Part IV

Address Specific Situations and Issues

Chapter 15

Implement New Technology and Systems

INTRODUCTION

A common way to achieve change is through implementation of new systems and technology. Yet, many systems implementations have met with failure to deliver the estimated benefits. In many cases the systems have been installed, the people trained in the use of the system, and success in implementation proclaimed. Then this is followed by resistance and lack of use of the new system. In this chapter we explore how change management can be employed as a tool alongside and mutually supporting systems implementation. When you stop and think about it, IT and change management are often closely linked.

Persistent and lasting change is often best supported through automation; getting results from investment in IT often requires change in the work and processes.

Automation tends to standardize the work and makes it more difficult to change by the employees. Hence, it is more stable. On the other hand, if people do not use the system properly or fail to use the system, then there really is no change. In fact, things can get worse and result in lower productivity. Benefits of automation only come through change.

If you carry out change that does not involve systems and automation, then the stability and persistence of the new process and methods depends upon the people consistently and constantly supporting it. This requires more training and supervision. Change is reinforced by automation.

SYSTEMS AND TECHNOLOGY ISSUES

There are a number of systems and technology issues that are relevant to change management. These include the following:

- Many existing systems are old and fail to meet changing needs of the business. This results in the employees in business units having to invent workarounds and shadow systems as well as to generate exceptions.
- Systems, like processes, deteriorate over time. The program code becomes more complex to maintain and enhance due to multiple programmers working on the system as well as increased size and number of modules.
- Software packages (off-the-shelf software) has great appeal since managers envision that it can be used with little change. However, many packages are relatively inflexible and cannot be substantially customized. Moreover, a number of software packages are old and so have even greater inflexibility.
- IT looks at systems for a department or process and sees the existing system so that replacement in their eyes focuses on a one-for-one replacement. However, it is not that simple. From the business employee view, the situation is quite different. They see the system, shadow systems that they built, workarounds, and manual exceptions. They have become comfortable with this mélange of systems.

FOUR MYTHS ABOUT SYSTEMS IMPLEMENTATION

There are so many myths that you can identify given what we have covered in change management.

- *Myth 1*: Requirements for a new system can be defined one time and are stable.
  This is fundamentally flawed in most cases. As you have seen, requirements change as the business needs change. If you implement a system that takes a long elapsed time, then the system is not likely to meet all of the requirements of the new situation.
- *Myth 2*: It is necessary to understand all of the requirements for a new system.
  This has been shown to be false. First, you cannot identify all requirements since there are too many exceptions. You cannot possibly automate everything. You have to draw the line somewhere.
- *Myth 3*: User acceptance occurs after the training and installation of the system are completed.

This is often much too late. There may be a major buildup of resentment and resistance. Moreover, users often resist accepting a new system since it means that they have to make changes in their processes and become more accountable for the benefits. User acceptance must be gained even before the system is installed through user dissatisfaction and unhappiness with the current system and situation.

- *Myth 4*: User participation in systems implementation can be limited. This is a sure recipe for failure. If employees do not have to be involved, they do not participate. They are remote from the new system and the effort. They will not become committed to change. They assume no ownership.

WHY SYSTEMS IMPLEMENTATIONS OFTEN FAIL

Here are some key reasons why so many system implementations fail.

- The project to implement the new system ends when the system is installed. However, there are additional steps. The business process must be changed. Also, the benefits must be measured. Without these additional steps, the project ends and the business employees are under less pressure to change their processes.
- The scope of the systems effort is limited to the existing system. Shadow systems and exceptions may often be ignored. Then the system will not meet all of the requirements of the business. Furthermore, employees in the business departments may have to invent new shadow systems to match up to the new system.
- The system project relies too much on “king bees” and “queen bees.” As was pointed out earlier, these people like the status quo and resist change.
- IT selects new technology to implement with the new process. However, there is long learning and expertise curve and time to be endured. This can doom the system implementation.
- IT staff often want to please the employees so they collect requirements from interviews as to what is desired. This is often different from what is needed. The result may not only be extra effort, but a more complex system than is really needed.
- IT staff pay too much attention to exceptions so that enormous programming effort can be expended on automating relatively rare and infrequent transactions.
- Many business staff want the new system to do the work the same way the old system did the work. Hence, they will make requests and issue requirements that tend to warp the new system back to the old. This is natural since they have worked with the old process and system for years and are loath to give it up.

A MODERN APPROACH FOR IMPLEMENTING NEW SYSTEMS

Change management and systems implementation need to be more tightly integrated. We have seen above that each often depends upon the other. The linking of these two has ramifications for how the systems implementation is planned, organized, and undertaken. From experience here are some guidelines.

