The Keys to a Successful Software Implementation—Part Two

Implementing even the most complicated budgeting system can be simplified with proper planning and understanding of several key factors. Addressing each of them will facilitate an implementation without surprises, and provide a healthy foundation on which to build your long term strategic budget goals.

Last month we discussed the importance of **establishing clear objectives**, **setting a realistic timeline**, and **ensuring the availability of resources**.

In this issue, we cover: **Customizations, Change Management**, and **Checks & Balances.**

System Implementations - Customizations

Most system applications are theoretically developed based on industry best practices and are intended to be deployed "as is". Most vendors offer configuration options that allow organizations to incorporate their own business rules but sometimes there remains functionality gaps after the configuration is complete.

Customers can use technical and non-technical solutions to reconcile functionality gaps by weighing the pros and cons of each. Technical solutions include rewriting part of the delivered functionality or development of new or additional functionality. These options are varying degrees of system customization which may be invasive and/or costly to maintain.

Alternatively, there are non-technical options such as changing business practices and/or organizational policies to better match the delivered functionality.

It is important to understand the difference between system configuration and customization. Software almost always has to be configured before use. This may include items such as setting up the organizational structure, costing centres, general ledger accounts, configurable screens, etc. Most software is designed to handle various configurations and behaves predictably in any allowed configuration.

Customization is always optional. It behaves less predictably, increases testing requirements by both the vendor and customer and may not survive upgrades to new software versions. While customizations may improve user acceptance, it does increase the time and resources required to both implement and maintain them. As a result, both your project timeline and budget may be impacted.

> **"That is how we have always done it" is not a reason to customize. Examine the process and look at best practices.**

When deciding what to customize, there are several considerations. Is the functionality a "need" or is it a "want". If you can live without it, perhaps it should be parked and scoped out as part of a secondary phase. "That is how we have always done it" is not a reason to customize.

Examine the process and look at best practices. Perhaps this is an opportunity to improve or re-engineer the process. Consider whether the customizations can be done within the implementation timeframe.

If you do decide to customize, make sure all of the stakeholders are involved in the process. Specifications need to be clearly documented and understood by both your team and the development team.

Many users are not able to easily express their requirements since quite often they simply don't fully understand what they want. In addition, end users generally don't have the technical knowledge to understand the implications of what they are demanding. This can lead not only to increased costs, but also be detrimental to the success of the project if the expected outcome is not achieved.

Make sure you have a mechanism in place to keep a tight control over end user requirements and to assess the resources (time, money and manpower) that each modification will require. In order for the project to be successful, you need to strike the right balance.

Change Management

Change management is a structured approach that uses tools and techniques to manage the people side of change. The goal of change management is to transition people (individuals, teams or organizations) from an existing state to a desired future state. This is done by helping them to accept and embrace change by reducing and managing resistance to process, technological or organizational changes.

A solid change management plan is one of the major keys to a successful system implementation, and should contain three phases:

Preparing for Change

a. Readiness: Assess the organization's readiness to change and the scope of the change. Is it a big, radical, or gradual change? How many people are affected? What type of resistance can be expected?

b. Communication: Prepare a communication plan that starts by building awareness of the need for change, creates the desire for change and communicates the risks of not changing. Consider your audience, and their differing needs depending on their role in the change. Timing is everything. Be sure to share the right messages at the right time.

c. Sponsorship/Buy-In: Sponsorship and buy-in are critical success factors. Sponsorship requires active and visible participation by the senior business leaders such as the City Manager, Treasurer and/or Director of Finance throughout the project. Buy-in is achieved through getting on board as they have the most direct influence over the employees affected by the change.

Managing Change

a. Resistance: Although resistance from employees and managers is normal, persistent resistance can threaten the success of a project. Identify and understand the source of resistance and create a plan to manage it. Avoid the loss of valued employees by asking for their input and incorporating their ideas where possible.

b. Involvement: Managing change is not a one way street, and employee involvement is a necessary and integral part of the process. Analysis and corrective action based on their feedback provides a robust cycle for implementing change.

Reinforcing Change:

a. Celebrate: Recognize and celebrate successes, big or small. Individual and group recognition is also a necessary component of change management to cement and reinforce the change in the organization.

b. Review: The final step in the change management process is to review the project, evaluate successes and failures, and identify process changes for the next project. This is part of the ongoing, continuous improvement of change management for your organization and ultimately leads to change competency.

Checks & Balances

We've talked about establishing clear objectives and realistic timelines and these are very important elements of a successful implementation. However, in every really successful project feedback dominates; so let it.

A successful project is not a project that delivers all of the initial tasks on time and within budget. Rather, it is a project that meets its business objectives at the time of delivery. These, like everything in life can change, so make sure that your project plan can accommodate changes.

Try to avoid preparing a project plan just to meet an administrative requirement that "a Microsoft Project template must be completed." This invariably leads to the invention of tasks long before anyone has any real understanding of the environment, the requirements, or the capabilities of the technology to be used. It may seem odd, but as a software vendor we see this all the time.

Beware of the propensity for groups of human beings to agree to concepts that make absolutely no sense. A great example of this is when purchasing departments insist that vendors supply a project plan as part of the RFP process.

Finally, don't wait until the end before checking that your system is delivering accurate results. Make data reconciliation an ongoing process.

In every really successful project feedback dominates; so let it.

Not quite finally!

Over these last two issues we have isolated a few ideas that collectively we think will contribute to a successful project. To a large extent they are our personal hobby horses, and you should feel free to agree or otherwise.

You may have your own gems of wisdom from which others may benefit. If this is the case we encourage you to send us your insight at education@questica.com.