

The Seven Deadly Sins of Software Implementation

Press Release 2005-02-01 (Source Unknown)

Did you know that according to a report of the Standish Group (www.standishgroup.com) 31.1% of software implementation projects are cancelled before completion and that another 52.7% go live substantially over budget?

To put it another way, this means that 3 projects out of every 10 are cancelled, another 5 out of 10 are substantially over budget and only 2 out of 10 go live on time and on budget. What's more the average cost of projects that go over budget is 189% of the original estimate. To put that into perspective, if you budget a million dollars for a project and it's not cancelled, it's likely to cost you almost 1.9 million dollars – in other words, almost twice as much as you thought!

What are the mistakes that people make that lead to implementation disasters, cost blow-outs and schedule nightmares?

Doomed before you start

Obviously, an implementation won't be successful if choosing the right software isn't done correctly.

If you pick the wrong system, you're doomed before you start. No matter how good your project manager is, how good your staff are, and how closely you monitor the project, if you choose the wrong package in the first place, the entire project will go belly-up. So what are the 7 deadly sins of software implementations?

1st deadly sin - set an unrealistic budget and timeframe

Many projects start with a false premise. They have an unreasonable budget and an unrealistic schedule set by people who have no real understanding of the day-to-day issues. If you want to go live on budget and on schedule, the first thing you have to do is set a realistic schedule and a realistic budget to take into account the inevitable problems that occur in software implementations.

Preparation is the key if you want to avoid this first mistake. The more preparation you put into the project the more likely your project will stay on track. Other measures to ensure success include defining the project succinctly and resisting modifications (putting in unnecessary or excessive modifications) - the fewer the changes to the package – the easier it will be.

2nd deadly sin - treat the project as a low priority

If an implementation is seen as getting in the way of business, rather than being an integral part of your business growth then you shouldn't even go ahead with it. The reason that you're implementing the software is to make your business better, to overcome problems and to be able to do new things. And that should be your priority. If implementing the system isn't your top priority give it away because

you're not doing it with the right attitude.

It follows that if you don't treat the project as a priority you:

Leave key people out of the project

Any software implementation is difficult. Even the easy software implementations are difficult. Even the smoothest ones run into problems.

And the people that know your requirements and your business best are your team leaders. They're the people who should be given the task of managing the implementation. Too often the key people are deemed to be too important to be put onto an implementation team or they're given the project as a part-time responsibility. If you want to hinder the project give it to less experienced people.

Nothing is more important than getting your software implementation right because a disastrous software implementation can crush your business and send it spiraling into oblivion.

Have no management buy-in

At some stage you're going to run into conflict. A key person needs to be in two places at once. Deadlines are going to be missed and someone with authority needs to make the decisions and needs to treat the implementation as a priority. If management is ambivalent or hasn't bought into the system, they're going to put short-term needs ahead of the longer term benefits of getting the implementation right. What an implementation needs is leadership ability rather than just management skill. It's this lack of focus and leadership ability that causes many projects to blow up in people's faces. Have no involvement from people on the coal-face. Many decisions as to how software will work are made at an executive level and this is the way it should be. But often executives are a long way from the shop floor and don't understand how things really work from an operational level. It's important to involve people who have to actually use the system and get their buy-in. Strategic decisions are important; operational realities are equally important and there's no point making a decision which is strategically correct but is operationally impossible.

3rd deadly sin - have no strategies in place to manage change

The best thing to do if you want to see money go down the drain is announce to your staff today that a new software product will be installed tomorrow and that everyone will find out in due course when the training starts. This will inspire strong resistance to change, maybe even incite a revolt! When you implement a large software product such as an ERP system you may need to make deep and far-reaching changes. If you want people to accept the change then it's vital to put change management strategies in place. How do you manage change? A clear sense of mission and purpose is essential. Give people plenty of time to adjust to the changes, get staff involved in the project, build a team, nominate team leaders and then let them do their job.

You should set flexible priorities and implement controlled procedures. Another way to ease staff into accepting change is for them to be involved in a review of business procedures and workflow processes long before the implementation. This will give them time to understand the potential benefits

of the new system and will go a long way to ensure a smooth implementation process across departments.

4th deadly sin - treat conversions as a low priority

If you want to ensure more delays and you happen to like the taste of Mylanta then don't prioritize converting data from your old system and making it readily available in the new system.

Getting your conversions right is one of the most important parts of any implementation. Your data is dirty. Everyone's data is dirty. Inevitably you will find that when you come to convert you will have alpha-numeric fields, numeric only fields and you will have things in the wrong place. This is always a problem. And it's your problem. No vendor can help you with this. It's essential that if you want to start the new system with clean data, you make cleaning and fixing your data a priority as soon as possible because you will need to do it more than once and often several times. Equally important, you want your staff to be able to train using your own data and not just demonstration data so they can use real-life examples in their testing.

A structured and comprehensive conversion process will ensure a smooth transition from your old system to the new system. Only then can you move forward as far as testing the new system and training personnel.

5th deadly sin - neglect to test the software

Test, test, test and test again! Software solutions, especially ERPs, are very complex. There is no way in the world that any vendor of a software package can test everything for you with your requirements, your people and your business processes. It is essential that you test everything as much as you possibly can in the knowledge that when you go live, it still won't be enough. And therefore not only do you need to test before you go live, you need to have a contingency plan after you go live.

6th deadly sin - have no budget for post-implementation support

The day you go live is not the end of your problems, it's the start of your problems.

No matter how well the conversions are done, no matter how well you test or how much thought you've put into it, something will come up. Someone will have forgotten an essential ingredient, something won't have been tested correctly or a customer will phone and you'll find out there's something the system can't do. It's just the way life works. This is not necessarily a disaster. You just need to plan for it. You need to put time aside after the implementation to fix the inevitable glitches and issues that arise.

You need to plan and budget for post-implementation support so your vendor can help you fix these issues and ensure your first month of operation runs smoothly.

7th deadly sin - have no personal relationship with your software vendor

It's easy to blame the software. It's easy to have a whipping boy. And sometimes vendors fit that role nicely. However, this is usually counter-productive. At the end of the day you're not just buying software, you're building a relationship with someone that will hopefully last 5-10 years or even longer.

It goes without saying that before you invest in a system, you need to assess if the software vendor is someone you can trust, someone that has an excellent support structure in place and that understands your business culture and requirements intimately – and then trust them to work with you as a partner.

Conclusion

There are a lot of horror stories about companies going bankrupt because of failed software implementations, yet many companies still don't pay enough attention to the risks involved.

Given the high software implementation failure, it's essential you manage the potential risks. One of the most important things you can do is reference checking. Talk to other companies who are currently using the package you choose and look into their experience of the implementation. Because in spite of the disasters, it's not that hard to get it right - if you avoid the 7 deadly sins.



Other takes (of many) on the Seven Deadly Sins:

1. View Panorama article online to see notes re the following seven critical areas:
<http://panorama-consulting.com/the-seven-deadly-sins-of-erp-implementations/>
 - Program Management
 - Business process and workflow definition and improvement
 - Organizational change management and communications
 - Business and technical integration
 - Globalization and localization
 - Independent oversight of technical resources [Vendor biased view]
 - ERP Benefits Realization

2. Another view:
 - #7: Denial that your Industry is Unique.
 - #6: Assume ERP can be Specific (Aim for as "Vanilla" as Possible)
 - #5: Choosing the System before Defining the Process
 - #4: Underestimating Cost & Effort
 - #3: Not Controlling "The Unknown" (aka Scope Creep)
 - #2: IT/IS Owns the Project
 - #1: Vision is too Broad