- Ensure that the major scope of the system implementation is to change the business process. Within this policy and procedure change, staffing, and systems fall as components.
- Early in the systems project, aim to kill off or eliminate exceptions. This will reduce the elapsed time for development. It will also reduce any customization of the software if a package is selected.
- In the early analysis work to define requirements, identify the shadow systems. You need to understand why these systems were created and what situations they are intended to address. After all, these were created after the old system was implemented so that the shadow systems meet new requirements.
- When you are collecting requirements, it is important to define the new process through transactions as well as to identify problems with the current process and system. Requirements and benefits can then be defined. Business employees must be heavily involved in these activities for several reasons. First, they see the problems with the current process and so will be more supportive of change. Second, business employees can define detailed benefits in more detail than IT employees.
- Since the systems work will take a substantial amount of time to carry out, implement the Quick Hits described earlier in parallel. Why is this a good idea? First, you pave the way for the new system. Second, you can address political resistance separate from the system being installed. Third, you raise morale and support for the new system and change.
- Traditionally, much of the documentation of the system including procedures and training materials were prepared by IT staff. This often runs into trouble since the language of procedures does not match that of the users. Moreover, the user procedures do not encompass the entire business process. There are often significant gaps and holes. Therefore, a better approach is to embed the user procedures into the operations procedures for the business process. Training materials and operations procedures should be written by the employees in the business with the help of IT staff. This will help them assume ownership.
- Training in a new system is most frequently performed by IT. The training scope is restricted to the system. This is also flawed. Consider a new employee who comes into the department 6 months after the new system

is in place. The person needs to be trained in the process that includes the system. This is best done by the business department not IT. IT, in fact, may not even know that the new employee is there. The guideline here is that recurring training should be done by the business users. The scope of training should include the process as well as the system. Thus, training in the new system initially should be carried out by the users with IT support.

There are a number of impacts when you combine change management and system implementation using the approach above. First, the business employees have a wider role and more responsibilities than is the case traditionally. Second, the employees are more likely to be supportive of change and will work harder to ensure that the process is changed.

ALTERNATIVE IMPLEMENTATIONS OF NEW SYSTEMS

After you define the new process, you are in a position to consider a wide range of automation alternatives.

- Live with the current system and change the process around the existing system. You can do this through changing policies and procedures. This greatly lowers the cost of change and vastly reduces the scope of the effort.
- Change or enhance the current system to handle some of the exceptions or shadow systems. This also is less effort than development or software package purchase. An example might be to add a web based front end to an existing system.
- Undertake new development while trying to salvage the useful part of the current system.
- Undertake massive new development.
- Acquire a software package for the new process.
- Adopt a combination of development and a software package.

Often, these alternatives are not all considered. Only new development and software packages are considered. This is a big mistake and often results in extra work.

You should also note that the fundamental purpose changes from getting a new system through development or purchase to trying to find the best solution that will be most affordable and consume the least amount of time. In general,

To minimize risk, you seek to avoid wholesale replacement of the current system unless absolutely necessary.

There are also additional guidelines for both development and software package acquisition. Figure 15.1 provides guidance for software development. Figure 15.2

Guideline	Comment
Avoid undertaking projects with technology you are not familiar	This raises the risk and unpredictability of the work. It also will more likely alarm users.
In developing a prototype focus on carrying out specific transactions rather than on entire user interfaces	This is in line with the approach of dealing with where the risk is
Pay much more attention to system interfaces early	This helps to reduce the risk
Select programming tools that minimize new written code	This is in line with the goal to minimize work
Try to implement the new system for some transactions as soon as possible rather than waiting for completion	This will tend to get more political support for the new system

Figure 15.1 Guidelines for Software Development

Guideline	Comment
Select software that covers the most functions not features	Functions are necessary capabilities. Features are often things that are nice to have.
Try to avoid any substantial customization of the system	Customization often ends up warping the new system to be the old. Customization increases time, risk, and cost.
Select any consultant who will help in the package implementation at the same time that you are selecting the package	The consultant will turn out to be as important as the system in many cases
Aim at a minimal implementation of a software package	Avoid implementing all parts of a package at the same time to reduce risk

Figure 15.2 Guidelines for Software Packages

provides guidance for software packages. For packages, it is important to distinguish between features and functions. Functions are basic capabilities. If a function is missing, then you have to invent, customize, or develop software for this missing capability. Features are things that are nice to have. A number of these are not critical. An example on a new car is cruise control—nice to have but not essential. Figure 15.3 shows the features of two packages A and B. Traditional evaluations would have you select A because it has many more features. Yet, consider Figure 15.4. Here package B has more functions than A. The outer box is the total required functions. The lesson learned is:

Select the package that covers the functions best so as to minimize extra work and time.

