The Top Ten Things to Look for in ERP

When you look for an ERP system, this report will help you to concentrate on the things that really make the difference between success and failure. You can probably, for example, figure out how to live with *x* characters in the account number; **there is, however, no way anyone can figure out how to live with endless implementations, blown budgets, bloated infrastructures, and truncated careers.** Good luck in your search.

1. **No Proprietary Architectures.** This really shouldn't be much of a problem anymore, but you should look for widely used, industry standard hardware and software architectures. Try to get to get (several independent) total cost of ownership (TCO) studies for each solution you look at.

2. **Profitable, Focused Partner.** Now, more than ever, you need to look for a partner who has shown that they are well managed, know how to make money, and will be in business next year. Additionally, you should ask yourself, and them, whether they are diluting their efforts trying to be all things to all companies, or whether they are leveraging their R&D investment by focusing on your industry.

3. **Are the Data Being Captured?** Even if you spend megabucks for a "trophy" system, you are still going to find that some data are not being displayed, or not in the way you want. The key question is: Are the data being captured? You can't report what you don't record. Follow-on questions include: How easy is it to create reports and inquiries? How easy is it to add new data fields? Or is there a simple a way to rename existing fields so that they are meaningful to you?

4. **Minimize Bolt-ons.** The integration of so-called "best of breed" point solutions has been widely shown to be very difficult, expensive, time-consuming, and, usually, unsuccessful. A far better, lower risk approach is to look for a vendor who has most of the functionality you need already integrated into one package.

5. **No Hiring.** You should be able to implement the system using the people you already have. You should not have to add people to your technical infrastructure; your people should be able to learn any new skills they need in a reasonable time at a reasonable cost.

6. **Minimize Cost to Acquire and Maintain.** You may have had the experience of being nickel and dimed by a vendor when you wished to add new users, or go to a larger server, or when a new module was introduced. A better, more customer-friendly, model, that helps you to predict and control costs, is **unlimited user pricin**g. You'll have to look hard to find vendors who routinely offer it. Also, you should find out what the vendor's policy is on new modules. Again, the most customer-friendly policy would be giving the new module at no cost to users who are current on their maintenance contract. (By the way, you'll want to look closely at yearly maintenance. Some companies use it as a cash cow and don't deliver much for it. Here's an area where you really should question existing users closely). A word about customization: Don't. Seriously, you should be looking for a system that meets at least 90-95% of your needs out of the box. Customization should be avoided wherever possible. If you feel you need customization, try to avoid it until core, unmodified functionality is implemented. And then make sure you're doing business with a vendor who is both staffed to provide the customization and willing to take on the (greater) burden of ongoing support for it.

7. **Minimize Time and Cost to Implement.** Even if you adhere to the software cost guideline, in most cases the cost to implement the system should not exceed one dollar for each dollar of software cost. Exceptions might include if you have multiple locations, or locations that are extremely remote, or your current staff is already maxed out. Look for **sensible training solutions that minimize travel costs and maximize your training dollar like interactive, video conference-based training**. A detailed project plan that is unique to your company and not just a generic template is key to your success. Ask potential vendors what proven methodology they have to understand your company and your objectives and then turn that understanding into a detailed project plan. The plan should precisely lay out tasks, timelines, and your company's and the vendor's responsibilities. On any particular day, there should be no doubt in anyone's mind what they are supposed to do. Remember that your risks increase substantially if your project plan exceeds one year; six to nine months would be a better target to shoot at. **Current owners of trophy systems please take note: If your implementation is less than successful for any of the reasons noted here, <u>you may be able to completely fund (software, hardware, services)--and complete within one year-- the implementation of an ERP system that meets these recommendations using only your current projected costs for the coming year.</u>** If you feel that you are going to need to hire scarce and expensive resources during the coming year, **<u>the savings from avoiding that hiring might also completely fund the implementation of a new system, to be accomplished within that year.</u>**

8. **Insist on Web Access.** The death of "client server" (whatever that means) is well documented. Internet access is ubiquitous. You should be able to get at the system using only a browser. (Decide for yourself if additional applets are worth downloading). And, character-based screens should be available for users who prefer them for head-down data entry.

9. **Money-back Guarantee.** You won't find much of this out there. But, really, it is the only antidote to endless demos and analysis paralysis. These are costly to the vendor and to you. Within reasonable limits, you should be able to return the software and get your money back.

10. **Full Disclosure by Consultants.** You may be using a consultant to help you look. Your consultant may offer services in implementing one or more ERP packages. It makes sense for them to let you know what those packages are, doesn't it? Now, it's possible that the package they know and their services in implementing are the best choice for you. But you should have full knowledge about potential conflicts of interest.