

The Software Implementation Project Failed — Now What?

By **David Shapiro** (January 29, 2018, 11:59 AM EST)

The implementation of a new software system is a complicated, expensive and time-consuming project. Sometimes it goes well; but in an alarming number of cases, it does not.

A software implementation project often begins with a request for proposal process: The company, either by itself or with the help of a consultant, tries to figure out which of the hundreds of software providers in the country has the right product (“Is it a good fit?”) and which implementation vendor is best suited to implement that product (“Have they done this before for a company like ours?”). In many cases, the company is trying to decide whether to develop the software itself or license a vendor’s product — a “buy” versus “build” versus “upgrade” debate.



David Shapiro

Unfortunately, most consultants assisting companies in this RFP process do not have an attorney as part of the team — and that can lead to problems. This is because vendors get to know your business before they start, but your company will have limited insight into the development of the software. In most cases, the only attorney involved in the process is in-house counsel, and their attention is usually focused on reviewing contracts.

When the software provider and the implementation vendor respond to the RFP, they have to make representations about what the software can do, how it will help the company, how long it will take to implement and how much it will cost. The contracts come from the vendors, and they contain a great deal of boilerplate language that is weighted in favor of the software/integrator company.

After that, implementation consultants are assigned to the project; and, depending on the size of the project, this could mean dozens of individuals over several months or years. The vendors will not tell the client that, statistically, a significant number of implementation projects will fail; and they will avoid representing in plain English, “This product is right for your companies’ needs.”

The contracts will also state that the company must assign “stakeholders” and share “ownership” of the project. While this is prudent given the collaborative nature of the project, it creates a problem when the project fails because of defective software or unskilled implementation consultants. And, if anything goes wrong, the vendors will claim it was because of the company’s failure to “own” the project and assign the right resources. There are some in the industry who refer to this as the implementers’ insurance policy.

On the other side of the coin, if the project is successful, the company becomes “referenceable.” The vendor will praise the cooperation and skill set of the company’s IT staff, but of course any income from an improved product or service based on the engagement will go to the vendor.

The project will then typically proceed in stages, from design to implementation to “go live.” Because of the costs involved, most companies cannot afford, along with the implementation vendor, to hire yet another vendor to audit the implementation. This means that the company is forced to trust that, behind the curtain, the implementation vendors are following best practices with appropriate oversight from above and using an implementation methodology best suited for the product and the company.

The Project Fails — What Does That Mean?

This should be an easy one: The project fails when it doesn’t “work.” In some situations, someone flips on the on switch and boxes of product go flying across warehouse floors. But in most cases, the story is not that dramatic. In many cases — usually after the company has exhausted the project’s budget — there is a realization on the part of operations or management that the system will not be able to meet the pre-contract goals for the software.

Tensions will rise; fingers will be pointed; and the parties will likely not agree on how to right the ship, and at what price. Many will tell you that a software implementation is like going to war or starting a major motion picture: once started, they are very hard to stop or change course. The company will find itself in a situation where the vendor will be able to claim that there hasn’t yet even been a “failure” and all that is needed is more time, more consultants, more design and customizations and, of course, more money.

At this point in the project, a key decision must be made: should you give formal notice of your intent to “rescind” the contract in order to potentially protect causes of action and remedies down the road? The intricacies of that decision will depend on state law and where you are in the project; it is crucial.

If the company has gone live, you will know the failure when you see it: The system is not working as promised, or the system is causing headaches, and the company has not realized the savings and profits expected, not to mention the money spent on licensing fees, maintenance and implementation services. In some cases, the very existence of the company is at risk because the software has not provided the benefits promised.

Post-Failure Options

If the company has concluded that the software project is a failure, there are a number of options. They are all based on one unshakable fact when it comes to dealing with software implementation failures: They are incredibly expensive to remediate.

Walk Away From the Project and Start Over With Someone New

Many companies with a failed project elect to write off the investment in a new system as a loss and move on with a different product or design their own system. From my experience, this decision is based on two reasons, one reasonable, and one not.

Reasonable: It is too expensive to litigate and we want our people focused on building a new system.

Unreasonable: The failure was at least partially our fault because we were helping the vendors and maybe we could have done something different. This is unreasonable if, based on your observations, the project failed because of a bad product fit or inexperienced or unskilled implementer which, in my experience, is often the case.