Now for whatever method you select, you are faced with the situation that one selected, or in fact all of them, will not lead to a perfect fit with the new process and work as defined. There is almost always some gap between the real and ideal.

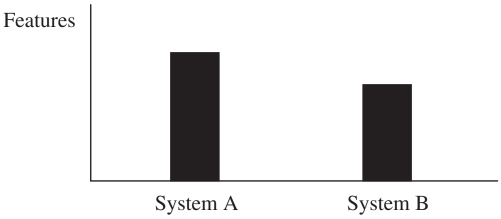

Figure 15.3 Comparison of Two Systems in Terms of Features

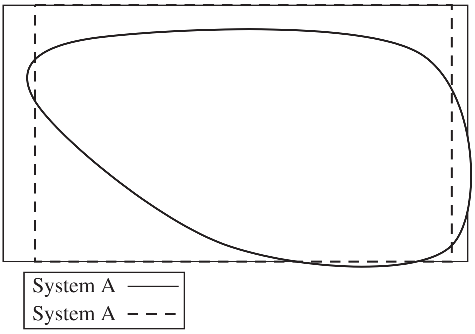

Figure 15.4 Comparison of Two Systems in Terms of Functions

Therefore, the business department and IT have to become engaged in trade-offs as to what effort to expend to approach the ideal closer.

RISK IN SYSTEMS AND TECHNOLOGY IMPLEMENTATION

Experience has indicated that the following areas of system implementation have high risk.

AREA OF RISK: END USER INVOLVEMENT

You should have continuous business staff involvement. What can the users be expected to do? A list for the business staff appears in Figure 15.5. Note that IT staff will have to participate in these tasks too.

This is a powerful and lengthy list of things to do. What if the business managers balk at having some of these things performed by their staff. They can use many excuses. Here are some common ones.

- Analyze the current business process and work
- Uncover issues and problems in the current work
- Define the new business process
- Help to define Quick Hits from this analysis
- Help in defining requirements
- Define the benefits and how they will be measured
- Participate in design and development
- Be involved in software package evaluation and selection
- Participate in data conversion work
- Be involved and support policy and procedure changes and other Quick Hit activities
- Support measurement of Quick Hits
- Participate in testing
- Develop the training materials
- Develop the procedures for the new process and system
- Train the business staff in the new process and system
- Support the cut-over to the new process and system
- Eliminate the old process and its vestiges including shadow systems and exceptions and workarounds
- Change the process to fit the new system and work
- Measure the benefits after change and implementation
- Provide on-going training for new employees in the process and system
- Support on-going measurement of the process and work
- Help to prevent process deterioration

Figure 15.5 Activities in System Implementation Suitable for Business Staff

- There are no employees available. Then the new process is not a priority. Why are we doing the work then?
- We do not know how to do this as we have not done it before. IT can help and the users can participate.
- It is mostly IT work. Wrong! It is almost entirely business work that needs some limited IT involvement.

None of these are really valid. After all, if the business wants the system and new process, they have to be willing to put "skin in the game" or be part of the effort. They cannot just be spectators. They have to be participants. One reason for this is that these tasks are not appropriate for IT to do. They don't have the knowledge or experience of the business staff. They do not know their terminology. They may produce work that is not complete or acceptable.

AREA OF RISK: REQUIREMENTS DEFINITION AND PROJECT SCOPE

Scope creep and unstable requirements are big problems in almost all major systems efforts. Prevention of these things requires that there be constant user involvement. The user activities defined in Figure 15.5 provide for this.

Another method for reducing these risks is to revisit the business process and work to see if there have been changes after the requirements have been gathered. This will help keep the people involved in implementing the new process and system in touch with the people doing the work.

A third method is to ask the following questions when someone wants to make a change to requirements or add to the scope. Answering these questions will dissuade people from changes. Here is a tip. Point out that these questions will be asked every time that there is a change early in the implementation effort.

- What is the change needed?
- What are the benefits of the change?
- Why did this change surface now? Why was it not detected earlier?
- How would the benefits be measured?
- What is the requestor willing to do to support the implementation of the change?
- If there are multiple requests, maybe the entire effort should be stopped and the requirements revisited since the entire situation has changed substantially?
- What if the change is not done? What will the business unit do?
- What if the change is deferred? What will the business unit do?

As you can see, these are very pragmatic questions that deserve answers.

