

Process Solutions Tom Bellinson

## **Process Automation: Keeping What You've Got!**

In the business world, we are programmed to believe that it is important to keep up with the latest technology. As both a technologist and a business process professional, I often find myself torn between two worlds. In the end, I try to revert to the value proposition between the cost of technology investments and their benefits. This is not always easy to do.

In my final Column of the year, I'd like to talk about two scenarios that I have witnessed repeatedly over the course of my career:

1. Wanting to throw out your existing enterprise resource planning (ERP) software
2. Wanting to automate an un-automated process

Once projects like this get a head of steam, they are hard to stop. Let's take a look at the lifecycle of each of these scenarios in turn.

### **The Case of the "Obsolete" ERP Software**

To be sure, there are many good and valid reasons to replace your ERP software. There are also not-so-good reasons. Here are some of the reasons I have seen over the years:

- The software vendor is going out of business
- The software is being "sunsetting" and will no longer be supported
- Our company has outgrown the software and it no longer has sufficient capabilities to support current business processes
- The software runs on hardware that is so outdated that it is increasingly difficult to maintain it
- The software has not been updated in a long time and it looks old and "clunky"
- The software has bugs that must be worked around, which frustrates users
- The data has become corrupted and the system doesn't do what users want
- Users hate the software and have lost faith in its ability to support their processes

This list begins with good reasons to replace your ERP software and transitions to...less than good reasons. Can you tell where the breakpoint is? The first four items are perfectly legitimate and beyond the control of your company (except maybe outgrowth, but growth is a good thing, right?).

The next two may be valid, but they must be weighed against the cost and hardship of swapping out your ERP software. The last two are almost always bad reasons to discard your existing ERP software. Unfortunately, they are also the most common reasons (in my experience). If your reason for getting rid of your current software lies in the last half of this list, consider the move carefully.

## What it takes to Swap

Before we discuss reasons not to swap out your old software, let's take a quick look at what is involved in getting a new product. Assuming that you do it right, you will need to do the following:

1. Perform a complete process review (and documentation) for all existing operational processes
2. Evaluate processes to discover causes of inefficiency and low quality
3. Develop requirements for the software based on the detailed needs of each process
4. Either hire a consultant or research to build a list of possible solutions
5. Do a request for proposal, get demos, perform due diligence, negotiate a contract and pay for new software (this is actually a lot of steps)
6. Organize an internal implementation team and possibly hire a project manager
7. Possibly pay for, design and evaluate customizations
8. Train employees on new software
9. Clean old data and migrate it to the new system
10. Ensure all processes are aligned with new system functions
11. Perform conference room pilots and remediate as necessary
12. Go live!

This is a huge effort. It can take more than six months just to find a good solution. Some vendors will claim that implementations can be done in under a year. Their definition of "implementation" is likely different than yours. Implementations are often broken into phases. The first phase should include only functions that allow you to minimally run your business. Subsequent phases will bring up all the other system capabilities relevant to your business. With pauses between phases, it is not uncommon for this process to take three to five years to complete. Keep in mind that the subsequent phases are usually when you derive the anticipated value of the initiative, so if you stop short, you get no value for your time, money and effort.

## To Remediate or Replace?

On more than one occasion, having been hired to help a company replace their ERP software, I have come to discover during the initial steps of the process that their current system was never fully implemented. Usually, the company is considering a replacement for one of the bottom four reasons on the list above. What happens in this case is that employees feel that the software does not have all the features needed to support their business processes. Further research reveals that in fact the software does have the required features, but it has either not been properly implemented or available modules have not been implemented at all.

Another situation that is common is that processes have been modified to work around absent system functionality. These patched up processes become the new

normal. When new software is evaluated, these “patched” processes become the basis for requirements, rather than evaluating how processes should be designed.

As an outsider, I can try to guide the company’s leadership towards reconsidering the change. With much less effort and expense, they can clean up their data and/or properly implement required system modules. I have had leaders tell me that their employees have become so disillusioned with the current software that they will switch as a morale booster. This is odd because I’ve never seen an ERP software implementation boost morale before.

## The Case of the Un-automated Process

Nowadays, there seems to be a push to automate every process at least to some degree. Companies like Amazon and Newegg have completely automated order management processes. Humans only need intervene when something goes wrong, which is rare and even then a highly automated system connects the user with an agent. Even the returns management process is mostly automated. Bricks and mortar operations have even gotten on the automation bandwagon with self-checkout lanes. With increasing options for automation becoming available, why not take advantage of the benefits? Because with benefits come costs.

Managers often overlook the maintenance costs associated with automation. While people can be less consistent and reliable, they are also generally more flexible and adaptable. Before considering adding automation to a process, examine its needs. Amazon has spent many billions of dollars creating a system that can handle all the many variations in their order management process. They continue to spend lots of money every year to [maintain and improve that system](#).

