Project Value Delivery

Large, Complex Project Leadership White Paper Series

White Paper 2018-04

How to Implement Information Systems in an Effective Manner in Project-Driven Organisations

In a series of White Papers, based on our extensive experience in this area, we cover challenges of Information Systems in project-driven organisations: from the overall architecture to systems implementation, development and ongoing management. In this second White Paper we cover the specific challenges of implementing commercial Information Systems. Implementation projects can be disruptive and even sometimes fail to bring value to the business. The key success factors of those implementation projects are described.

The Systems Implementation process

The implementation of an information system usually follows those process steps:

- Define the Business Requirements. It is a relatively high-level document focused on functional specifications from the perspective of the business. One of its key appendices is the checklist of functional requirements including an analysis of mandatory and 'nice to have' requirements,
- Define technically the way the new system would interface into the overall System Architecture of the organization,
- Execute the procurement process, which includes screening the market for prospective suppliers and choosing a solution after a Technical and Commercial Evaluation,
- The chosen systems supplier should then proceed through a detailed design and functional specification phase. The resulting detailed functional specifications (blueprint) will account for the specific features of the software product. Future users need to be involved in workshops during this phase,

Implementing new software systems is a major cost for organizations. It often carries substantial risk. The necessary involvement of the business can also be a major distraction.

- The software is then implemented in accordance with the defined blueprint. In parallel the affected processes are updated to align the workings of the system.
- A deployment plan including training and a stabilization phase is implemented,
- Further improvements of the product are identified, controlled through a Management of Change process, and if approved, specified, implemented and rolled-out after some usage.

Common issues and traps observed during implementation

- The Business Requirement document is too detailed or too influenced by a specific product, not leaving open the option of other, more effective ways of working
- The software scope proposed does not properly fit into the overall Systems Architecture (creating duplication or gaps), the interfaces with other software are not specified and/or fail to account for common information management features such as the organisation's coding system
- There is a lack of facilitation and coordination from the organization's information systems personnel during the detailed functional specifications workshop with the future users, leading to excessive specification of software customizations. This leads to much increased cost, risks and delays in the implementation
- The software provider team is not local, or unexperienced with project-driven businesses, creating communication issues and lack of participation to workshops
- The ambition of the first release of the software in the organization is too large in terms of functionalities and coverage
- There is a failure to conduct a pilot application on a small scale to iron out bugs before spreading the usage across the entire business

Higher level best practices for systems implementation

Taking a step back, Information Systems implementation needs to be taken carefully since such substantial Capital Expenditure projects can have a massive impact in the business effectiveness over many years. Even when the software expenditure is limited, for certain essential tools, the cost of lost time and effort and the cost of lost productivity, can be substantial if the implementation fails to deliver the expected value

Some of the following recommendations may appear counter-intuitive and are definitely not consistent with the advice of most software implementers. Still they have proven their effectiveness.

Implement first the most standard version possible of the software

For a brand-new software, implement first a standard version (with relevant interfaces setup). It will be much more effective to specify adaptations after experience with the tool has been gained by the users. This practice also reduces drastically the lead time of implementation. Experience shows that up to half of the customisation effort that would be requested upfront can be avoided when people get accustomed to the software;

specification of customisations are then much more to the point.

If such an approach is not possible, at least implement an extensive pilot program with users getting a good grip on a standard version of the software before launching carefully considered customisation specification

Design an implementation team driven by the business

The implementation team should be carefully chosen and driven by a Project Manager that is more business than IT oriented. The participants from key departments and divisions should be full-time and empowered to take the right decisions regarding the systems implementation.

Do not force functionalities upon a software that is not initially designed for. Prefer niche software

The effort to build into an existing software features and functions it is not designed for initially is always under-estimated. Of course, it is a great opportunity for the software supplier as it gets paid for developing new functionalities. The software supplier is unlikely to complain and may even encourage the client to take such a route. However, this invariably leads to convoluted development projects with long lead time and a lot of debugging. In addition, the final solution will be likely to be much less suitable to the task than a more specialized software. Still we are always amazed at how often organizations try to make software do things for which it is not designed for. It is a bit like using the wrong tool for the task: it can work but is more dangerous and definitely not effective.

Some of our recommendations may appear counter-intuitive and are definitely not consistent with the advice of most software implementers. Still they have proven their effectiveness.

Avoid too much customisation that will impede version update

We encounter too many organisations that run on antiquated systems that cannot be updated simply because they have implemented too many customisations. Any update of the underlying version would mean updating all the customisation, which would be very costly. As a result, it is not rare to see organizations running on 10 to 15-year-old software that is not even web-based, ruining effectiveness in the current world. This syndrome absolutely needs to be avoided, which means that customisation should be kept to the absolute minimum. This links back to favoring specialised software that is fit for the task, even if that means a bit more thought on the data structure to be used across the business and perhaps adapting the ways of working slightly to suit the tool.

Be careful with automated interfaces

Automated interfaces are fine as long as the quality of the data is guaranteed. The risk with automated interfaces is to have unreliable data spread throughout the entire set of systems. We find that it is often preferable to have semi-automated interfaces (including a manual check of data), without impeding effectiveness.

Also, while automated interfaces in one direction are often fine, interfaces that work both ways can be riskier and should be minimised. The overall system architecture needs to take this into account.

Never forget that an Information Systems has an ongoing cost (external and internal)

All too often decision makers focus on the investment cost without realizing that any Information Systems has an ongoing operational cost. Annual maintenance fees are generally charged of 20-25% of the initial software license cost. In case of heavy customisation there will also be recurrent substantial version upgrades projects.

But what is often overlooked is the need to have an organisation to administer the software on a day-to-day basis: resolving smaller user issues, cleansing the data on a regular basis to ensure its integrity, producing performance reports for the business, setting up new projects, etc. Too often organizations spend money on software without planning for the administration position. The future administrator must also be mobilised early during the implementation project.

Conclusion

Implementing new software systems is a major cost for organizations. It often carries substantial riskand requires substantial involvement of the business. This White Paper describes best practices which we believe carry substantial opportunities for organisations.

Proper leadership of such projects is required, insisting on minimum customisation, selection of proper solutions that fit the business needs and the overall Systems Architecture. This is the role of strong Information Systems management resources, which are too often lacking in organisations.

.