (e.g. acceptance testing issues). Finish any late reports, forms, workflows etc.	Data load rehearsals. Organization changes	Training delivery. Detailed cut-over planning. Communications. Contingency measures
5. Go-live & Support	Fixing issues	Measure & track benefits	User training. Measure business efficiency (ramp-up operations)

1. Stop, take stock, gather your data and create an overview of the big picture. Every challenge can be broken down into manageable phases.
2. If you fail to prepare then you are preparing to fail.
3. Always talk to other businesses that have done a similar implementation as yours, and then share your knowledge with others who follow after you.

Chapter 7 – Project Preparation: Signing Contracts and Building a Team

As you develop your implementation plan you will also need to build a team that can help guide your business through the process. This team will naturally include the best and brightest members of your staff and some people from the software vendor's company. You'll also want to consider hiring consultants. Part of the project preparation phase concerns contract negotiation. By the end of this phase you'll have a core team, a signed contract with your software vendor and potentially with a systems integrator, and you'll be ready for the Design workshops.

The first challenge is to choose the software system and the vendor you want to work with. You've probably decided on the vendor you want by this stage, but keep your second place choice as a backup in case negotiations with your first-choice do not pan out. If the vendor knows that they are the only company you are talking to then they are in a much stronger bargaining position.

Software contracts can be challenging. You have a choice between accepting the vendor's one-size fits all terms of service or end user license agreement (EULA) vs. negotiating something to suit you. The answer is try to do the first—most software vendors are fair and have spent time building a water-tight contract which suits both sides. That's not to say you shouldn't have a lawyer or a friend with a legal background review the contract first. Just watch out for "lawyer justification syndrome". You don't want a lawyer who identifies problems with the contract purely to justify their fees. You want a lawyer who will tell you what's important and what's not, so you don't waste your valuable time or money on the unimportant stuff, not to mention the loss of goodwill that results when the software vendor marks you down as a 'trouble-maker' because of your over-zealous lawyers.

In the negotiating phase it's important to remember that a win-win is the best outcome. Just because you can score a win-lose over the vendor doesn't mean you should. Think about this as the start of a relationship that you expect to last for the next few years. Things will go more smoothly if each party feels they have a good deal.

That sentiment extends into incentivizing the entire team for success. If everyone concerned - whether it's your staff, the software vendor, the consultant, or a staffing firm to fill urgent resource shortages - is rewarded based on deliverables for the project as a whole then you are all pulling in the same direction.

A tricky question that will inevitably pop up early on is what to outsource. You'll need to weigh the pros and cons of consultants versus doing it in-house. If you are a successful small business then most likely nobody has the time to sit around discussing process activities, report layouts or training and communication plans. But someone has to do it.

Again there is no one-size-fits-all, but our recommendation is a minimum of one and maximum of one to one. In other words, include one outside party or consultant in your implementation team and never use more consultants than you have internal staff members working on the project. Be creative; it is amazing what can be done with temp workers, students and friends. Whether you need them to back-fill current staff going into your implementation team, do simple tasks like manual data entry or if you are bringing in people with particular skills, there are ways to get outside help without breaking the bank and without signing your life away to the likes of Accenture, IBM, CGEY or Deloitte.

People often talk about governance or steering committees for these kinds of implementations as a way to ensure executive support and to escalate any issues or risks that need senior management attention. Steering committee meetings are also a

handy way to bring representatives from all companies involved to the same table to make sure everyone has skin in the game. Aim to have one or two steering committee meetings per month, with the first and last ones face to face with all parties, and follow a formal agenda with presentations by the project lead and input from all sides. Have an agreed issue raising and resolution process as well as an active list of project risks and countermeasures.

Big-bang or bit-by-bit - deployment strategies

Another thing you need to consider as part of your plan is how you want to go-live. Do you want to convert everyone and everything on the same day, or in dribs and drabs? It might not surprise you to learn that transitioning everyone and everything on the same day to your new system is a much higher risk, but gets the whole job done quicker and normally at a lower cost.

There is no right answer, but a sensible rule of thumb is that if you are transitioning over fewer than 10 or more than 1000 users in one go-live, then you are either being too cautious or too aggressive. If your project timeline is four-fifths dedicated to go-lives, then you are either a Fortune 500 company or thinking like one. If you are replacing more than ten old systems with new systems at the same time, perhaps with some custom development, then you are in for a bumpy ride. Expect your management team to swear at you after going live as they have to bring in emergency temps and extol their staff to work overtime to catch up on day-to-day transactions while you sort out all the go-live issues.

You are no doubt wondering: How long will it take and how much will it cost? It depends on your company – the complexity of your business, cleanliness and availability of your data, and readiness of your organization to change. And of course it depends on the tool you select. But it depends less on the tool than you might think. Of the three work-streams mentioned in chapter six, only the system set-up depends on the tool.

Organizing your business and getting ready for the go-live is roughly the same whether you are implementing Oracle or a new spreadsheet template.

It is the complexity of your business that is the real driver here.

Again there are some rules of thumb you can look at, if nothing more than to compare to what your software vendors, consultants and reference customers are telling you.

- Time spent in Design should be roughly the same as Build & Test.
- Neither should be more than two months unless you have more than 1,000 employees and a large-scale go-live planned.
- The go-live time for an end-to-end solution, even in a small business, will be at least three months long.
- Anything more than nine months—for even a large business—until your first go-live means something is wrong with your plan.

You can expect an end-to-end solution for a medium complexity, lightly regulated business in manufacturing, retail or professional services with 100 to 200 employees based in a single country to take about three weeks for each phase: Design, Build & Test, Go-live Preparation and Support. So the go-live, which is at the start of the Support phase, is around ten weeks after you have completed your plans, signed your contracts and pulled the trigger to start.

Planting the seeds for future success

Sarah had been worrying about the negotiation phase for a while, but in the end it did not prove difficult to select a vendor. One of her options had a clear EULA, was prepared to negotiate on some points, but firm on others, and displayed a genuine excitement and enthusiasm about the business. The other wanted her to sign an

agreement that her lawyer was concerned about, wouldn't budge in anything at the negotiating table, and appeared stressed and tired.

Sarah was uncomfortable when Jasmine told her that she needed to include paid consultants from the software vendor or its local value-added reseller (VAR) in her project team. Jasmine explained that consultants or contractors have their roles. They aren't able to run Sarah's business, but they can tell them about best practices and what Beauty in Bloom's peers and competitors are doing. Jasmine shared that consultants are often young, energetic and intelligent people happy to work sixteen-hour days, but they also have their downsides. Jasmine was honest and explained to Sarah that the consultants:

- May not have her best interests at heart (if they did they would be applying for a permanent position in Beauty in Bloom);
- Cost anywhere from 40-150% more than their equivalent salary;
- Might not be keen on sharing their technical knowledge too much for fear of cutting the engagement short;
- Tend to have the habit of rushing off to the next big job that comes along even if yours is not quite yet finished. They are contractors, after all.

Sarah understood the pros and cons and realized that having one or two consultants in her team would be a bonus, even though she understood this did not let her core of Rob, Megan and Jennifer team off the hook.

Having selected her software tool and scoped out the project, Sarah sat down with Jasmine, two representatives—one from the software vendor and one from a local VAR—plus an SAP consultant college friend of her husband's who offered to show up in return for a beer night with her husband. She appointed Rob, her operations manager who also keeps track of her current orders, customers and marketing campaigns, as her full-time project lead. Rob joined the rest of the crowd for a half-day workshop to map out the implementation process. Sarah decided she wanted everyone to work as a team so she introduced the session with some ice-breakers, brought home-made brownies and deliberately held the meeting in a neutral location (a nice offsite hotel) rather than in her work-place or one of the vendor's offices.

After the ice-breaker Jasmine kicked off the meeting by explaining the objectives of the workshop which were to come up with a plan for the implementation that everyone could agree to and support, and left clear accountability for who was doing what.

The software vendor and VAR consultants each had their own plans already mapped out, which were slightly different. The software vendor was keen to downplay the software set-up tasks and wanted the whole thing done by Sarah's staff, so she was not 'wasting money on consultants' so to speak. The VAR consultant had been there before and was aware of the time spent testing and fixing issues, whether as a result of the software or how the customer used it, so he was more cautious. The SAP consultant turned out to have a similar plan, except that every week mapped out in the software vendor or VAR consultant's plan was factored in as a month in his plan!

After some back and forth everyone agreed to a project with two-weeks on-site for design, three weeks mostly remote for the build & test (including user acceptance test), two weeks on-site for final preparation and data load rehearsal, and two weeks of on-site post go-live support.

