

# The ERP Software Promise

Tom Bellinson

## ERP Software – A Brief History

In the early days of mainframe computing, accounting was one of the first enterprise-wide applications that became pervasive. At the time, computers weren't very powerful. That meant that the software did little more than collect data from already processed transactions so that financial reports could be generated for subsequent analysis.

Some years later, IBM developed material requirements planning (MRP) software that eventually evolved into what was called manufacturing resource planning (MRP II, since MRP was already taken). Over the years that ensued, both accounting and MRP II software advanced with the power of the hardware on which it ran.

Finally, sometime during the 1990s, someone had the idea of combining these two systems into a single integrated application, and enterprise resource planning (ERP) software was born. Since then, ERP software has continued to evolve even to the point that now ERP software doesn't need to contain MRP II functionality. The term ERP software has come to mean any software that manages multiple aspects of an enterprise.

## ERP Software Today

By the modern definition, a company doesn't need to be a manufacturer to use ERP software. Distributors, retailers, and professional services firms can all find ERP systems that are specifically designed for their needs. As a matter of fact, there are ERP systems that are narrowly designed for very specific types of businesses. For example, a plastic injection molding company might purchase IQMS software that has such features as optimizing the shop schedule based on the color and type of resin in order to minimize setup times; or a business to business (B2B) distributor might purchase NetSuite, a web-based system optimized for inventory and distribution management.

Modern day ERP systems have taken on greater responsibility for managing processes. They enable process optimization by

1. Making decision data available in real-time or near real-time
2. Enabling "manage by exception" communications with alert technology
3. Trapping or eliminating data collection errors
4. Enforcing business rules

In addition to enhanced depth of functionality, the breadth of functionality has expanded as well. On top of the traditional financial management and MRP II functionality, there are a host of new modules that provide functionality for shop floor control, labor reporting, warehouse management, customer/supplier relationship management, customer and supplier portals – the list goes on and on. Many products claim to have dozens or even hundreds of modules.

## The ERP Software Promise

ERP software is one of the more costly investments that a company will make. The costs are not all so obvious:

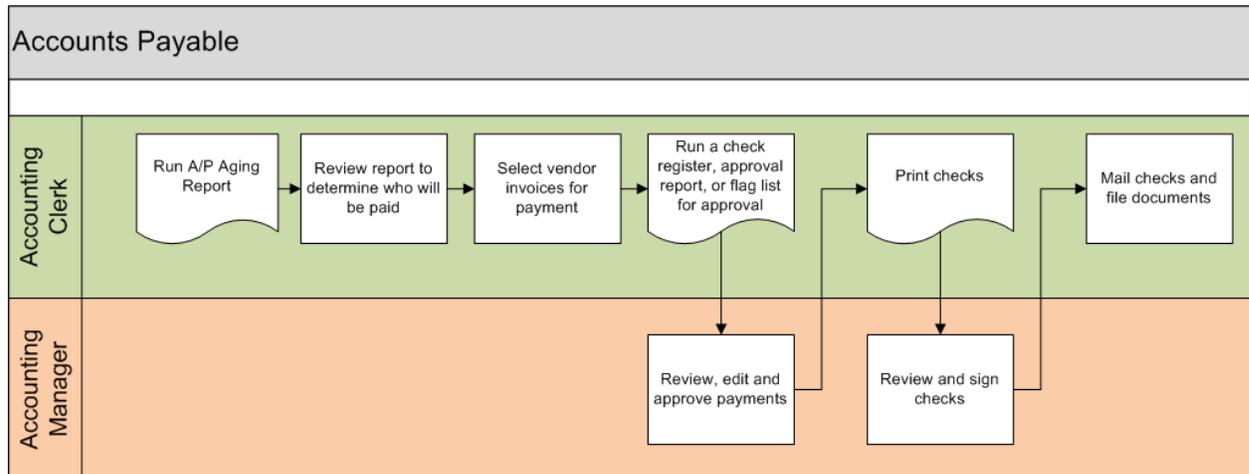
- Lost productivity
- Overtime
- Employee burn-out
- Customer service disruptions
- Supply chain disruption

These are in addition to the obvious costs such as implementation and training, which often end up taking more time and money than initially anticipated. So why do companies go through it? Clearly, they believe they will derive some of the benefits discussed above.

In a recent BPTrends Monthly Advisor, Paul Harmon made a reference to what Michael Porter called “Competitive Advantage” in his book of the same name. The argument is that since ERP systems promise to provide a “best practice” solution, companies that use them will find themselves operating in a “cookie cutter” environment where nobody can win.

