

# CHAPTER 13

## TRANSITION TO THE NEW SYSTEM

| IMPLEMENTATION    |                                                             |
|-------------------|-------------------------------------------------------------|
| TASK<br>CHECKLIST | <input checked="" type="checkbox"/> Program System.         |
|                   | <input checked="" type="checkbox"/> Test Software.          |
|                   | <input checked="" type="checkbox"/> Test Performance.       |
|                   | <input type="checkbox"/> Select System Conversion Strategy. |
|                   | <input type="checkbox"/> Train Users.                       |
|                   | <input type="checkbox"/> Select Support.                    |
|                   | <input type="checkbox"/> Maintain System.                   |
|                   | <input type="checkbox"/> Assess Project.                    |
|                   | <input type="checkbox"/> Conduct Post-Implementation Audit. |

This chapter examines the activities needed to install the information system and successfully convert the organization to using it. It also discusses postimplementation activities, such as system support, system maintenance, and project assessment. Installing the system and making it available for use from a technical perspective is relatively straightforward. However, the training and organizational issues surrounding the installation are more complex and challenging because they focus on people, not computers.

### OBJECTIVES

- Explain the system installation process.
- Describe the elements of a migration plan.
- Explain different types of conversion strategies and when to use them.
- Describe several techniques for managing change.
- Outline postinstallation processes.

### INTRODUCTION

*It must be remembered that there is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than the creation of a new system. For the initiator has the animosity of all who would profit by the preservation of the old institution and merely lukewarm defenders in those who would gain by the new.*

—Machiavelli, *The Prince*, 1513

Although written nearly 500 years ago, Machiavelli's comments are still true today. Managing the change to a new system—whether or not it is computerized—is one of the most difficult tasks in any organization. There are business issues, technical issues, and people issues that must be addressed in order to prepare for and successfully adapt to the change. Because of these challenges, planning for the transition from old to new systems begins while the programmers are still developing the software. Leaving this planning to the last minute is a recipe for failure.

This chapter discusses the transition from the as-is system to the to-be system and ways to successfully manage this process. The migration plan encompasses activities that will be performed to prepare for the technical and business transition, and to prepare the people for the transition. We also present several important support and follow-up activities that should be performed following the installation of the new system.

## MAKING THE TRANSITION TO THE NEW SYSTEM

In many ways, using a computer system or set of work processes is much like driving on a dirt road. Over time with repeated use, the road begins to develop ruts in the most commonly used parts of the road. Although these ruts show where to drive, they make change difficult. As people use a computer system or set of work processes, those system/work processes begin to become habits or norms; people learn them and become comfortable with them. These system or work processes then begin to limit people's activities and make it difficult for them to change because they begin to see their jobs in terms of these processes rather than in terms of the final business goal of serving customers.

One of the earliest models for managing organizational change was developed by Kurt Lewin.<sup>1</sup> Lewin argued that change is a three-step process: unfreeze, move, refreeze (Figure 13-1). First, the project team must *unfreeze* the existing habits and norms (the as-is system) so that change is possible. Most of the SDLC to this point has laid the groundwork for unfreezing. Users are aware of the new system being developed, some have participated in an analysis of the current system (and so are aware of its problems), and some have helped design the new system (and so have some sense of the potential benefits of the new system). These activities have helped to unfreeze the current habits and norms.

The second step of Lewin's three-step model is to *move*, or *transition*, from the old system to the new. The *migration plan* incorporates many issues that must be addressed to facilitate this transition. First, the *conversion strategy* needs to be selected, determining the style of the switch from the old to the new system, what parts of the organization will be converted when, and how much of the system is converted at a time. Plans to handle potential business disruption due to technical problems during conversion should be outlined in the *business contingency plan*. Arrangements for the hardware and software installation should be completed, and decisions about how the data will be converted into the new system will be made. The final major segment of the migration plan involves helping the people who are affected by the new

