Five Key Elements for a Successful Systems Implementation

The implementation of a new or upgraded system in your distribution operations can be a daunting task. Many things can lead a system project off track. To prevent this, consider five key project components that have proven effective in providing successful outcomes to these implementations.

1. Upper Management Support

Upper management is always involved in the financial support of a systems project. Some executives focus only on the financial justification and the payback of the project. However, the key to a successful implementation is to also convince them of the business advantage of a systems change.

Success of the project depends on the executive team believing that the new system will provide the company with a new competitive advantage to service customers better. When that belief is in place, executive sponsors can champion the project when it hits its inevitable rough spots. They can effectively allocate additional resources to shore up tasks falling behind, arbitrate differences in a timely manner and reinforce expectations during conversion. They can spread the energy and excitement about what this change will mean to all involved.

2. Operations Leadership

There should be a broad representation of groups in the project to ensure that all perspectives are accounted for during the design and implementation. However, operations staff should lead the project. Whether it is the warehouse staff or shipping department, the project should be led by the people who will actually use it to improve their job performance.

Two groups who tend to vie for project leadership are IT and finance. Although these teams are critical to a successful project outcome, in the end, the changes that are being made will most directly affect the people using the software on a daily basis. If operations leads the project, the ownership will increase and so will the project’s success.

3. Testing

To make sure that the system is functioning as designed, it must be thoroughly tested. A test plan should be detailed as soon as designs are completed, and testing resources should be focused on the main functionality of the system. Avoid the common error of focusing too much of your testing efforts on complex functionality that may only be used a small percentage of the time.

Where possible, utilize end users to do the testing. They will learn valuable troubleshooting skills and, more importantly become advocates of the conversion. Make sure that the functionality used everyday is working.

For all testing, load the software onto the machines which will be utilized in production. This allows communication and architecture issues to be resolved in a test environment, as opposed to a production environment on cutover day.

Track all issues and their corrective action status. Do not lose sight of status of the overall importance of a particular issue. Set your threshold for acceptance based on percentages of classes of test cases passing. For example, 90% of B priority test cases pass.

And remember, testing does not prove that software is free from errors; only that it performs as expected.

4. Training

After the system functionality passes operational testing, final training can begin. Ideally, users have been given some overview classes prior to actual hands-on training. The hands-on end user training is most effective when delivered as close to the cutover as feasible, in order to achieve optimal information retention. Again, the focus should be on key everyday functionality. Operators should be comfortable performing everyday functions flawlessly.

Laminated 'cheat sheets' can be given to users, and will serve as a tremendous memory refresher to use during conversion. During startup, the team members responsible for troubleshooting should be focused on handling the inevitable exceptions or scenarios missed during testing. If they have to perform on-the-job training for everyday functions, errors will be made, frustration will increase and overall risk increases.

5. Metrics

The project was likely justified because it would make improvement to processes. Utilizing points of comparison can provide valuable tools in diagnosing problem areas, so it is necessary to validate the baseline metrics well before conversion. It is important to keep the number of items tracked small, simple and based on the overall impact of the metric. For example, measuring full case pick rates in a primarily loose piece environment is a poor choice.

Examining specific metrics that are not meeting expectations can help focus corrective efforts. Efforts can be as simple as further training clarification or as disruptive as a system fix. Deficient metrics can also point to operators struggling to meet accuracy expectations due to system or operational constraints. Without a valid baseline for comparison you will be unable to ascertain the level of success.

Many events can reduce your chance of successfully implementing a system, which is why it is so important to carefully consider the roles that upper management, operations leadership, testing, training, and metrics play in your project. Manage the project with a keen focus on these key areas, and you will greatly increase the project's chance of success.