Sarah was a little nervous about letting some outside consultant influence how she was going to run her business but Jasmine convinced her that she and she alone needed to agree to everything before moving past the Design phase. At the same time, the SAP consultant was very firm in telling Sarah that she had to make decisions quickly, and not allow things to drag out since time is money. Everyone left the meeting excited for the journey ahead and having eaten rather too many brownies.

That evening her husband shared some negotiating techniques that he felt might be helpful. He also suggested forming a steering committee to see the implementation through and put together a convincing argument for its merits. Anything that the project team was struggling to deal with could be escalated to the steering committee for resolution and regular meetings would keep the whole process on track.

Before she started, Sarah collected a list of five similar companies, two from the software vendor, one from Jasmine and two from the SAP consultant (one for SAP's small business product Business By Design). She called them up, expecting only one or two to speak with her about their experiences. To her surprise, all five companies she contacted were more than delighted to talk to her and encouraged her to stop by. So she, Rob, Jasmine and Megan (who normally manages one of her florist shops) spent the next week on a road-trip up and down the East Coast.

Takeaways

Here are some of the pertinent points to take away from the project preparation stage.

1. Negotiating a win-win contract is always better than a win-lose because you want a good working relationship going forward.
2. Always have a lawyer check over the EULA before you sign, but only push back on what is critical.
3. Make sure everyone is aiming for the same goal, and link their rewards to the same measures so the team all pulls in the same direction.
4. Have at least one outside partner or consultant in your team and never more consultants on hand than your number of internal staff.
5. Plan on an initial implementation scope and scale slightly more ambitious than you are comfortable with. Plan between 10-100 users in each 'go-live', with a broad scope each time while avoiding partial retirement of any legacy systems. Let benefits drive your scoping decisions, implementing areas with the greatest and quickest benefits first.
6. Consider going live first with the group that is most "ready" or has leadership that is most open to the change.
7. Don't forget to include data cleanup and organizational readiness, change, training and communication in your plans.

Chapter 8 – Design: Teeing up for Long-term Success

You picked your team, signed contracts, made a business case, agreed to a plan, presented it all to your CEO (or if you are the CEO, to the board of directors) and everyone told you to “Go.” Now you can commence the implementation project for real, starting with defining how you want your business – processes, people, data and systems – to work after your go-live on the new software. This is called the Design project phase. It is normally organized in the form of “workshops” or for very small companies – “interviews”.

Treat the design phase as not only a chance to get input from all your key employees, but also a chance to communicate to your employees how the new system will work and start the process of getting their acceptance and buy-in to the new processes.

Workshops are a great way to do this, with between 2 and 6 people from your business meeting up with 2-3 members of your core implementation team. Typically you have one consultant and one of your own team members in each workshop. The format of each workshop can be refined as you go but generally each workshop is focused on the “to-be” or end-state vision covering the business process, any key requirements or pain-points with current systems, a demonstration of the new tool live in the workshop with Q&A, and a discussion on how the new system should operate without making any firm decisions at this stage. The trick is to encourage your key employees to keep an open mind, think about what is right for the company as a whole and to remove any shackles imposed by current systems, current data or current thinking. Leave the job of how to make the system behave as your key employees asked, or reporting the bad news that not everything requested is possible given the project’s timeline and budget - until after each workshop.

One question is what workshops to schedule. The scope that was prepared in the project planning phase is a great place to start. Break it down into sub-topics based on the potential workshop attendees, business process, legacy/existing system or some other criteria. Depending on project scope, aim for 5-15 sessions in total with some contingency for another 4-6 sessions for catch-up or more detailed solution explorations.

For commercial off-the-shelf packaged software you should configure or set-up the software in parallel during the workshops. In practice this won’t take long so don’t leave it to the consultants. Encourage your own team members to get their hands dirty in the software as soon as possible. One excellent IT project manager went round everyone’s desk each day to see what applications were open on each person’s computer. Team members with MS project, Excel and PowerPoint open did not last long on the project, while people with new system screens up in the first few days of the project fast became irreplaceable experts.

Try to keep the design process informal, as it is discussions inside the team that foster innovation. But make sure when decisions are taken they are written down and if necessary signed off by appropriate managers or process owners in your business. The deliverables at the end of the Design phase should be a report listing:

- All the business processes with “to-be” process maps for core processes,
- Departments and users with job re-alignment or training needs assessment for each person,
- All data that needs to be loaded with information on where it is now and what needs to be modified, added or cleaned up,

- The new activities or screens in the software tool that Beauty in Bloom is going to use with any changes to the software (or plug-ins) carefully documented, and
- Any special forms (printed outputs), reports or interfaces required.

Beauty by design

Immediately when they returned from the East Coast trip armed with insights from the experiences of other companies that had implemented similar systems, Beauty in Bloom's project team went into the Design phase. A schedule of workshops and interviews has been agreed for the Design and the deliverable at the end of this phase was to be a report listing: to-be processes, any changes in job roles or job descriptions they needed to write, a list of all key data and attributes to be migrated, and relevant system requirements and configuration settings that would be required. All three parties – the client, the software vendor and the consultant or VAR would have to sign-off on this design report and Jasmine would facilitate if there were any misunderstanding or disagreements.

Sarah was surprised to learn that of the consultants in the planning meeting, only one showed up at the workshops, plus one new person who appeared to have been briefed on the way in from their hotel breakfast. Jasmine was not put-out and they started with the scheduled two-hour interview/workshop formats with different groups from different departments talking about specific business processes, pain points with Beauty in Bloom's current tools, and potential risks and benefits.

Two workshops were organized AM/PM each day during the first week covering finance, receivables, client management, marketing, retail store management, inventory, planning, reporting, suppliers and field/mobile access. The second week was devoted to contingency planning and for writing up the report.

The first workshops were a little haphazard, but they eventually got into the swing of things and found that they had enough time in the two hours to cover more ground than the initial scope. Some of Sarah's employees thought the workshops were training sessions so they were a little disappointed not to see much of the software, but Sarah sent them all an email to explain why they were doing the workshops—no less than to define the future for the company. Attendance by her employees at the workshops shot way up after that memo.

Fortunately one of the two consultants was much more comfortable setting up the software than talking to business people so as it happened the software was quickly set-up and first the consultant, then Megan, and finally Rob, were keying in transactions and loading employees, users and product lines by the end of the first week. This helped so much to really understand what the system options were that they decided to hold four workshops on Monday and Tuesday the second week with the system being demonstrated by the second consultant in the actual meeting. The demonstration covered finance, accounts receivables, accounts payables, supply chain and spoilage, CRM and a redo of the field/mobile access workshop because a key employee was out of work the previous Friday and missed the first workshop.

On Wednesday, Thursday and Friday of this second week the two consultants worked feverishly on the design report with first Jasmine and then Rob, Megan and Sarah getting gradually sucked into what seemed like endless editing rounds, design and policy discussions, formatting and finally printing. Sarah half-jokingly told the consultants their flights would be cancelled if the report was not finished on time. They weren't sure if she was serious or not, because the final 75-page report appeared on her desk neatly printed and bound at 4:50 pm on Friday of the second week.

Takeaways

1. The project is a full-time endeavor even if your staff must perform their daily jobs. Try to postpone less critical day-to-day tasks and back-fill where possible, with temps or students added if needed.
2. Only do one of these complex projects at the same time.
3. Establishing good momentum is key at this early stage of the project. Be aggressive and plan workshops to be completed within a couple of weeks with a few days to configure and then demonstrate back the system.
4. Keep the core team focused in the new system, and do not give them time to revert to their comfort zones of preparing long-winded project plans or PowerPoint presentations.
5. Understand your benefits and let them inform where you spend the most effort during the design phase.
6. Aim for best-practice adoption for all non-core processes, and only request custom development for core processes that set you apart from your competitors.

Chapter 9 – Build & Testing: Guiding the Process

Having designed your end state system, the Build & Test phase is where you make sure you will get what you need when you "go-live". This means the right system, the right data, the right business processes and in the Go-live Preparation phase in chapter 10, the right (trained) users.

If you are implementing an off-the-shelf packaged software with no or minimal enhancements or data bridges from external systems (also called "interfaces") your focus will be on testing. But if you have many or complicated custom Reports, Interfaces, data Conversions or Enhancements (called collectively RICE in consultant-jargon) then the first order of business in this phase is to knuckle down and finish the build, before you start testing.

Since for most clients the focus is on testing, let's discuss that part first. There are many kinds of testing as follows:

- **Unit testing** is where you make sure one feature or screen performs in isolation as requested
- **System testing** is where you test an application or module with several interconnected moving parts
- End-to-end or **integration testing** normally covers a complete business scenario from end-to-end, like a "day in the life of" a customer order or similar
- With **volume testing** you load up the system with your maximum anticipated volume of concurrent users or data records imported to see how the system performs
- **Stress testing** is where you try to literally break the system in all sorts of ways, hardware, data and user related
- **User acceptance testing** is where real end users (not users in the full time core project team) get to play with the new solution and give feedback
- Finally, **regression testing** is where you go back and retest something which worked previously, to make sure it was not adversely impacted by some other change or fix.