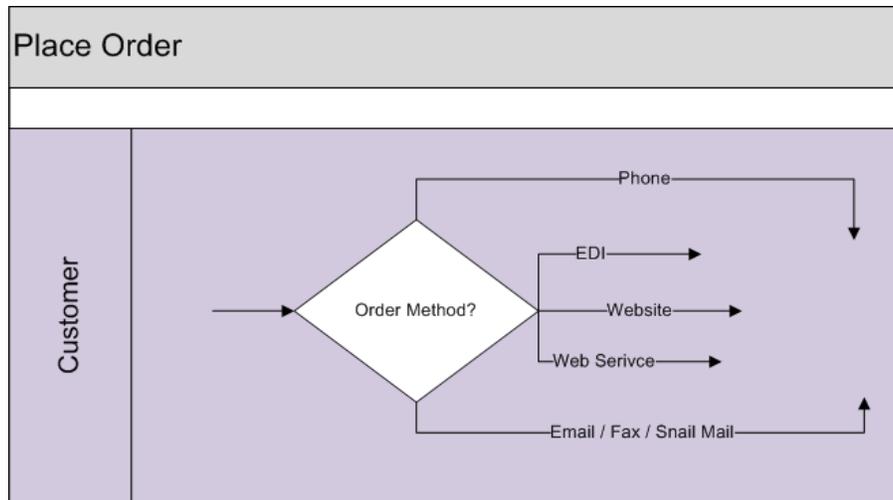
I would contend that this is not the case, at least not entirely. No doubt, many ERP system modules are built on business best practices. However, these modules tend to be the enabling processes that are not critical parts of the customer value chain. For example, how companies process payroll or pay vendors is not likely to provide a significant competitive advantage in the marketplace. Porter argues that any process can be part of a competitive advantage, and I would not dispute the point. The detailed analysis of this point usually reveals policy level practices that fall outside the scope of the mechanics of the process execution.

Let’s take the example of accounts payable. This is a very mechanical process that usually looks something like this:



Variations on this will not be likely to relate to automation. More likely, they will be policy decisions about whom to pay and when. Any fully functional A/P system will permit the user to select vendor invoices to pay by a variety of criteria because this is all the flexibility that this process needs. The system supports the best practice of evaluating whom to pay and then streamlines the process of producing the checks. Some systems will support electronic payments, so companies that need this capability will be certain to get a system that offers it, but this only serves to eliminate the “mail checks” step of the process. The basics of the process remain essentially unchanged by this variation.

The point here is that a process like this requires the software to have limited flexibility in order to best serve the interests of a wide range of companies. This is not the case with all processes. Generally speaking, any process that touches the customer will require a very high degree of flexibility. As consumers’ expectations rise, it becomes increasingly critical that world-class organizations have processes that are flexible enough to offer a broad range of strategies for serving each customer’s preferences. Whereas the A/P process above has few decision points, customer facing processes will often have more decision diamonds than actual tasks. Not only that, but many decision points may have many output options:



Fortunately, ERP software designers are acutely aware of these process variations. Any product with a large installed base and years of field operation will have had to address most common variations. Further, anyone crazy enough to enter the ERP software business, as crowded as it is, will be very intimate with the need for massive amounts of flexibility in these processes.

However, there is a gap. The gap can be summed up with the statement: flexibility equals complexity. As a business consultant and ERP system specialist, one of the most ubiquitous comments I hear is, “The system can do it, but we’re not using that functionality.” I have, on more than one occasion, seen companies replace their old ERP system in order to purchase a new one with the same or even less functionality because they believe they will never be able to implement all of the functionality of their old system. There is a “promise” that the new system will be easier to use. To be sure, the interfaces on newer ERP systems are slick. Some have found ways to hide unused functionality so that users are not overwhelmed, but the problem is usually not *using* the functionality, but rather *configuring* it.

Unless a company has access to a business process practitioner, they will tend to approach implementation by assessing what functionality the system has and how they can use it. In essence, this is the tail wagging the dog, but for small and medium sized organizations without the budget for full-time process specialists, this approach is the norm. Enter business process management systems (BPMS).

### BPMS Software: A Patch for Deficient ERP Software

This heading may appear to be an attack on BPMS software, but it is not. If anything, it is more of a prediction of the future. To understand this statement, let’s step back and look at how ERP software typically gets designed. Most of the products on the market began with one or two clients. Those clients may have had some sort of process maps. We can even assume for our discussion that they did. After having collected the requirements from the client(s), the designers set out to build a system that meets those requirements. Now that they have a product to sell, they offer it to new clients. Naturally, those clients have somewhat different requirements. So, the developers proceed to add the necessary functionality. This process continues over the years. With each new iteration, the product becomes more and more complex. What started as an “easy-to-use” application is now a highly flexible, highly complex application. Attempts to hide unneeded functionality often serves to add yet another layer of complexity for system administrators.