AREA OF RISK: GETTING THE BUSINESS RULES

Business rules are the detailed directions for how specific pieces of work or transactions are handled. As such, their understanding is key to whether the new system and process meet the requirements of the business and deliver the benefits.

Where are the business rules? In the process. However, many are in the program code of the existing, old system. Here is a tip. Begin to gather business rules from the IT programmers and staff who support the old system. They tend to be more familiar with them than most of the business staff since they have to know their programs. Once business rules are programmed, many users then just assume they are there and do their other work. After you have exhausted the IT staff, then go to the “king bees” and “queen bees.” These people are now useful here in that they can provide the business rules as well as how and why they were created. This is your best use of these people.

Area of Risk: Process Documentation and Training Materials

It has already been pointed out that these are properly the domain of the business staff. IT employees can work with the business staff to get these documents started. Here are some additional guidelines.

- Assemble and organize documents from past IT projects and implementations that can be used as models. There is nothing wrong copying or following what has worked before if it is relevant.
- Have people develop procedures using the method of successive refinement. That is, the employees develop successively more detailed outlines. This will lead to reduced risk since you know the status of the documentation at any one time.
- Have other business staff review end products for language, tone, politics, and content. These are the best people to do reviews.

Area of Risk: System Interfaces and Integration

This is an area of major concern. Most new systems do not operate in a vacuum. They have to interface to surviving parts of the old system or to other systems. Some of the problems with interfaces are:

- Systems change over time so that interfaces have to be monitored and maintained.
- Systems were written at different times by different people so that there are likely to be differences in meaning, format, creation, and other attributes of data elements.

System interfaces have to be designed in terms of content, timing, frequency, validation of interface, backup if there is a problem, recovery if there is a problem, and format. Thus, you want to gather interface information early in requirements.

Similar comments apply to system integration. Design of integration of systems and how to test the integration are key ingredients to systems success. This is so important that system integration should be pulled out as a separate subproject.

Area of Risk: Data Conversion

Converting data from the old system to the new has been a real curse and problem for us over the years. Some of the problems are:

- There is missing data
- The data elements of the new system are more comprehensive than that of the old

- The data elements of the new system have meanings different from that of the old
- The data in the old system is of questionable validity and accuracy
- Data quality is bad in the old system

As with interfaces, you want to start analyzing the current data early. Then you want to map in to the data elements of the new system. You also have to make provision for data cleanup. In more than one systems effort, this was ignored and the result was that the entire system implementation was held up while the data was cleaned up for conversion.

There are some critical activities and areas to address in data conversion, including:

- What is the quality and nature of the current data?
- How does the data in the old system map into or relate to that in the new system?
- What data is missing?
- What will be done about the missing data? There are several options: live without it, add it to the old system and then convert it, or add it to the new system.
- What will be the conversion approach?
- What is the timing of the conversion? If you convert too early, then the data in the old system that is still in production has changed.

AREA OF RISK: USER ACCEPTANCE OF CHANGE

In traditional system implementation user acceptance of change is a milestone left to the end. The dream is that people who had resisted change will see the new system. The light will then come on. Then the users will wholeheartedly endorse the new process and system. In your dreams!

A more realistic approach is to get as many different users involved in the system implementation. Also, you want the users to acknowledge the problems in the current process and system. Then they can be involved in the implementation of Quick Hits. With these steps you achieve user acceptance. User acceptance does not come overnight.

Moreover, just because a business manager accepts the system does not mean that the lower level business staff do. They may just continue to do things the old way even after implementation and acceptance. This brings up the major questions of “what is acceptance?” and “what does acceptance mean if the lower level users do not accept it?”

Area of Risk: Benefits Attainment

Attaining benefits is a major concern that has been pointed out in this and previous chapters. The ingredients of achieving benefits are the following:

- Initial definition of benefits
- Definition of how benefits will be measured
- Determination of what will happen when the benefits are achieved
- Implementation of the new process as well as the new system
- Measurement of the actual benefit
- Decision on what to do with the benefits

These are important. Just because you get benefits, if you do nothing with them, then there are really no benefits.

Another guideline is that all benefits must be translated into tangible benefits. That is, you should not allow fuzzy benefits. Systems projects are often cursed with fuzzy benefits. Let's take an example of how to do the conversion from fuzzy to tangible. Suppose that the new system is much easier to use and is more "user friendly." What does this mean in the real world? Training time should be less. Documentation should be simpler and faster to develop. There should be greater throughput of work. The time to do the work may be less. These are all tangible.

Now remember our discussion of benefits for the new process. You measured not just the new process, but also what would happen if the old process were to continue to live. There would be more deterioration. Keep this in mind when measuring benefits.