Whole books have been written on testing methods but in practice - especially with a smaller company implementing off-the-shelf cloud software - you can dispense with stress or volume testing. The software vendor has had bigger clients than you before on their system, plus you probably can't upgrade their hardware anyway. Most core project teams do unit testing as part of the system setup, build and self-learning. If you have just one go-live then formal regression testing does not apply either, since you only need to test everything once.

So in practice you can focus on integration or end-to-end scenario testing, and user acceptance testing. You can even combine these two by organizing end-to-end testing into "cycles" and having real end users perform the latter cycles of this integration testing.

Once you have written down a series of one hundred or more test conditions and organized them into steps, scripts and scenarios, a sensible start-point for testing your new system is to plan on three rounds or cycles of integration testing as follows:

1. The first round covers "sunny day scenarios" (maybe 60% of your pre-planned test cases and conditions) and is typically executed by the core project team, with lots of consultant help. This first round uses synthetic data, i.e. data which might be loaded specially for the purpose of testing.

2. The second round covers all your test conditions and scenarios including many rainy day situations or "what can go wrong". It is still performed by the core project team but with limited consultant input; your own staff must run the tests instead. Also in this cycle you should test with a mix of synthetic and real data. Real data comes from an earlier data conversion rehearsal or copy of the new production data you are in process of manually uploading.
3. The third and last cycle doubles as user acceptance testing and must be driven by real end users who are not part of your core project team. While they still follow the same test scripts you used in the first two cycles, they are encouraged to consider - or even sign off - what will support their day to day jobs post go-live. Also you should ban synthetic data from this cycle; only real production data must be used because you are not just testing the system. You are testing the data, the users and any training, policy or procedure guides you wrote at the same time.

Don't forget to allow enough time in between each test cycle to fix bugs or issues that arose. Make sure that you formally track issues in some kind of system or spreadsheet.

For those of you unlucky enough to have specified and scoped a large number of customized reports, interfaces, data conversions or enhancements (RICE), you've got to add time to the Build & Test phase schedule to complete these essentially custom programming tasks before you can start testing and training in earnest. This is the number one graveyard of large-scale IT implementations and the reason why so many companies are turning away from custom-built systems to commercial off-the-shelf solutions in the first place. Please limit those enhancements and keep reports, forms, workflows and data uploads as simple as you can – as chapter 12 explains, you can always enhance things later.

Keep custom development Agile

If you do have to build many enhancements, interfaces, automated uploads or reports first then be sure to formally track progress and status of these to completion before starting integration testing.

The biggest problem with custom development is scope creep - either because users ask for more and more, or because you got the design or scope wrong in the first place. Unfortunately there is no easy answer to this. Try to hit a balance between agreeing and signing off prototypes, feature lists and screen mockups with users up front, vs. being "Agile".

There are good books on Agile and it is strongly recommended to get familiar with this new custom programming method if you have more than a couple of weeks custom development to do. Agile means that:

- You work in short "sprint cycles" with day by day accountability from programmers to what they will do that day, working to deliver user testable software at the end of each two week cycle
- Programmers do what they can in the time, but remain flexible to incorporate user feedback up to the end of development
- Your programmers work directly with educated business analysts and end users to iterate a working product until the user is satisfied it is good enough or a so-called "minimum viable product" that they can use
- Formal documentation is avoided in favor of mock-ups, feature lists and everyone literally talking while playing with the product as it is being built
- Rather than detailed up-front 'guestimates', you measure progress as you go and adjust your plan, scope and expectations accordingly.

Make sure everyone from the programmer to the end user is measured the same way and has the same definition of success. Too often the programmer is satisfied when the 40-page specification has been completed without running any real tests, while the end user never understood what the heck that 40-page document was about. They just want something they can do their jobs with (without having ever told the programmer what that actually means).

At the end of the day, if you have more than 4 weeks of custom development in your plan you either picked the wrong package in the first place, or you are really trying to build a custom solution rather than implement a commercially available package. Find a good book on Agile, double your estimates and hold your breath.

Can we build it? Can we test it? Sarah's experience

The next two weeks everything went quiet and Jasmine ended up insisting on daily status calls with the whole team to understand what had been done the day before, issues and decisions required, and plans for the coming day. Rob and Megan were busy analyzing their data and going back over the users' training needs and job descriptions. The remote working consultants seemed to be doing god-knows-what and Jasmine was at a bit of a loose end until Sarah suggested she spend three days working with the consultants in their offices learning more about the software. After this Jasmine came back talking like them.

Sarah focused on the cutover plans including what can go wrong and what can be printed ahead of time. She drafted out letters for suppliers, customers and employees covering each stage of the cutover. She had been given tons of useful tips on this from her reference visits and was determined to put them all into action.

One of the consultants recommended that Rob and Megan produce swim-lanes for their end-state business processes as a training and education tool. Swim-lanes are process flows organized into "lanes" by job role. Imagine taking a typical business process flow-chart and organizing all the activities or steps carried out by each job role into horizontal lanes, with connections going along a lane when the same role does two consecutive actions, and jumping lanes up or down when data is handed off between job roles. Swim-lanes are a handy tool for end users to visualize what steps and activities they need to carry out, understanding data hand-offs between roles. But Sarah realized that to build swim-lanes required a significant investment in time and money, somewhere in the region of 1-2 days effort per processes documented covered in the workshops, and 3-5 days for to-be business processes that were not documented. She reckoned that Beauty in Bloom had fifty or more processes running their business end-to-end, and she simply did not have the patience or deep enough pockets to see this one through.

For larger companies with higher staff turn-over, swim-lanes are a great tool to bring new people quickly up to speed. Since Beauty in Bloom was a smaller business with more day-to-day variation in their business processes, Sarah put it on her to-do list for a later project. It was not worth the time and money for the initial implementation.

After two weeks one of the consultants came back on site to run "integration testing" which meant getting Rob, Megan and now also Jennifer roped in to transact a day in the life of a wedding, a walk-in retail customer, a vendor shipment of flowers, including receipt and storage, finding a new client, invoicing and collecting money, and end of month financial and operational reports.

The end-to-end scenario testing started off terribly with one issue after another being quickly fixed by the consultant either with configuration or code changes. In fact they seemed to spend so many hours each evening on the phone to their development team back home Sarah didn't understand how the earlier clients managed to go-live at all. But things got much smoother by the end of the week and Jasmine was pleasantly

surprised to see some enhancements that the vendor had initially pushed back on during the sales cycle popping up in the product. Nevertheless, the VAR and vendor's project manager approached Sarah and Jasmine on Friday of the third week to request a one-week extension to the Build & Test phase, explaining it was 50% due to Rob and Megan not having properly prepared the data, but they would swallow their side of the labor cost increase because it was 50% due to the software issues encountered in the first "two days" of testing.

Jasmine advised Sarah to accept it and pay for the extra week's travel expenses. The second round of integration testing in week six of the project went much smoother. Jasmine was a tiger though, going through the list of forms and reports promised in the Design report and asking the consultants where they all were. She seemed to be coming up with requirements and reports from the Design document which Sarah, Rob and Megan had completely forgotten about.

Takeaways

Bear the following points in mind during your implementation, especially in the Build & Testing project phase:

1. Technology is only a part of your implementation. The softer issues such as organization and process change, business readiness, and end user training and acceptance are harder to do and more critical to the success of your project than the technology solution you give them.
2. Schedule 1-2 steering committee meetings per month to escalate and resolve issues and policy decisions. Make sure that these meetings are attended by your entire senior management team plus representatives from vendors. Have a formal agenda and agreed risk and issue management procedures.
3. Avoid custom development even if it appears to fill gaps in the software. Never modify the code delivered by the vendor and don't expect the product to work out of the box in all scenarios.
4. Testing is there to find (and fix) issues. Organize your integration testing into three or four cycles progressively dealing with rainy-day scenarios, using real vs. synthetic test data and involving end users from your business who are not in the core project team (i.e. have little exposure to the software).
5. Treat the last round(s) of integration testing as a combination of user acceptance testing and training.
6. Don't underestimate the effort required to find, clean, prepare and then upload your data into the new system; even for small organizations the volume of data required can be staggering.

Chapter 10 – Go-live Preparation: Steady, Ready and Trained

Following build and test, the last stage before going live is typically user acceptance testing and training, also called "Go-live Preparation". In this phase you make sure everyone and everything is ready for the change, data migration works and end users are trained. Sometimes the third or fourth round of integration or user acceptance testing is part of this final preparation phase.

