

Keep it Simple!

How listening to users can make large IT system implementations fail



By Dr. Peter Theisen, Freewill Solutions

Even though times are challenging companies should not stop innovating their ways of doing business and as advised by many management experts it's actually the best time to focus time of your key resources on business enhancements rather than purely running the day to day business.

Consequently, company leaders probably feel it's a good time to act on a long-term talked about initiative to finally replace legacy management information systems with something more modern, more integrated which will make users more efficient and deliver much better management reports. By reinvesting profits from past years of growth into new computer systems, the company probably could have something better in place within the next year and hence have a head start over the competition when the economy picks up again.

And certainly, once the initiative and the budget is approved, vendors will come in and confirm the great vision to achieve a 'perfect world' with happy users, higher process efficiency and more transparency on key business value drivers for top management.

But hold on! Didn't we hear global statistics of large scale systems implementations

which says that less than 20% of these initiatives deliver the expected ROI? Aren't there companies in Thailand that admit to have invested more than 100 million baht into a dream of perfect IT systems and experienced reduced productivity in their business operations in the first year after the systems deployment or significant time and budget overrun?

So, what do you as a business leader need to know to ensure your IT enhancement initiative delivers the expected benefits within the planned timeframe without significant interruption to the day-to-day business or budget overrun?

One key success criterion, frequently discussed in the management community is proper project and change management that should be in place for any kind of business improvement, i.e. for IT and non-IT related initiatives. Boston Consulting Group's DICE framework helps to understand specific actions that need to be taken to address proper project delivery and involvement of key stakeholders.

Another area that is critical to the success of large IT initiatives is the approach used to determine the project's functionality scope. Often user requirements and user expectations are not well

managed which leads to a heavily over-engineered system that will be complicated to implement, expensive to maintain and difficult to use.

You may have experienced it in your private life when you go out to buy a computer for private use at home. You may set your budget at ThB 30,000 before you enter the shop. While comparing different models you will suddenly find out that there are other models, certainly more expensive, that can do lots of things that you never thought about before. You're "sold"! and as a result, you end up buying a computer for ThB 45,000 with lots of features that you will probably never use.

A similar phenomena can happen in system implementation projects if user expectations are not well managed. Often software vendors sell users on beautiful system features that they never dreamed about but are "irresistible" and as a result users what it all. The important question that is seldom asked is how these "nice to have" features can add tangible benefits to the business rather than adding extensive complexity costs.

In the case of IT systems it's even worse as increasing complexity of requirements often increases costs of maintenance exponentially and

in many cases the standard functionality of your software package doesn't cover the expected scope. As a result programmers need to write extra software around your software (what we call "add on" development) which apart from extra development costs brings additional cost and effort to future upgrades.

It sounds quite paradoxical - in fact by adding more 'good features' to the scope of a systems implementation, users may end up with opposite result: The new system may come much later, be much more complicated to use, may be buggy and certainly will be more expensive than expected.

Accordingly, to succeed on a large system implementation user requirements of the first release should be kept at a

minimum. By doing this, required process changes, the associated change management efforts and the resulting systems complexity will be manageable and chances are high that first benefits can be realized within reasonable timeframes.

But how do we keep user requirements close to real business need? How do we ensure, users don't ask the vendors to implement any kind of nice to have feature which looking from the Top Management perspective is just adding marginal benefits?

One answer is to apply the principle of Selective Implementation (SImple).

Similar to the concept of Zero-Based-Budgeting SImple introduces the idea of

minimalism to the requirements (budget) definition process.

Traditionally vendors ask users "what do you need?" which ultimately leads to status quo plus whatever fancy features the new software can deliver. So you may even get legacy process waste into your new system.

The SImple approach turns the question around and challenges the users with the question "What will happen to the business (please note: not "to the user") if we don't implement the respective feature in the new system?" By using this questioning approach, people will realize that in many cases even features from existing systems are actually not adding real value to the business. Often some smart process work arounds, or little changes in policies will make this feature obsolete.

In addition to applying the simple principle it's also beneficial to optimize processes before or during the systems implementation. This will also reduce the business requirements to the system and help to keep the implementation manageable. Especially if you are running multiple businesses of the same nature, standardizing core business processes across the units or even introducing shared services may help reducing complexity through simplification and increases the chance to really obtain a much more effective systems landscape within reasonable timeframe and budget.

Client Example:

Client Challenge

A large European retail company planned to replace their existing inventory management system. They budgeted 24 million Euros. After requirements gathering from users, it was found that a leading ERP system's Retail application or other packaged software couldn't cover the full complexity of the required systems scope. As a result, the company invested 10 million Euros to license a self-developed software of another retailer. After six months in the project, the project management admitted that the budget has to be extended to more than 90 million Euros and the roll-out had to be postponed by half a year.

Top Management Intervention

Due to the significant estimated budget and time overrun, top management stopped the project and asked to review the requirements using the SImple principle.

As a result, requirements could be reduced to an extent that the standard scope of a leading ERP system's Retail application could cover more than 90% of the required functionality.

Success

Despite the required relaunch of the project with additional requirements analysis and sunk costs for software licenses, the project could be completed within the original budget and time frame.