## Is It Worth The Effort?

Before choosing to automate a process (or set of processes), it’s a good idea to do a thorough analysis. There are many good articles right here on BPTrends to help you with that, so I won’t repeat the methodology for that exercise here. However, make sure that whatever methodology you use includes a cost analysis of material and labor, as well as potential opportunity cost (lost sales, increased margin, new markets, etc.). One methodology that I have found recently is called Value-Driven Process Management. You can read about it in [a book](#) by Franz and Kirchmer.

Armed with real data, you can quantify a reasonable budget for the automation project. If the automation involves custom software development, my recommendation is to get the most reliable and conservative estimate you can and double it. This may sound cynical, but there are two factors at play here: 1) the people doing the estimating are unlikely to have enough information no matter how familiar they may be with the domain, and 2) the people that educated them about the domain probably don’t know exactly what they need the software to do. Reparations for these deficiencies will require ongoing discovery and more money.

Non-software automation projects often involve the selection of multiple products that must be integrated. There may also be software components that must also be integrated with hardware. The cost estimates of integration projects are also subject primarily to increases as opposed to decreases. The good news here is that hardware estimates tend to be more accurate -- unless you forget something major, which only happens sometimes.

With “the numbers” in hand, it is a simple matter to calculate the payback period of the initiative. Before you embark on the time-consuming effort to capture all the necessary data, your intuition may play a part in deciding whether to even pursue the discovery process as it is clearly not trivial. It is best to remain skeptical about automating processes unless there appears to be a clearly substantial opportunity.

Students of modern computer technology are aware of the many advances taking place that push computer automation further into the realm of traditionally human functions. Artificial intelligence has come out of the laboratories and is become increasingly mainstream. Articles about IBM’s [Watson](#) technology and Google’s [TensorFlow](#) system. It is easy to assume that AI is available to solve process automation problems. However, implementation of AI remains expensive.

## The Perils of Software Development

IBM has commercialized their [Watson](#) product and brought it to the cloud. They provide software components that provide the ability to access sophisticated AI algorithms without writing them from scratch. While this may bring the sophisticated world of AI one step closer to business software development, there remains major challenges to develop sufficient problem domain knowledge to effectively tell the algorithms how to behave.

Most of building a solution is getting the design right. Using iterative techniques attempts to minimize the impact of design errors by providing the ability to refactor code as new information is uncovered. This can create a false sense of security because those outside of the development team assume that issues can be easily fixed. As we have seen at the University of Michigan, it is not uncommon for fundamental inconsistencies to reveal themselves much later in the project. So, even though there were regular reviews with the stakeholders and everything seemed to be going fine, a new requirement uncovers a flaw in the basic architecture of the system. This could result in the need to either scrap the whole thing or make compromises that if not limit the system’s capabilities, will add cost to maintenance and new development,.

Organizations are reluctant to scrap work that may have taken many months (or years!) to develop. They often make the mistake of assuming that the redevelopment effort to bring them back to the current point will take the same amount of time. It rarely does. All that has been learned is not lost. Having done it once, discovering ideas, approaches and interfaces can be short circuited. The psychological impacts of scrapping an application is far greater than other factors. Those involved will often expend even more effort to save their prior work product.

## Pick the Right Reasons

If this Column made one point it is that IT projects almost always take far more time, effort and money than was originally anticipated. There always seems to be more people under rather than overestimating the work involved. It is no wonder, they are the ones that will get the work and if it is how they earn their living, they probably want the work. It pays to be skeptical.

It also pays to have the right data to make good decisions. Since getting it will require significant effort in most cases, a critical gut-check is in order. The old adage

“if it sounds too good to be true, it probably is” applies. Armed with a good strategy, you will be better equipped to choose the right projects to pursue. Choose wisely!

### Tom Bellinson



**Mr. Bellinson** has been working in information technology positions for 30 years. His diverse background has allowed him to gain intimate working knowledge in technical, marketing, sales and executive roles. Most recently, Mr. Bellinson finds himself serving as President of a BPM related software start-up company called UnaPage that provides solutions based on Microsoft SharePoint. From 2008 to 2011 Bellinson worked with at risk businesses in Michigan through a State funded program which was administered by the University of Michigan. Prior to working for the University of Michigan, Mr. Bellinson served as Vice President of an ERP software company, an independent business and IT consultant, as chief information officer of an automotive engineering services company and as founder and President of a systems integration firm that was a pioneer in Internet services marketplace. Bellinson holds a degree in Communications with a Minor in Management from Oakland University in Rochester, MI and has a variety of technical certifications including APICS CPIM and CSCP.