Most clients treat user acceptance testing as a handy training tool, in terms of showing users participating in acceptance test cycles “how” they can complete certain transactions. For an end-to-end system you should also consider other kinds of user role-based education and training, selecting from any of the following options depending on how many users you have in each role and the complexity of the system and their learning requirements:

- **System awareness** and navigation is normally training you can get from your software vendor as it needs little tailoring for your company. It can be self-learning (or computer-based training) and can range from a basic introduction to a detailed explanation of common features such as navigation, collaboration, personalization and so on. Everyday employee actions such as putting in timesheets or clocking attendance or updating your own HR profile can also be self-taught in large companies, where classroom style training of all employees is impractical.
- **User education** should never be overlooked and really addresses “what” users need to do in their new jobs and to some extent “why”. It does not need to go into the system at all, rather it is more focused on new processes or swim-lanes (explained in chapter 9), changes in their job roles and data hand-offs from other people in your company. This can be either data they need to get from others, or people downstream who are reliant on their inputs and updates. Many experiments have shown user satisfaction and data accuracy shoot way up when education is included for an IT system go-live.
- Formal **classroom training** by job role is expensive and normally only worthwhile if you are a large company, i.e. if there are six or more employees that have the same job role who can get to the same place for one training class. Many vendors offering standardized or customized end user role-based training and while getting outside help to make the materials look professional is wise, it is strongly recommended to try to get a super-user or core project team member on your workforce actually delivering the training. They know your company and can answer questions being raised by users, plus if you picked your best and brightest for the team then they will be respected by the folks you are training.
- Lastly – and this is kind of the catch-all if none of the above apply – you can run **informal training workshops**. They are like classroom training except you do less preparation (no slides, only prepare system demos) and the training is more along the lines of a Q&A. Here you probably need two trainers, one of your own employees and one consultant to deal with system issues and demos your own employee had not anticipated. Informal training is ideal for smaller companies where one training session might cover different users in different roles at the same time. You spend more time and money on the delivery, and less time in preparing materials ahead of time.

A vital question asked about end user training is when to deliver it. Professional training managers might insist on three months to prepare and four weeks to deliver the training

for a medium sized company – something that is completely impractical given that you will be still tweaking and enhancing the system a few weeks before go-live. Few users really care that much about the new system until their old system is about to be taken away.

A sensible rule of thumb for an end-to-end system is to estimate four to eight hours of training per “professional user” (one hour or less for occasional users like every-day employees) and fifteen hours of preparation for one hour of delivery.

Squeeze down the preparation time to the last four or five weeks before go-live so you have a more stable system to work with, and split the actual training delivery so it runs for one to two weeks prior to the go-live and one week after the go-live. That way it will be fresh in your users' memory when the panic starts. And don't expect them to retain everything; give out cheat sheets and stage people who know how to operate the system close-by to support them when the first questions come up.

The Countdown for Beauty in Bloom

Go-live Preparation looked to Sarah like more of the same testing cycles as the Build & Test project phase, but with all her employees as users running tests, not just the core team. They came in for 1-2 hours acceptance testing / training combined each day before starting on their day jobs. This consisted of a new consultant (how they learned about her company was over another breakfast meeting Sarah imagined) explaining their business processes and giving a quick demo and then some hands-on exercises based on the test or scenario scripts produced by Rob, Megan and Jasmine the week before.

Things seemed to be going well with not many issues in the software being hit, and most discussions about how to use the system to run the business. Sarah was pleasantly surprised how much of Beauty in Bloom's data seemed to be used in the training. Jasmine explained that it is far better to do training and acceptance testing on realistic data for two reasons: the users understand it better; and if there are issues caused by the way the data was loaded these come out before the go-live.

Sarah was a little perturbed that the training seemed to concentrate exclusively on “sunny-day scenarios”.

During the second week of training/testing Sarah, Jasmine, the lead consultant and the vendor's project manager met three times to run through the cutover plan. Sarah showed them her contingency plans and letters, printed forms and paper back-ups. All the consultants were mightily impressed, and one even said that she could run her business without any Internet for six months on that lot. Rob and Megan (who seemed to now know as much about the software as Jasmine did earlier) ducked out of the second week of training and focused on the data loads, manually updating anything that needed to be modified and going through two rehearsals of the automated or scripted data uploaded from the various Excel files they had pulled together. The first rehearsal took all day and the second one took just over two hours.

After eight weeks they were ready for the go-live. The next chapter will explain what happened when they flicked the switch...

Takeaways

As you complete final preparations for your go-live, bear in mind the following:

1. Triple-check your data, and implement counts to compare data between old vs. new systems before going live (e.g., inventory quantity on hand, sum of standard prices, etc.). Allow conversion programs to be run more than once without corrupting or duplicating data.
2. Don't start preparing training materials too early when the system is still being changed. Focus training on the end users' job roles and how to perform transactions in the system. Try to avoid building fancy slides, and focus on exercises instead.
3. Expect your end users to only want to learn how to use the new system after the go-live. Be on hand when they need your team to support, guide and coach after the go-live.

Chapter 11 – Go-live & Support: Weathering the Storm

Going live with your new system is the process of converting from your old ways - old processes, procedures, systems, data and even job roles to new ways, systems and processes. It typically involves user training and education, final data migration, last minute tweaks and fixes, turning on the new system and then ramping up business operations on the new system, all the while measuring business performance in the short term and business benefits in the longer term.

You can practice parachuting in a hangar all you like, but one day you are still going to be up at 5,000 feet in a plane standing near the door scared out of your wits before your first solo jump. And at the end of the day you just got to have faith and jump.

Going live with any new system that affects a significant part of your business is a bit like that. You can do as many dry runs and what-ifs as you like. You, as the boss of your company or of the migration program, are still going to have to stick your neck out and flip the switch. You've got to turn off the old system, or at least stop updating it. Dual running both old and new systems in parallel is possible with enough temporary labor to do all the extra data input, but you've got to be super disciplined to wean people off their old systems on a strict timeline. You might as well do it in one go, and keep open access to the old system in view-only mode for a couple of weeks, but turn off updates to the old system at the same moment as flipping the switch to using the new system.

A successful go-live is a boring go-live. No fire-fighting, no heroes working to 3 a.m. to manually ship customer orders, no stories for your grand-children about working three days and nights straight and then crashing your car on the way home, no screaming users and no consultants temporarily running your production lines or warehouses. A successful go-live is pretty much an anti-climax after the hectic project implementation schedule. And above all, a successful go-live is one where you first measure business performance and then you track business benefits, to find out if the new system, processes and training have really lived up to their promises.

To make your go-live as dull as possible, you first need to understand, and then eliminate, all causes for "excitement." These are typically system glitches, bad data, Murphy's law, lack of user knowledge, work-arounds, blame-games, fat, dumb and happy syndrome and going live smoothly but not realizing any benefits. These pitfalls and how to avoid them are explained below.

System Glitches

It is amazing how many things escape the various test cycles, often because your consultants persuaded you to focus testing on "sunny day scenarios" (so they could get signed off for payment) while your actual users had a string of rainy days after you went live. Whole scenarios can be missed because none of the right people were asked, forcing your team to make it up on the fly. To avoid this, involve everyone you possibly can in user acceptance testing, making sure anyone affected understands that it is their responsibility to call out before the go-live if they have any doubts on how to perform their jobs with the new system. Above all, include rainy day scenarios in your testing. Rainy day scenarios include stuff like data which is missing or badly set up, urgent orders from new suppliers or customers not set up in your master file, things that don't match, user error, repairs or failures, etc.

Bad Data

One of the biggest causes of post go-live stress is data that is either poorly converted, badly setup or missing. Check all your master data at least three times with three different people including the person who is mostly using it. Don't assume anything. For

data uploaded using a program, check it after every upload, do a detailed sample analysis of 5-10 individual records and count the number of records (or some other balance such as inventory numbers) on both old and new systems to compare. When it comes to data, 95% accuracy is not good enough, because one bad record can bring a business to its knees. One ERP go-live had the police out directing traffic around a string of 30 trucks backed up from the warehouse receiving-bay to the previous motorway exit, because the standard weight for a single light bulb was wrongly converted as 1 ton. None of the warehouse racks would accept the weight of a whole pallet of light bulbs, preventing the first truck from being unloaded for 45 minutes.

Murphy's Law

New system go-lives are prime targets for Murphy to strike. You can't be sure when or how, but you know he will strike at some point. Make sure that your team is resilient, dedicated, energetic and working together for a single cause: a successful go-live with minimal impact on your business. While people will work hard in the run-up to the go-live, try to avoid many sleepless nights as lack of sleep not only reduces productivity; it leads to basic human errors being made exactly when you least want them.