Reject the unreasonable: It is rarely the case that a massive mission-critical software failure is the result of lack of effort, commitment or skills on the part of the company. The relationship and the contracts, however, are often structured in such a way that operations, management and IT professionals may start thinking that way.

Redefine the Project and Continue Working With the Vendors

Experts in the field also like to use this analogy for major software implementation projects: Patty Hearst and the Stockholm syndrome. In major software implementation projects, the IT department has put its credibility on the line — they picked the product and the vendor, often without operations and management taking a leading role. Careers hang in the balance.

The IT professionals can become hostages to the project and the vendors running them; they come to believe that the IT department and the vendor are in it together and have a common enemy, whether it is the users or upper management. Retreat becomes an impossibility and so new contracts, timelines and milestones are negotiated.

Litigate

Many companies take the position that litigation in the software implementation world should always be the company's last option and should be avoided at all costs. This conclusion is fueled by the same twin demons mentioned above: It is too expensive, and maybe the company is at least partially to blame. This, however, ignores two vital realities.

First, it used to be the case that the only option in a failed software implementation case was to retain a BigLaw firm. That, of course, meant millions of dollars in fees, no matter what the out-of-pocket damages were. Litigation was an option open only to corporations with significant cash flow. Today, however, there are small firms who have developed niche practices in this space and who offer alternative fee arrangements. This is also true for jury consultants who can add value throughout the process, from before the complaint is drafted until the jury is seated.

Moreover, in today's world of "virtual" offices and professionals (e.g., paralegal services), the cost of litigating a software implementation failure no longer has to run into the scores of millions of dollars. Litigation funding has also made software implementation failure cases affordable to small or midsize companies who cannot marshal the resources necessary to sue a major vendor. Small companies that have been damaged by a failed software implementation project will find themselves in a David v. Goliath situation: Vendors will protect their brand by vigorously defending an action. In order to stay in it for the long haul, companies need to identify representation that is flexible in its fee structure and with a reasonable hourly rate.

Internal opposition from folks in operations or IT who were involved in the project should also be questioned. Even Patty Hearst eventually stopped being Tanya. These professionals obviously played an important role in the project, but they are not the experts in the vendor's software or how it should best be implemented. This is a psychological trap that can happen because of how the contracts are

structured. Companies hire software providers and implementers for their expertise in their product and services, and the company's professionals should be allowed to rely on that expertise and have the confidence to believe that the software will do what vendor said it could do.

If you go down this road, you obviously have to think about the likelihood of success. In making that determination, I recommend focusing on the following:

- What did the company pay the vendors for licensing fees, maintenance and implementation? Based on my experience, these cases will settle for four times that amount; your litigation budget has to make sense given this number.
- Did the vendor make explicit, fact-based and "material" representations of fact that induced the company to contract with it (i.e., not "puffery")? If yes, then — depending on the jurisdiction — you may have the right to lost profits.
- Are you within the statute of limitations for your state's consumer protection act? In most jurisdictions, if the vendor committed fraud or negligence in providing the goods and services the company obtained, they are liable under the state's consumer protection act, but those statutes have notoriously short statutes of limitations. And if the company complained about the progress of the project before starting the action, it will be faced with the argument that the clock started ticking long before the suit was brought.
- Is your damages model based on non-speculative and verifiable numbers? This is critical, especially if you have facts upon which you can credibly claim fraudulent inducement. Consider bringing in a damages and litigation consultant before drafting the complaint and pulling the trigger.

The best way to avoid a failed software implementation project is through a rigorous RFP process, with an experienced attorney on the team and an outside vendor auditing the project and making course corrections when needed. Unfortunately, this rarely happens.

Further, once the project fails, many companies will not consider litigating because they view it as too expensive; and, in many cases — because of the structure of the relationship and the way the contracts are drafted — the IT professionals may blame themselves. These assumptions should be vigorously questioned.

David J. Shapiro is the founder and shareholder of The Shapiro Litigation Group, a virtual law firm based in New York City. He has represented corporations in suits against software providers and implementation vendors.

The opinions expressed are those of the author(s) and do not necessarily reflect the views of the firm, its clients, or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.