AdaptiveGrowth

ERP System Acquisition & Implementation Viewed as a Single Project

It is a truism that in any complex activity, the critical, defining decisions should be made as early in the process as possible. In ERP implementation projects this principle is routinely undermined by the nearly universal practice of separating acquisition and implementation activities into separate projects, performed by separate teams operating under separate control. This causes any number of critical decisions that ought to occur in the earliest phases of acquisition planning to be deferred to the implementation phase, where they are then made by the wrong people for the wrong reasons, well out of the sight of those who have the most at stake in them. These stakeholders have, in fact, needlessly surrendered control of the project to outsiders in the usually unquestioned but always mistaken belief that technical proficiency and product knowledge trump their own needs and abilities in all matters relating to system implementation. What must be turned over to outsiders is the technical process of system configuration – the actual work of making the product do what the user company needs from it. This is a lot, but it doesn't include three things that should never be turned over to outsiders, but often are:

- The working blueprint for what the user company needs the system to do.
- The enforceable definition of project success as nothing less than the working, efficient configuration of the system according to that blueprint.
- The day to day monitoring and control of the implementation itself.

These three things can't be separated. Day to day control of the implementation is ceded to implementation vendors when they are given the task of deciding what the system will do (as opposed to how it will do it). When that happens, the implementation team has no choice but to take control: they've been given the task of defining the project, so define it they must. And day to day control slips away unnoticed when the implementation team is held to some standard of performance other than the configuration of the system according to the user company's precise instructions.

This being the case, why is the job of defining what the system is to do ever passed to the implementers? Because of some common, overlapping misunderstandings.

- It is often viewed as a technical, software-dominated task, either too complex for the user company to do for itself, or too product specific for anyone but a product expert to do.
- The implementation vendor offers business analysis as a professional service; sometimes it seems attractive to outsource this work, especially if the user company believes the vendor can do as good a job as it could do itself.

Most commonly of all, the user company sometimes thinks that they've defined their requirements when in fact they haven't. All three of these notions are misconceptions. We'll take them in order.

2021 Midwest Road *** Suite 200 *** Oak Brook, IL 60523 *** Phone: 630-443-7790 *** Fax: 630-443-7796
E-mail: jfrano@adaptivegrowth.com Web-Site: www.adaptivegrowth.com

First, the description of what the user company needs the system to do is not a technical, software-dominated task: the task is rather one of describing the complex internals of one particular company's business, in common business-oriented terms.

In many ways, a well implemented ERP system is a mirror of a company's operations. In this mirror, a single business event – say, a shipment to a customer – is reflected back to different people in different ways.

To an accountant looking at one aspect of the transaction, it becomes a $135.53 debit to the cost of sales; to accounts receivable, it becomes an open invoice to be tracked; to someone else it represents a sales commission earned; it is a warranty responsibility for a business unit fifty miles away; it is a line item on a dozen sales reports, and part of a summarized total on a dozen more; it is a fragment of consumed sales forecast, that later became a fragment of a master schedule, that still later became a demand against manufacturing that was fulfilled by a now-closed shop order. That shop order incurred fourteen different labor charges and required as many component issues, each of which was built or bought according to an ever-widening chain of events that can be traced for years to come. This is only the beginning – it is drawings drawn and checks written and requisitions approved and tools sharpened, but if you're reading this, you get the idea.

The point is not that it's all so marvelously complicated but that every true picture of such things, every true reflection of this kind is so marvelously unique. All of these individual details, those we've mentioned and those we haven't, happen in countless ways in countless companies, but here they happen as this company does them. Likewise, this particular mirror, as it is configured and used, is right for only this company. Change it just a little and it reflects some other company, or no company at all. To get it right takes almost as much knowledge of what this one company does as it took to perform that transaction in the first place. More to the point, to describe everything that goes into that transaction in just the right detail takes about the same knowledge as it takes to perform and manage it in the course of this company's daily life. What outsider can do that?

None of this is a description of software; the closest it gets to that is as a description of business life that must somehow be reconciled to software.

This is not to say that everyone knows how to create such a description, but we are sure that everyone can be shown how. We do that part.

Second, the business analysis that implementation service companies provide does not create a comprehensive mirror of their client's operations in terms that the client itself can recognize. No matter how well they do their jobs – and most of them do it very well indeed – their audience is their own company. They talk to their clients diligently and diligently feed back to them what they've absorbed, they make many lists and check them all twice but, in the end, the pictures they draw are drawn in the terms of their trade and their product.

2021 Midwest Road *** Suite 200 *** Oak Brook, IL 60523 *** Phone: 630-443-7790 *** Fax: 630-443-7796
E-mail: jfrano@adaptivegrowth.com Web-Site: www.adaptivegrowth.com

If those pictures are the only operative blueprint of their client's business, the implementation team that follows after the business analysts will be speaking a different language than the people they're working for.

Third, companies often rely far too much on the very real understandings they've come to with their chosen ERP vendors in the presale phase. These understandings can simply dissipate in the implementation phase to follow. The basic reason is twofold: first, as discussed above, the implementation team is not working to a precise definition of the detailed meaning of success. Second – and this is even more insidious – the implementation team can also fail to learn much of what the presale team discovered. Of course, a handoff of some kind takes place between the presale and implementation teams, but it tends to be a handoff of summarized conclusions, biased toward system-oriented concepts and terms. The nuanced feel for the client's company that the presale team had to acquire and demonstrate in order to earn the business – the detail behind these summarized conclusions – is lost in this transition. This is discussed somewhat more in the following topic, "ERP System Acquisition Project Planning"; if that still doesn't go into enough detail, and we'll explain.

These three misunderstandings, when they're allowed to determine events, have the effect of deferring the implementation team's full understanding of what they're supposed to be accomplishing; they defer it at least until the implementation phase is well underway, and sometimes until the team believes it's almost done. When that happens, it's not the implementation team's fault; it's the natural consequence of having them work to inadequate blueprints. But when it does happen, the cost of correcting the resulting disconnects – when they finally come to light – is far greater than would have been the cost of preventing them in the first place.