Lack of user knowledge of what to do

Most users either skip or sit through training for the new system in a daze because they have more important things to do. They then go into a blind panic when the old system is turned off. You can shout and scream about people attending training, or give tests and other incentives like free food or performance incentives - and these all help. But at the end of the day your average user only really cares how to operate the new system once access to the old system is taken away. This is when they have to learn the new system to do their jobs, so this is when you need people on hand to help, train and guide them.

It can be as simple as observing the ten-to-one rule of thumb: one person in the user community who understands the new system well for every 10 new users. Or you can be extreme and run most of your end user training after the go-live. Either way, you've got to teach your new users on their terms, and go back a few days and weeks after the go-live to get feedback, reinforce the training, and find out what is not working and fix it.

Work-Arounds

Work-arounds are good to start with because they demonstrate that your users are finding creative ways to get their jobs done. But ultimately work-arounds will kill your new system because they almost invariably leave data in the system which is not updated, and therefore which cannot be used by the next group or department reliant upon it. This leads to another work-around or worse, a way for users to communicate data updates entirely outside the system. It is a vicious circle, as users become wedded to these temporary spreadsheets and the data in your brand new system becomes increasingly useless.

Help your users create temporary work-arounds that do not contradict or undermine the new system in the first place. Then fix the system quickly so the work-around is no longer required. Finally, once fixes are in place, convince upper management to sabotage work-arounds by insisting that their team report in meetings etc. using live system data, not from spreadsheets.

Blame Game

Any large system will cause users some stress and many, even highly successful companies, experience a slight dip in business performance at the time of a major new system go-live. But some companies blame their new system for months and months after go-live. If the system is that bad then rip it out, go back to your old system and call your lawyers to start suing your consultants and software vendors. However, even if the system is running smoothly, employees can whine about the change for months. It

might be legit at the start; but after six months, if the complaints persist even after a decent work-around is in place, people may be using the system change as a convenient excuse for not doing their jobs.

The way to stop complaints is to have objective measures. Measure at least 10 key business "operational" performance indicators such as: number of warehouse shipments, on-time delivery rate, invoicing errors, days of WIP, orders input and so on. Measure them before and after the go-live and make sure that top management buys into the whole exercise—from how these performance indicators are measured, to how long the performance dips should last after the go-live.

What you will find is that unchecked, the number and volume of complaints and excuses remains constant while actual underlying business performance gets better and better over time. The reason you need buy-in from top management is that with their support, the complainers and excuse-makers may have to be told eventually to shut-up or ship-out.

Fat, Dumb and Happy Syndrome

With this condition your team and perhaps top management are congratulating each other on the successful go-live. Meanwhile your users are struggling under an increasingly unmanageable workload to do their jobs, because of basic flaws in the new system. If this sounds like you, then you have great employees to cherish, but you misguided the implementation. Listen to your users, learn about their issues and fix them. Find out their work-arounds and get them baked properly into the new system.

Don't assume that no news is good news, and don't expect your average user to log a ticket or call a help-desk every time they get frustrated using the new system. Sometimes you have to reach out for feedback.

Lack of Benefits

Most implementation teams think the purpose of the implementation project is to go live smoothly to the new system. The real purpose of the project is to bring about a far greater benefit to the company than the project's cost. You can't bring about any benefits unless you measure them. This can be as simple as writing down your top five key benefits and business goals from the system and meeting with your management team every three months to assess if these benefits have been achieved. Or it can be as complex as setting up a benefits register with tangible (i.e. monetized) and intangible benefits mapped over a three-year timeline, signed off by your line managers and baked into next year's budgets, and then measured automatically using the new "business intelligence" software you implemented as part of your new solution. Either way, you need to measure benefits on an ongoing basis. This also makes for a handy tool to assess the impact of future process or system improvements.

Build benefits into your project methodology from the get-go when setting out your goals, when deciding on project scope and roll-out sequence, when deciding where to invest more in your new system, for getting user buy-in and for measuring post go-live business performance. Measuring and realizing benefits from a major IT system implementation is worthy of a whole book on its own.

Bloom or boom! Beauty in Bloom's Go-live

Sarah felt good about going live. User acceptance testing was running on schedule and although some users seemed to be treating it as a mixture of training and a forum to raise new requirements, they generally seemed happy and agreed that they could do their jobs using the new system. Data conversions were running approximately on schedule, though everything seemed to take 30% longer than planned and Sarah was worried about the consultant reporting a 4% "error rate" in the customer load file. When she asked about it she was told it was due to duplicate records, without an explanation

as to which of the duplicates was the correct record, or why the consultant said that Megan had duplicates in the file when she had assured Sarah the customer file was cleaned and checked several weeks back.

The software vendor executive involved had called Sarah and assured her of their support for the go-live, which made her feel good. At the same time he had told her to expect a few bumps, make sure her contingency plans were in place and get involved personally in every aspect of the go-live. This worried her because she thought it might be a pre-cursor to the software vendor running out on her. But her husband assured her that this was quite normal “cover your rear” vendor behavior and getting involved in a more hands-on capacity would stop her feeling like a spare part when she was around the project team. Sarah quickly agreed – her role of late had been relegated to finding new take-out menus to avoid an exclusive diet of pizzas.

She was amazed how hard and happily her team worked. Rob, Megan and the consultants merrily swapped banter every night as they toiled day after day until 10 or 11 pm, so much so that she wondered if they really had a life outside work, and what they would do to fill the time after the go-live. She felt confident that her team could take on whatever was thrown at them come the go-live.

So on Thursday evening they had their final go/no-go meeting for that weekend – a quiet weekend in Beauty in Bloom’s normal calendar. Everything was in place bar some data conversions and a less than perfect end user training attendance record. Letters were being sent to all employees, suppliers and current customers explaining the transition, and all key business documents were printed out on paper just in case nothing worked on Monday. Even a power-cut would not stop her now. So Sarah agreed to flip the switch, stop updating the old systems (she still gave her employees four more weeks access to old system data as a back-up), and run the final data conversions. Now back to the pizza menu for the team’s meals over the weekend...

Come Monday it was eerily quiet. The consultants were watching who logged in (not many to start with) and waiting for tickets to be raised. Rob, Megan, Jasmine and Jennifer were spread out in each site with the end users as they came into work. Everything seemed to go as a normal day and it was so quiet Sarah was actually worried until she realized that on Mondays people normally don’t update much in the system.

Tuesday was a completely different story. First one problem and then another seemed to come in. Some users had forgotten their passwords and the Internet line to one of her remote stores seemed to be playing up. It was nothing to do with the new system but because that store only used the Internet for the new system the project team took the heat. Most of Tuesday seemed like an exercise in user training over the phone, with the consultants jumping between end users coming into the project room to demand answers and Rob, Megan and Jennifer on the phone with technical user questions they could not answer. Sarah was both pleased to have the consultants on-site and worried that she could not get by without them. It seemed like issues were getting solved and by Tuesday afternoon the consultants even had time to start logging issues back into the centralized tracking spreadsheet that Jasmine had set up. Not only that, but they seemed to have time to make some tweaks to the software through the software vendor’s development team; seemingly directly in production. When Sarah questioned their change control procedures the lead consultant assured Sarah that these “urgent fixes” were following their normal QA and internal testing procedures.

Wednesday morning Sarah received a phone call from the manager of the remote store with Internet issues the day before complaining about the new system and stating that the wrong flowers were delivered for a wedding as a result. Sarah kept her cool and explained that all customer order forms were printed on paper and distributed to each shop specifically for this eventuality, and he had no excuse getting the order wrong even if he had to work completely from the paper copy. There was silence on the other

end of the phone. Nevertheless Sarah realized that there was a potential change management issue with this store and she should give it special attention, at the same time making sure no blame-games were going on. She resolved to send the summary status of all tickets and business performance metrics to her entire management team with a “system implementation update” every other day for the next week or two so that everyone was on the same page.

Things quieted down again after that so much so that she wondered if her users had gone back to the old system. Business performance metrics continued to trend upwards, system usage and transaction volumes remained steady and updates in the old system were few and far between (for technical reasons she could report on, but not stop, updates in the old system). After one week the consultants were leaving and she decided to have an education workshop at each site by way of getting feedback, boosting confidence and refreshing the end user training. Attendance at these workshops was phenomenal – people she didn’t even know she employed showed up – and everyone said it was valuable.

These workshops resulted in a list of 15 more issues (around half of which were really long-term enhancements), which gave Rob and Megan something to work on next week. That next week she started insisting on reviewing actual system dashboards and reports in her order status, financial performance and pipeline weekly meetings - not spreadsheets with data massaged in favor of the presenter. This shocked her managers, especially when she told them how to navigate to the dashboards concerned, and she could see the penny dropping as they realized the new system was not going away anytime soon and they needed to get used to it.