This is not the case for BPMS solutions. Why? Because they take a toolkit approach to designing their solutions. Designers of this type of software recognize that there is no way they can anticipate the endless variability of the processes they will automate. So, rather than design

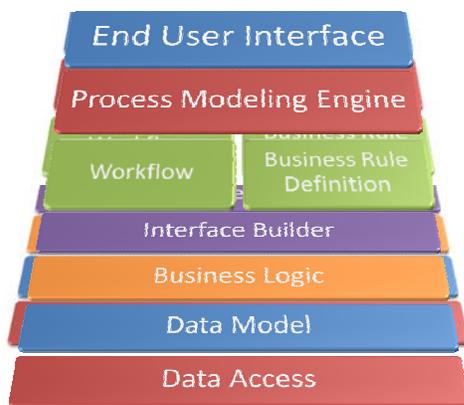
fixed and finite functional modules, they allow the end-user of the software to construct their processes and create the supporting user interfaces.

The challenge for BPMS software providers is that their systems are not the ultimate central repository for the data they create or the data they use to validate some user input. This statement is predicated on my own philosophy that the ultimate value of an ERP system is to have a single, fully integrated solution that manages the entire organization. Some may argue against this premise, but, after years in the industry, I can truthfully say that I have never seen a situation where having multiple systems that needed to be integrated together was more efficient than a single system doing the same thing. I am not talking about third party products that were designed to become a fully integrated part of the system. I'm talking about systems that have some generic integration tools for data import and export (however automated they may be). Ultimately, BPMS software is just another integration point that must be managed.

No matter how easy the BPMS tool makes it, users are still left to the individual philosophies and strategies of two separate software firms. And, while their philosophies and strategies may be in alignment when the decision to integrate is made, there is no promise that it will remain that way. From this perspective, the drive to use BPMS software to front-end an ERP system is driven primarily by the need to simplify the tasks associated with aligning optimized processes with ERP system functionality.

### The Future of ERP Software

There are no technological reasons why ERP software designers don't build solutions that are based on a toolkit approach. It just hasn't been done yet. Whether a BPMS software developer decides to build an ERP system, or an ERP software developer decides to front-end their system with a BPMS toolkit, these two solutions are destined to become one. It is only when they do, that ERP solutions will truly deliver on their promise. Imagine an ERP system that had the following architecture:



This architecture allows the end user to model processes using a workflow tool and to define associated business rules. These definitions are fed to the interface builder and business logic interpreter, which, in turn, restructure the data model accordingly so that the system conforms to the processes as defined.

This design allows the end-user's administrative staff to be able to configure the system to conform to their unique (or not so unique) way of doing business, while only exposing the functionality necessary to meet process requirements. The end result is a system with much less complexity for those individuals tasked with configuring the system because they are configuring

the system based on workflows that they already understand.

Each process that the ERP system supports could have a standard definition preset. The user need only modify the process flowchart to suit their specific needs, or they could build the process flowchart from scratch if they were so inclined.

The architecture of a system like this would be more complex, and early versions would undoubtedly not have the breadth of functionality that a wide variety of organizations would like. However, the important point is that the designers could continue to add new capabilities without adding complexity to the end-user's experience.

SAP clearly understands this model. Their strategy, which employs a BPM version of their NetWeaver product, allows users to build BPMN flowcharts that drive the business logic for a

given process within the system. SAP has an advantage here as their product has always been more of a toolkit type of product. However, their traditional R/3 and R/4 environments have required administrators to have extensive technical training in order to manipulate the system. The NetWeaver product represents the beginning of a toolkit that less sophisticated users can employ to design process-specific applications.

Unfortunately, for most, SAP is too expensive to procure and deploy. Middle market products from companies like Infor, Epicor, and Microsoft will need to adopt the BPMN front-end strategy in order to bring these tools to the vast majority of companies.

## Summary

ERP software designers have been trying for years to expand on the functionality of their products. The reason is simple: Porter is right – every company must find some way to be unique in order to be competitive in an increasingly competitive marketplace. That means that enterprise software companies must accommodate maximum flexibility in order to serve a broad enough market for their own sustainability.

BPMS software is a relatively new entrant into a space that ERP software vendors have left open. They provide a toolkit that allows business people with relatively basic technical skills to manage the process integration of their information systems. This lapse in vision on the part of ERP software designers will eventually close. This feat will not be performed quickly or easily, so BPMS solutions may have their place for some time to come, but their ultimate demise is not only inevitable, but preferable. It is preferable because the advantage of having a single system that manages all enterprise information will be that it is always more efficient and less expensive than integrating multiple systems. The arguments against this revolve around functional deficiencies and the need to find “best of breed” solutions. Fortunately, the BPMN toolkit approach will go a long way to eliminating these arguments.

-----

## Author

Tom Bellinson consults with risk businesses in Michigan through a State funded program administered by the University of Michigan. In that capacity, he consults with companies to identify opportunities for cost containment and market expansion.

Prior to working for the University of Michigan, Mr. Bellinson served as Vice President of an ERP software company. He can be contacted at [tjbelli@umich.edu](mailto:tjbelli@umich.edu).