Area Risk: Process Measurement

Many organizations implement new systems and then perform a post-implementation review. If this is successful and the business unit is not unhappy, measurement often stops. There is no provision of on-going measurement in IT systems implementation. Big mistake! Remember that the system and process can deteriorate individually and collectively. Thus, there must be the on-going process measurement that was discussed in Chapters 13 and 14.

QUICK HITS FOR TECHNOLOGY IMPLEMENTATION

There are some Quick Hits that apply to systems and technology implementation over and above the process oriented ones that were mentioned earlier. Here are some common ones.

- Enhance or fix some of the problems in the current system if this does not consume too much effort. This can buy time for the long-term system solution.
- Improve or upgrade PC's or servers so that the response time using the old system is better. This is not wasted since the new system may require these upgrades anyhow.
- Improve the network so that there is greater capacity and improved performance.
- Retrain users in how to use the old system properly.
- Make a shadow system more standardized and provide support for it.

These things may seem like wasted effort since you are implementing a new system. Also, you may not want to divert resources from the new system implementation to these tasks. However, these negatives can be offset by the benefits that are provided as well as the time bought for new system implementation.

UTILIZE THE SYSTEMS AND TECHNOLOGY SCORE CARD

Figure 15.6 displays the systems and technology score card for implementing new systems and technology. The following comments apply to the score card components:

- Involvement of business staff in analyzing existing system and process—the more and greater the involvement, the more likely will be the support for change.
- Involvement of business staff in benefits estimation—this is critical since the business staff will be analyzing the benefits later.
- Number of exceptions to be included in new system—the more exceptions that are included in implementation normally means trouble since the elapsed time will be longer for implementation.
- Extent of replacement of shadow systems by the new system—the more functions of shadow systems that are replaced by the new system the better.
- Availability of performance data for the process through the new system—having the new system provide automated performance information helps in on-going measurement of the process and deterioration.
- Elapsed time of system implementation—this is critical; the longer the implementation the more the requirements change, the less confidence the users have in the new system, and the more chance of erosion of management support.
- Amount and extent of new technology employed—from before the more new technology that is used, the greater the risk due to the learning curve.

Factor	Score	Comments
Involvement of business staff in analyzing existing system and process		
Involvement of business staff in benefits estimation		
Number of exceptions to be included in new system		
Extent of replacement of shadow systems by the new system		
Availability of performance data for the process through the new system		
Elapsed time of system implementation		
Amount and extent of new technology employed		
Involvement of business staff in documentation and training		
Number of business staff involved in implementation; percentage of total staff involved		
Stability of scope and extent of surprises during implementation		

Figure 15.6 Score Card for Systems and Technology Implementation

- Involvement of business staff in documentation and training.
- Number of business staff involved in implementation; percentage of total staff involved.
- Stability of scope and extent of surprises during implementation.

EXAMPLES

ROCKWOOD COUNTY

Rockwood employed the traditional approach for system implementation. As a result many systems efforts resulted in little or no benefits. It got worse. Process improvement and change management efforts were addressed outside of IT systems. This meant that many changes that were implemented were neither persistent nor long lasting. In some cases, the process improvement effort resulted in new shadow systems making the overall situation worse.

The situation was only improved when a new management policy was put forth. It required that all significant IT efforts had to be linked to process improvement and change management. After this things greatly improved. However, there were still efforts to minimize systems efforts so that some could avoid the process and work change.

Legend Manufacturing

Legend had experienced two major systems failures where the systems were completed at great cost and yet the benefits were not achieved. They implemented the same rule as Rockwood County. When the change effort started, this was reinforced. All shadow systems were tabulated and analyzed. More useful requirements were found here than in some parts of the old systems. Legend implemented different solutions depending upon the conditions: enhance the current system, buy a package, develop modules for the current system, and develop a new system.

LESSONS LEARNED

- Long-term change often depends upon systems being improved or implemented. The pace of implementing long-term change often lags due to the time required for software change or installation.
- The details of a business process and the transactions must be linked closely to the automation. If the automation effort leaves significant manual work and steps, then there is higher likelihood of reversion and process deterioration.
- In terms of user involvement, you must have as many different users involved as possible. This will increase support for change and for the new process and system.

SUMMARY

System implementations often fail or run into trouble because there is a lack of change in management elements in the implementation planning, organization, and execution. Specific steps that you can take to reduce risk and problems in system implementation come directly from the concepts of change management. To be successful in change management, automation provides stability over time. To be successful in system implementation, you have to implement changes to processes and work. Change management and system implementation are intertwined.