She was amazed how much Rob and especially Megan understood the entire business. If she ever wanted to retire she knew that she had a capable successor in Megan and resolved to promote her as a result of the program and as a way to keep her. Rob was also great but as he had become an IT professional she knew he would ultimately find supporting Beauty in Bloom’s smooth-running IT systems unfulfilling and perhaps leave to become a consultant or contractor. Having two more people with such intimate and end-to-end knowledge of her business was an unexpected bonus. Now she had to get back to work and focus on other improvement projects such as new store displays, an update to her brand/logo and a new pricing strategy to be more aggressive in locations with stronger competitors.

Takeaways

1. Assume anything that can go wrong will go wrong. Be a scout, not a Murphy—prepare!
2. Have a resilient project team working toward a common goal and expect a few last-minute glitches and late nights. The less excitement around the actual go-live, the better.
3. Be part of building any temporary work-arounds as required. Make sure they don’t undermine or invalidate data in the new system.
4. Measure business operational performance before and after the typical go-live “dip” to replace whining and speculation from users with facts and figures about the health of your company.
5. List, agree and above all, measure benefits of the new system on your company. After all, that is why you did this in the first place, right?

Chapter 12 – The New Solution: Continuous Improvements

You have made it. You've analyzed the needs of your business, assessed systems in the marketplace, chosen your solution, collaborated on a detailed plan, and seen it through to live status. You and your business are now in an excellent position to reap the benefits. But before you put your feet up, remember that going live is just the first step in a new journey to the ever-elusive isle of business perfection.

Sadly, no system implementation is ever flawless. Corners are cut, scope is trimmed back, and something always gets swept under the rug. Your first task after the go-live is to iron out the creases. Just because the budget for implementation has been spent, don't think that you're done investing in the new system. Post go-live is your chance to really get a return on your investment because you are in control, and you have the expertise and power to make changes that will really leverage your new system.

Now that you have access to real-time data and a solid structure of accountability, you can make serious process improvements. The ability to access a single, authoritative version of the truth about your business is a powerful tool. Don't squander it! Dig into that data, mine those insights, and use anyone remaining in your project team to realize continuous improvements going forward.

Japanese manufacturers were able to take the world by storm because of their philosophy of continuously changing for the better. The simple idea at the core of their success was to empower all workers to engage in the improvement of processes throughout the company, at every level. No one knows the job or how to cut wastage better than the people doing it. Encourage everyone in your company to suggest improvements and reward them when they do. Don't rely on "process owners," external consultants, or business analysts to steer your efforts. Everyone at your company can improve things and they should be constantly striving to do so.

Supporting your new system effectively is also vital, especially in the first few days and weeks after implementation when you are still working out the kinks. Effective support will rely on a solid partnership between you and the vendor and it should be based on open and honest communication. You will want to outsource as much support as possible to the vendor - provided it is included in the subscription fees in your contract, but be wary of any detrimental impact on the level of service you are providing for your end users.

There are essentially three levels of support:

- **First level** – user knowledge gap, maybe someone upstream did something wrong,
- **Second level** – a real system issue potentially fixed through configuration or data tweaks, or a modification to a process, procedure or training documents,
- **Third level** – an actual bug or enhancement request.

It is best to keep first level support in-house. You need to maintain a good working knowledge of the system and ensure a positive experience in terms of your users. Second level support will often require the involvement of your vendor. Third level support will definitely have to be handled by your software vendor.

With regard to software bugs and enhancement or change requests, it is important that you understand the distinction. You have chosen the software based on its ability to meet your requirements, but it is unlikely to tick every single box. If a piece of functionality within the software does not work as advertised (as described in your

software vendor's documentation) then you may have discovered a bug and you have every right to request that the vendor fixes it.

If you find that some functionality doesn't work quite the way you expected, or that a feature you want is absent, you can consider requesting it to be added. But this is an enhancement that the developer is not necessarily under any obligation to provide. Don't assume that a feature or functionality that seems like an obvious requirement to you will appear that way to the software vendor.

The smart way to back up change requests and get the enhancements that you want is to harness the power of the user community. If you engage with other customers using the same software and you are able to find common requirements, then it will be easier to persuade the vendor to work on them. Always ensure that you properly explain what you need with example data scenarios; the finer the detail, the better. This can work out as a win-win because the software vendor is effectively developing for your needs, but they can be persuaded to do it for nothing if it improves their overall product and they retain the intellectual property rights. The trick is to convince them that your enhancement request adds value to their product.

A vital tool in your efforts to ensure good support is the SLA (Service Level Agreement). You should establish SLAs with all of your external vendors that define things like the time it will take to respond to a bug report, or to fix an issue with the system. A clear definition of the level of service establishes expectations and can help you to avoid a lot of potential turmoil. Even your internal team should establish an SLA as a contract with your end users. It's always better when everyone knows what to expect and what is expected of them.

Upgrades to your software can be extremely problematic. If you have physically installed software on your network and you've enhanced it since the original implementation, then updates will be disruptive, time-consuming, and expensive. If you opted for a cloud solution then you can forget about this headache. Other than installed apps or plug-ins like mobile clients, upgrades are a thing of the past. This ability to continuously update software on a rolling basis is one of the strengths of a cloud system. It saves you money because you don't need to upgrade everything every two years and you are always using the latest and greatest version of the software.

Be careful not to hire too many programmers, as programming is probably not your core business. Programmers can be hard to manage, hard to retain, and tend to write a ton of custom code you can't maintain after they are gone.

Your new system should facilitate, not hinder, acquisitions. If you've chosen a good system and implemented it properly then it's liable to be better than whatever the company you are merging with is using, even if they are bigger. It should also only take around a third of the time and a fraction of the budget of your original implementation to deploy the same solution to a similar-sized acquisition or merging company. Properly implemented packaged end-to-end software can be an enormous help with acquisitions as it allows you to deploy your processes, policies and procedures to the acquired business with relatively little effort.

Similarly, if you are considering outsourcing or centralizing certain business processes, such as payables invoice matching, warehouse management, or chasing of customer payments, your end-to-end software is a fantastic tool to facilitate this. Don't be taken in by external business process outsourcing vendors who claim to have a better system. You have a brand new system set up for your business available over the Internet, so they can use it directly and everyone can have one version of the truth.

Don't forget to keep measuring those benefits. You probably have implemented analytical tools and the ability to draw on data to create detailed reports on your business. You can assess the impact of every change and enhancement that takes place in order to ensure that you're always moving in the right direction.

One final point is that wherever your business goes, it is vital to keep a handle on your own data. You should always be able to quickly and easily extract the data you need to get an overall snapshot of where your business is. If a better tool comes along or the supplier landscape changes you should be ready to jump, without locking your data on some server owned by your old vendor.

Bedding in at Beauty in Bloom

Sarah could hardly believe it when the go-live process came to an end. The new system was up and running, the staff seemed to have soaked up the training, and things appeared to be running smoothly. The buzz of daily meetings, outside consultants, and vendor visits had died down. She sat in her office with a self-satisfied grin and decided to phone her husband so he could share in her triumphant moment.

When she told him she could finally put her feet up and relax he burst out laughing. When the mirth subsided he explained that there would be bugs and errors, there will be confusion and support requests, and she still needed to measure the effectiveness of the new system and identify ways to improve it. He was speaking from experience, but Sarah couldn't suppress the thought that perhaps she had done a better job of ensuring a smooth go-live and Beauty in Bloom would never come up against the same problems. By the end of the week, she felt differently.

Megan phoned her just before lunch complaining about a major shortfall in stock. After Sarah checked the system and spoke to her Californian suppliers she realized that an error had occurred somewhere along the line. The stock was there, but it had disappeared from the system. A couple of hours of investigation and a long call with the vendor led to the revelation that no expiration date had been set on the incoming flowers and so the missing stock had been assigned wastage status, meaning that it didn't appear as available. This led to an argument about the functionality for assigning status and expiry dates to stock and the lack of an alert system about batches exhausting their shelf-life. Sarah felt this was a bug in the system, but the software vendor described it as a change request. When she pushed for an immediate resolution the vendor referred back to the SLA, which clearly stated, "We will respond to every enhancement request within four weeks, but make no guarantees what will be developed."

That weekend Sarah met Jasmine for dinner. It was to be a celebratory reward and post mortem of the process, but Sarah was preoccupied with the teething troubles she had encountered earlier in the week. They talked it out and Jasmine reassured her, but also pointed out that the go-live was really the first step in improving the business. They discussed ways in which she might continue to improve both the system and the business by involving the staff. Continuous improvements can take many forms and together they came up with four kinds of improvements they could plan over the next few years:

- Purely procedural improvements related to process, policy, training, or education,
- Enhancing the software tools you with new features or a module from the vendor,
- Slim bolt-on applications or getting some custom built scripts developed around making core processes more effective, and
- Totally new deployments for a new site, an acquisition, or a major new module.

