



Manage to advantage

IN THOSE FRANTIC days leading up to the execution of a large software implementation deal, there is frequently a trade-off between getting a contract “right” and getting it “done”. In particular, there is inevitably a pressure to cut corners early on to achieve internal or supplier-driven deadlines.

Our experience in numerous projects shows many of these trade-offs could have been avoided if the wider issues and risks in the deal had been worked through earlier. We repeatedly see lost opportunities for greater wins — such as more services or flexibility — for the same or less price, when a little more time and focus could make a real difference.

CIOs and CFOs generally know this, but persuading stakeholders outside — and even inside — the IT team can be challenging.

Our checklist (below) is intended to paint a picture of some of the key risks and issues to be considered for a large software implementation. While it covers ground that will be familiar for many CFOs and CIOs, it can assist in helping stakeholders obtain a more holistic understanding of project. Often, once the team is aware of the key issues, a strategic approach can be taken to better secure benefits and reduce risk without unduly compromising the timetable.

What are some of the key risks?

- Historically, large scale software implementations have a relatively high incidence of project failure and/or under-delivery. They tend to be complex, time consuming and expensive.
- Once committed to the project, it is difficult to transition to a different solution. So, a key theme must be to “arrange the chess pieces” to avoid being unduly exposed when pre-deal leverage is lost.

Customers should at least prepare the implementation contract for the project.

■ Failure to design and maintain key deal constructs and leverage points across the documents and negotiations, unduly exposes the customer when problems occur. A strategic approach pays off in the long run.

Have the hidden costs been addressed?

- A fixed price is often compromised through vaguely worded and conflicting service descriptions and obligations.
- Standard vendor agreements tend to be complex and contain a number of provisions that can be used to “hook” additional licence and support fees down the track (For example, complex role descriptions and unexpected charges for access to the system by affiliates, business partners and other systems).
- Where possible, the licence agreements should be “future proofed” to reduce exposure to additional licence fees as the business evolves (a keen understanding of potential “future state” requirements is key).
- On-going costs for support, maintenance and additional licences should be locked down, along with pricing principles for out of scope work.
- Potential “blow out” points tend to include data cleansing, conversion and migration, training, customisation, change management and system integration.
- The agreement needs to be crystal clear on what the customer must deliver to ensure project success. All other resources (and the risk around inadequate due diligence) should ideally be the vendor’s responsibility.

Is there a real commitment to delivering the solution?

- Expect little if any real remedies, warranties or “skin in the game” to initially be offered by the software vendor (there are plenty of “gotchas” in the terms).
- Special care is needed to avoid the customer’s initial requirements being inappropriately superseded by vendor documentation that will be used during the project (such as the design documents and “blueprints”).
- Expect extensive obligations on the customer (and related assumptions) that will need to be tempered. For example, what will actually happen if an assumption proves to be invalid?
- There should be robust post implementation warranties and support along with ongoing support and maintenance for the underlying software.

How will the inevitable challenges/opportunities be addressed?

- Strong governance (with senior executive involvement) is essential. Top-notch project management also plays a critical part in project success. Are key vendor personnel locked in for the project?
 - Have key risk/benefit scenarios been identified and worked through?
 - Are there clear and understood mechanisms in place to detect, avoid and manage problems as they arise (For example, testing regimes, contingency plans, remedial processes, escalation, “at risk” amounts and so on)?
 - Is the commercial model structured to maintain leverage across key phases of the project?
 - Have the opportunities for wins been pursued in the early stages, before it is too late? Often wins for the customer are not at the expense of the vendor. ■
- Stuart van Rij is a consultant at Wigley & Company, a law firm specialising in ICT. Reach him at stuart.vanrij@wigleylaw.com*