Software is not Strategy: The Four Basic Rules of Enterprise Systems

By Michael Currie

You have heard the statistic: 70% of enterprise software projects fail. How can that be? If results are so dismal, is technology really driving productivity gains in the mining industry today? How does a business stay in business? The alarming statistic obscures something more important, that most organizations relying on enterprise software are not getting their money's worth.

Mines wrestle with an array of software solutions that include financial, operating and planning technologies that both support their businesses and satisfy regulators' reporting requirements. It is not an option to let things languish.

Is applying technology, especially on a global scale, an art or a science? It is hard to say, but organizations getting value from information systems treat them the same way as any other investment; they have to be aligned with the business and provide measurable payback. There is no magic bullet that guarantees success with enterprise software, but a certain management approach, a specific set of priorities, seems to pay off. I call these attributes the **Four Basic Rules**.

1. Put the Business First

Enterprise software is often touted as delivering strategic advantage as the basis for change and the means to transform a business. Software, however, is not strategy; while it can automate processes, boosting efficiency and productivity, it is not a substitute for conceptual thinking, business planning or managing an organization to produce results.

Companies court disaster if they try to implement enterprise software without its being part of a larger initiative to improve the business. Management differentiates successful firms from their competitors, and profoundly so when making technology investments.

A study done by McKinsey in 2004 for the European Commission analyzed whether the productivity gap between Europe and America was due to lack of investment in technology by European companies. The EC wanted to understand the effects of having less technology.

McKinsey studied a large sample of European companies and ranked their level of technology investment relative to total factor productivity gains over an eight-year period. Then he compared this ranking to a number of variables, the most interesting of which was an objective measure of "management practices."

The results were telling. Companies with low levels of both technology investment and management capability showed 0% improvement in productivity. Those with high levels of technology investment but with meager management capability gained only 2%. Well-managed companies, with low technology investment levels, still showed 8% productivity improvement, while well-managed companies that invested heavily in technology exhibited a whopping 20% gain in total factor productivity.

These results show that the approach that says, "We need to improve; software may help," works better than, "We can only improve if we buy new software."

2. You Can Show Them the Money

There is no return on investment in enterprise software, or so it would seem. Direct savings from streamlined business processes might allow companies to benefit from fewer errors, less staff and reduced duplication. These advantages, however, are usually small compared to the cost of today's implementations, so that it is very difficult to build a business case based on these gains alone.

It does not mean, however, that companies should not buy these systems. The trick is that the benefits may be indirect and more difficult to identify.

For example, a mine that upgrades its maintenance management system might expect direct benefits, such as lower inventory levels or better allocation of tradesmen, but indirect benefits, such as improved machine availability due to better planning, or lower insurance rates due to improved record keeping, will also occur and they will exceed the direct savings.

Indirect benefits are, by nature, difficult to estimate, but they are real and should be included in the justification for the project. That justification will withstand scrutiny only if all the likely financial impacts are included, if they are represented reasonably and the impact is shown not just departmentally but rolled up to the full organization P&L.

3. A Simple Plan, Violently Executed, Will Defeat a Perfect Plan Every Time

This quotation, attributed to General Patton when he defeated Rommel, is also a good strategy for an enterprise software implementation. While implementations are not exactly military campaigns, you need a plan with well-defined objectives to succeed. As in a battle, the keys to success are leadership and execution.

Leadership manifests itself first by communicating why the change is being made and how. Headship also means ensuring that the project gets the resources it requires and that real responsibility is delegated to the individuals running it, in particular to the project manager, the most important individual leader by far.

The project manager makes the day-to-day decisions about people and technology, manages to a strict schedule, understands interdependencies and navigates the gauntlet of third parties, nervous users and internal snipers. The role has to be filled by a competent insider, someone who can handle responsibility and authority. Every successful project I have seen has had one; every unsuccessful project did not.

With leadership in place, what comes next is execution. It is fair to say that organizations with a history of

successful projects tend to do well at implementing enterprise software; they have a culture that expects results.

Unfortunately many companies do not perform so well at executing large projects and a few are so mired in the status quo that overcoming it is a significant challenge. Affecting change requires a means to circumvent an inert culture. A simple plan, with clear objectives and timelines, is the best way to achieve a result. The goal for any software implementation is to offer utility to users and visibility for managers who need reliable information to make decisions. Set the bar to these achievable levels. Perfection can come later.

4. It is Never Over

At the end of an enterprise software implementation there is usually a collective sigh of relief. The vendors and consultants leave, the system is handed over to a new administrator, and everybody gets back to his or her regular job. This is just the beginning. The management would want to see results rather soon and a year later many organizations are left asking themselves, "How did it go so horribly wrong?"

Whether the initial implementation is considered a triumph or not, neglect will punish even the best-delivered systems. Ongoing success depends on three things: acceptance, data and vendor relationships.

Acceptance can be difficult to achieve right away because users tend to be skeptical about a change that many may see as having been imposed on them. The key is to acknowledge that the system is a continuum, an environment that will evolve and improve based on users' input. The best organizations formalize this process by creating an expert user group that meets regularly and, if there is one distinguishing feature about the ones that work, it is that expert users are the ones informing the system administrators how to improve the system, not the other way round.

Data sneaks up on people. Most projects require a significant conversion effort that takes data from the old system and transforms it to work in the new one, but a lot often gets lost in translation because many organizations neglect their data to the extent that it becomes virtually useless. Companies need to develop continuously and cleanse their data. In fact, data is more important than functionality. Many "failed" enterprise system projects are really data gridlock in disguise.

Finally, the vendor relationship greatly influences the long-term success of enterprise software. It can take a bit of fortitude because the sales/implementation cycle is very draining, but it is worth the effort in order to stay current with new system developments and to benefit from the community of users that exists for all major enterprise software products.

Systems disorient most organizations because they are an uncomfortable departure from day-to-day business. What works for them in the normal course gets lost in the funk of technology, change and indecision. The statistics associated with enterprise software only reflect an unsuccessful search for a process to apply it. ■

While software can automate processes, boosting efficiency and productivity, it is not a substitute for conceptual thinking, business planning or managing an organization to produce results, says Michael Currie

Michael Currie is principal of M.G. Currie & Company, which helps clients optimize investments in industrial assets and in the technologies that support them. He has led business units at PricewaterhouseCoopers and Finning (Canada). Mr. Currie's course, "Haul Truck Life Cycle Management," is published on EduMine, and he can be contacted at mgcurrie@mgcurrie.com.

Links and References

- Enterprise Software
- European Commission
- General Patton
- General Patton's Quotes
- Haul Truck Life Cycle Management
- M.G. Currie & Company
- Return on Investment and Enterprise Software
- When IT Lifts Productivity

Click here for full list of links:
http://go.mining.com/jan10-a5