Jasmine reminded Sarah to keep a tight focus on the changes and their impact, and to measure the success of each new enhancement by analyzing the wealth of data that was now at her fingertips. By the end of the meal Sarah's optimism had come back.

She loved a challenge and this whole process had been about striving to make the business as strong as possible. She had no desire to rest on her laurels.

The next Monday she called a staff meeting at each of the branches and praised the staff for their handling of the implementation. She also called for suggestions and vigilance from them in helping the new system bed in and to ensure that every aspect of Beauty in Bloom's processes was as refined and efficient as possible. She continued to insist on reports in all meetings to be based on live system data, banishing PowerPoint or Excel from being used for any status reports on the business.

She called another CEO in the late afternoon, the catering company executive that she had spoken to before about implementing this new software system. She thought the wastage issue could apply to their business as well and she felt a timely call requesting the same functionality could be just the thing to persuade the vendor that it was a worthwhile enhancement.

The rest of the week was spent establishing a base line for her business by generating reports from the new system and reviewing some of the suggestions that had come in from her staff. A few more minor issues cropped up, but they were dealt with efficiently. By the following Monday her vendor had already rolled out a basic alert function to highlight potential wastage, which really pleased Sarah and the catering company CEO. Sarah was busily working on a new list of enhancement requests.

Much like life, Sarah had realized that improving her business was all about the journey, not the destination.

Takeaways

Here are a few points worth remembering in the post go-live phase for your business.

1. Don't stop working to improve the system. You will identify potential improvements worth making and uncover bugs or shortfalls. It takes time and effort to polish the system after implementation.
2. Include your staff in the mission to improve your business and listen to suggestions from everyone.
3. Ensure that you have a solid SLA in place and that your support strategy for the new system is clear and functional.
4. Enlist the help of the wider user community to persuade your vendor to develop the enhancements you want.
5. Always maintain a firm grip of your data and use it to measure the success of any changes that you make. Improvements to your business must be quantifiable.

Chapter 13 – New Rules, a Summary & Why iBE.net

Congratulations on reading chapters 1-12 of this eBook (or being clever enough to skip to the Summary). You are either about to embark on selecting and implementing a new business management solution to run your business or looking at tips and tricks to gain expertise and confidence for when you do undertake this journey. Perhaps you know about problems other people have experienced with large-scale IT system implementations or read about some in this book.

Rest assured no-one who implemented a good off-the-shelf end-to-end system, especially a cloud-based one with modern architecture and an intuitive user interface, ever regretted what they had done.

Once you have gone through the process of finding, selecting, negotiating, designing, building, testing, training, going live and then improving your business and its tools, you will be in a really good place. You will have more insight into your business than ever before and everyone will be working from one set of data or "one version of the truth". Improvements to your business - whether through tweaks to processes, staff / training, system modules or other strategies - are easier to do than ever now that you have a common platform with all your business data in one place. As chapter 11 explains the go-live is the start of a long journey of continuous improvements to your business.

As a handy reference the key messages from each chapter are included below:

1. Who are you? Understand your business:
 - a. Work out what business model best fits your company. You need to focus first on either price, technology or service (not all three at once), so choose one to lead with.
 - b. Work out what is your core market, where your company is a major player. It is a much smaller market than you think. It is better to be a big fish in a small pond so you can focus your messages and marketing investments appropriately.
 - c. What three to six measures really define your business, things that mean the difference between success and failure? Know these measures by heart and benchmark them against your competitors so you can measure them objectively every week.
 - d. Never lose sight of your target customers, your core market and your competitors. Keep one eye on your environment - including regulatory, technology and organizational limits and opportunities - and the other eye on your customers and competitors.
2. Define your new system requirements:
 - a. Keep things simple. Aim for 25-75 core IT system requirements so you can separate the sheep from the wolves without muddying the waters.
 - b. Consult with your staff, customers and key suppliers to gain valuable business inputs from other viewpoints. Don't rely on a consultant to do all the work for you, but do get a second opinion from an external partner without a vested interest in any particular software option.
 - c. Create a simple scoring system and get consensus from your staff on which ones are critical vs. wish-list requirements. Try to keep requirements at a similar level of granularity, avoid too much detail or too broad or sweeping requirements going into the same list.
 - d. If you plan on getting vendors to respond to requirements questionnaires then throw in a couple of "spoiler requirements" to test the integrity and confidence of your potential software partners. Spoiler requirements are ones where a positive answer is wrong. You need vendors you can trust to tell you what is in your best interest, weeding out vendors who simply tell you what they think you want to hear.

- e. Have a staff member or intern have a go at responding to vendor requirements themselves based on on-line research into customer feedback, forums, talking to other users of the software, and so on. If nothing else it is a handy tool for gauging the honesty of the vendor responses.
3. What kind of system do you want? Build a roadmap for your businesses' IT needs:
 - a. Consider the pros and cons of a cloud vs. on-premise installed solution, taking into account future expansion, on-going maintenance and upgrade costs. Most smaller businesses are moving towards cloud-based solutions in droves, for good reason.
 - b. Understand your options around social collaboration - document sharing, comments, chat, Q&A, knowledge-bases and so on. Consider how social collaboration benefits your business and define your scope and requirements accordingly, rather than a knee-jerk response to a "cool demo" or on-line article raving about someone's new chat, message or video sharing tool.
 - c. Do you need an end-to-end solution or can you get by with point solutions as a stepping stone towards an end-to-end suite down the line? You must choose your vendors carefully and think about your expectations for them now and in the future.
 - d. Think about how you want to leverage mobile technology for use in your business, remembering to plan for proper integration and security of any mobile devices your employees use. It's always worth asking whether it will enable your staff to perform at a higher level; don't buy a batch of tablets simply because they are in vogue right now.
 - e. Consider your preferred management style and the importance of cash flow in your business before deciding on which vendors to partner with. The contract you put in place; whether payment is up-front or over time, capitalized or expenses, it has to work for you.
 4. What systems are available? There are plenty of fish in the sea:
 - a. Create an initial short-list of potential vendors from the wealth of software solution options out there. Use multiple sources - online and offline - to look for solutions; don't just rely on a Google search or what a friend advised you to look at. The more sources you use to look for potential suppliers the better your final selection will be.
 - b. Get to grips with some basic industry jargon to help you gain a better understanding of what is being offered.
 - c. Refer back to your requirements and your IT roadmap defined above when looking at potential solution options. Research the systems that might tick all of your boxes by checking out independent software comparisons, white papers and customer reviews.
 - d. Always seek out independent opinions that are divorced from vendor vested interests. Analyst reports and independent comparison websites can really help with this. Don't be fooled by so-called independent reports and blogs written or sponsored by one or other software vendor.
 - e. Set up an email address specifically for the research and demo phase of your journey so it doesn't distract you when you need to focus on the day-to-day running of your business.
 - f. Aim for 15-25 end-to-end solutions or 10-20 point solutions (per module or feature) in this initial round. There is plenty of time for down-selection later.
 5. Pruning back potential partners, three is the magic number:
 - a. Always dictate your own criteria and weighting system to assess potential partners, and listen to their advice but don't be tempted to adopt their criteria.
 - b. Independent advice is invaluable, so seek it out from customer reviews and online feedback, third-party reports, independent consultants, and by talking to other companies that have used the same software. Always try to check for potential bias. Remember the more sources you gather information from, the more reliable it is.
 - c. Empower your staff and enlist their help to narrow your selection. It gives you a big picture overview and it secures their buy-in.
 - d. Insist on hands-on testing time with the system where you are free to explore it for yourself. Don't settle for presentations, demonstrations, or guided tours.

- e. Don't allow vendors to influence the process by going behind you to other staff or stalling for time. Make sure they all have exactly the same information to create a level playing field and show that you are in control. If you give dates to vendors when decisions will be taken then try to stick with them.
 - f. Always keep the bottom line in sight and be careful about the impact of lost time on the day-to-day running of your business.
6. Implementation overview - the big picture:
- a. Any large-scale system implementation can be divided into three distinct and parallel work-streams (Software & Data set-up; Process, People, Technology & Information Organization and Go-live Readiness) running across five broadly sequential project implementation phases, as follows:
 - i. Project preparation & benefits: here you define scope, benefits, key drivers for change and a deployment strategy, and additionally get your implementation team, partners and contracts signed and agreed.
 - ii. Design: here you define how you want your business processes, system and users to operate post go-live, understand what will change, analyze and start cleaning up data, configure and demo the software and specify any forms, reports, interfaces, data conversions, enhancements or workflow scripts required.
 - iii. Build & test: after building any necessary forms, reports, interfaces, conversions or enhancements you complete end-to-end integration scenario and user acceptance testing, revise job description and business policies for the go-live, perhaps document your new business processes as swim-lanes, and complete initial cutover planning.
 - iv. Go-live preparation: final tweaks are completed and acceptance test issues resolved, as well as rehearsing data loads and delivering end user training. Contingency plans for the go-live (back up paper copies of key documents, communication plans, what-if scenarios considered) are also completed in this phase.
 - v. Go-live & support: aside from the actual cutover which involves final data uploads and turning off old systems, this phase includes go-live end user support, tracking and fixing go-live issues, measuring business performance through the typical go-live "performance dip" and ramping up business performance.
 - b. Plan on an initial implementation scope and scale slightly more ambitious than you are comfortable with, between 10-100 users in each 'go-live' and with a broad scope each time while avoiding partial retirement of any legacy systems. Consider going live first with the group that is most "ready" or has leadership that is most open to the change.
 - c. Be a scout; if you fail to prepare you are preparing to fail. Take stock of the big picture and build a plan that works for your business. Break every challenge down into manageable chunks, phases and tasks.
 - d. Only do one of these complex projects at the same time.
 - e. Always talk to other businesses that have done a similar implementation as yours, and then share your knowledge with others who follow after you.
7. Project preparation & benefits:
- a. Create a list of tangible (monetized) and intangible benefits and have your business leaders sign-off on them. Consider how you expect to realize benefits over time and how to measure them with the new software.
 - b. Negotiate a win-win contract is always better than a win-lose because you want a good working relationship with your software and implementation vendor(s) going forward.
 - c. Always have a lawyer check over the vendor's license agreement before you sign, but watch out for lawyers who come up with fruitless concerns in order to justify their fees causing unnecessary delays and loss of goodwill between you and your vendor.
 - d. Keep your second placed software vendor in the running until you sign a contract with your preferred vendor in case something goes wrong during negotiations.

- e. Make sure everyone is aiming for the same success and link their rewards so the team all pulls in the same direction. Measure (and reward) everyone - staff, consultants and software vendors - on the same overall success of the project.
 - f. Have at least one outside partner or consultant in your team and never more consultants on hand than your number of internal staff in the project team.
 - g. Projects like this cannot be done on a part-time basis; your core team will have to spend at least six hours a day on the project. Here is where you learn that there are 24 hours in a day. With some hard work, creative backfilling and delaying of unnecessary tasks, many team members can do their core day jobs and the project at the same time.
 - h. Ensure that top management of your business, from the CEO down, is behind and actively supportive of the project. Not only does it improve focus and morale, executive alignment behind the project makes it so much easier to resolve business issues that come up, such as policy or organization changes or getting managers to pull their weight.
8. Design:
- a. Establishing good momentum is key at this early stage of the project. Be aggressive and plan workshops to be completed within a couple of weeks, keeping the core team focused in the new system, with no time to prepare long-winded project plans or PowerPoint presentations.
 - b. Understand your benefits and let them inform where you spend the most effort during the design phase. Aim for best-practice adoption for all non-core processes, and only request custom development for core processes that set you apart from your competitors.
 - c. Measure your benefits throughout the project and once you have gone live.
9. Build & test:
- a. Technology is only a part of your implementation. The softer issues such as organization and process change, business readiness, and end user training and acceptance are harder to do and more critical to the success of your project than the technology solution you give them.
 - b. Schedule 1-2 steering committee meetings per month to escalate and resolve issues and policy decisions. Make sure that your entire senior management team, plus representatives from all vendors attend these meetings. Have a formal agenda and agreed risk and issue management procedures.
 - c. Avoid custom development even if it appears to fill gaps in the software. Never modify the code delivered by the vendor and don't expect the product to work out of the box in all scenarios. Testing is there to find (and fix) issues.
 - d. Don't underestimate the effort required to find, clean, prepare and then upload your data into the new system; even for small organizations the volume of data required can be staggering.
 - e. Don't forget to keep everyone in your business informed at each stage of the project. Treat every event, communication, email or workshop as a chance to get employees behind the new system and the new way of working, as a way to undermine resistance that might crop up later during the go-live.
 - f. Keep training simple and close to the go-live while remembering that it is not the system, but your end users, who will drive any improvements in your business.
10. Go-live preparation:
- a. Triple-check your data, and implement counts to compare data between old vs. new systems before going live (e.g., inventory quantity on hand, sum of standard prices, etc.). Allow conversion programs to be run more than once.
 - b. Don't start preparing training materials too early when the system is still being changed, and focus training on the end users job roles, what and how to perform transactions in the system. Try to avoid building fancy slides.
 - c. Expect your end users to only want to learn how to use the new system after the go-live. Be on hand when they need your team to support, guide and coach after the go-live. Be part of building any temporary work-arounds as required.
11. Go-live and support:
- a. Assume anything that can go wrong will go wrong. Murphy will strike, so prepare!

- b. Have a resilient project team working toward a common goal and expect a few last-minute glitches and late nights. The less “excitement” around the actual go-live, the better.
 - c. Measure business operational performance before and after the typical go-live “dip” to replace whining and speculation from users with facts and figures about the health of your company.
 - d. List, agree and above all, measure benefits of the new system on your company. After all, that is why you did this in the first place, right?
12. Continuous improvement - the go-live is the start of the journey:
- a. Don't stop working to improve the system. You will identify potential improvements worth making and uncover bugs or shortfalls after even a successful go-live. It takes time and effort to polish the system after implementation.
 - b. Measure the benefits you promised with the new system and processes to make sure that they are realized. Bake them into next year's budgets if you can.
 - c. Include your staff in the mission to improve your business and listen to suggestions from everyone. Instill a culture of continuous improvement by and from all.
 - d. Ensure that you have a solid service level agreements (SLAs) in place with vendors and between your internal IT team and your end users, so that your support strategy for the new system is clear and functional.
 - e. Understand the difference between bugs or issues and enhancement or change requests. Enlist the help of the wider user community to persuade your vendor to develop the enhancements you want.
 - f. Always maintain a firm grip of your data and use it to measure the success of any changes that you make. Improvements to your business must be quantifiable.
 - g. Be in a position to extract your data should you decide to opt for a different system down the road.

Footnote

This eBook attempts to arm the small business owner with the tools and tricks of the trade to successfully select, implement and use commercial off-the-shelf software like a fortune 500 company. Often, smaller business get overwhelmed and can be treated like little people by large (especially enterprise resource planning or ERP) software vendors.

Some of the tips in this eBook are there to overcome deficiencies in the software tools available in the market today, either because these tools are complicated, hard to use or data hungry (i.e. you put tons of data in but get nothing useful out). It is true that the new breed of cloud-based business management tools inspired by consumer apps and mobile user-friendly screens are making implementations and training easier, but don't be fooled. Even the best cloud-based business management tools like Salesforce.com or WorkDay offer a point solution covering a small part of your business such as CRM, HR or financial accounting, which does not require much tailoring for your industry. End-to-end solutions supporting all of your business are still generally complicated, hard to implement and expensive. But they are still so much better than deploying 10, 20, 50 or eventually 500 different point solutions with a spaghetti junction of data bridges and interfaces holding everything together.

What if there was a modern, easy to use, cloud-based end-to-end business management tool which worked for your business, combining the strength of an industry-tailored end-to-end or ERP solution with the ease of implementation, and use and support of a modern cloud-based point solution? The good news is that there are a few.

The vision behind iBE.net

One of these systems, [iBE.net](#), has been developed by the author of this book specifically to help smaller and medium sized businesses outgrowing Quickbooks to get on a level playing field with fortune 500 companies who can afford Oracle or SAP. iBE.net has been developed from a clean sheet as the most intuitive yet sophisticated cloud-based business management tool available. It is initially targeted at professional services firms such as IT and business consultants (not florists, sorry!) but it is built on a scalable platform including world-class collaboration, mobile device access, work-flow and big data analytics, so it can be quickly scaled and extended to other modules and industry sectors. It is based on modern architecture and user interface principles while recognizing that every industry and every business is unique, so it is as flexible and configurable as large-scale ERP systems like SAP or Oracle. Best of all, it is priced at the same rate as a point solution so it is affordable to smaller businesses, and can be deployed initially with just a single module such as project management or timesheets and then extended over time to cover the entire business.

We hope that if you read this eBook, you felt it was not biased towards iBE.net. Our advice to look at 10-15 different options for end-to-end solutions and narrow down based on independent hands-on evaluations and reference customer discussions still stands, whether you narrow down to iBE.net as the best-fit solution for your business or to one of our competitors.

If you enjoyed and got valuable insights from this eBook then please pass it onto friends and colleagues in other small businesses contemplating a new end-to-end business management system. And include iBE.net in your (and their) list of suppliers when looking for potential solutions.