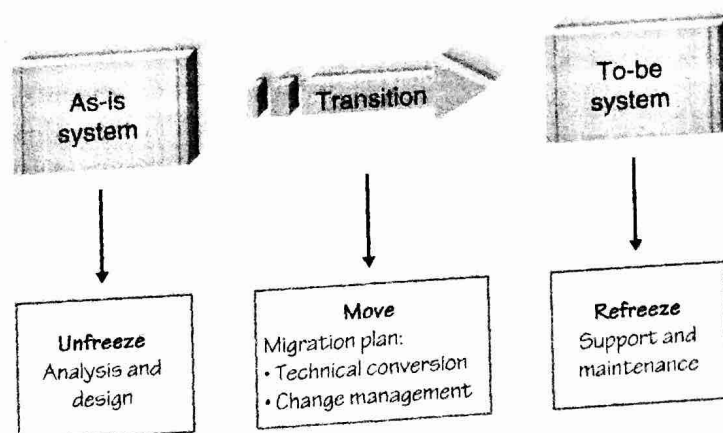


FIGURE 13-1  
Implementing Change

<sup>1</sup> Kurt Lewin, "Frontiers in Group Dynamics," *Human Relations*, 1947, 1:5-41; and Kurt Lewin, "Group Decision and Social Change" in E. E. Maccoby, T. M. Newcomb, and E. L. Hartley, eds., *Readings in Social Psychology*, New York: Holt, Rinehart & Winston, 1958, pp. 197-211.

system understand the change and motivating them to adopt the new system. The next section of this chapter discusses these aspects of the migration plan.

Lewin's third step is to *refreeze* the new system as the habitual way of performing the work processes—ensuring that the new system successfully becomes the standard way of performing the business functions it supports. This refreezing process is a key goal of the postimplementation activities discussed in the final section of this chapter. By providing ongoing support for the new system and immediately beginning to identify improvements for the next version of the system, the organization helps solidify the new system as the new habitual way of doing business. Postimplementation activities include *system support*, which means providing help desk and telephone support for users with problems; *system maintenance*, which means fixing bugs and improving the system after it has been installed; and *project assessment*, which is the process of evaluating the project to identify what went well and what could be improved for the next system development project.

## THE MIGRATION PLAN

The transition from the old business processes and computer programs to the new business processes and computer programs will be facilitated by ensuring that a number of business, technical, and people issues are addressed. The decisions, plans, and procedures that will guide the transition are outlined in the *migration plan* (Figure 13-2). The migration plan specifies what activities will be performed when and by whom as the transition is made from the old to the new system.

In order to ensure that business is ready to make the transition, the project team must determine the best conversion strategy to use as the new system is introduced to the organization. Also, plans should be made to ensure that the business can continue its operations even in the event of technical glitches in the new system. These plans are termed *business contingency plans*.

Technical readiness is achieved by arranging for and installing any needed hardware and software, and converting data as needed for the new system. These arrangements, while essential, are usually the least difficult of all the issues dealt with in the migration plan.

Ensuring that the people who will be affected by the new system are ready and able to use it is the most complex element of the migration plan. Managing the “people” side of change requires the team to understand the potential for resistance to the new system, develop organizational support and encouragement for the change, and prepare the users through appropriate training activities.

### Selecting the Conversion Strategy

The process by which the new system is introduced into the organization is called the conversion strategy. Those implementing this strategy must consider three different aspects of introducing the system: how abruptly the change is made (the *conversion style*), the organizational span of

| Migration Plan                                                           |                                                       |                                                                 |
|--------------------------------------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------------|
| Preparing the Business                                                   | Preparing the Technology                              | Preparing the People                                            |
| <input checked="" type="checkbox"/> Select a conversion strategy.        | <input checked="" type="checkbox"/> Install hardware. | <input checked="" type="checkbox"/> Revise management policies. |
| <input checked="" type="checkbox"/> Prepare a business contingency plan. | <input checked="" type="checkbox"/> Install software. | <input checked="" type="checkbox"/> Assess costs and benefits.  |
|                                                                          | <input checked="" type="checkbox"/> Convert data.     | <input checked="" type="checkbox"/> Motivate adoption.          |
|                                                                          |                                                       | <input checked="" type="checkbox"/> Conduct training.           |

**FIGURE 13-2**  
Elements of a  
Migration Plan

the introduction (conversion *locations*), and the extent of the system that is introduced (conversion *modules*). The choices made in these three dimensions will affect the cost, time, and risk associated with the transition, as explained in the sections that follow (Figure 13-3).

**Conversion Style** The switch from the old system to the new system can be made abruptly or gradually. An abrupt change is called *direct conversion*, and, as the name implies, involves the instant replacement of the old system with the new system. In essence, the old system is turned off and the new system is turned on, often coinciding with a fiscal-year change or other calendar event.

Direct conversion is simple and straightforward, but also risky. Any problems with the new system that have not been detected during testing may seriously disrupt the organization's ability to function.

A more gradual introduction is made with *parallel conversion*, in which both the old and the new systems are used simultaneously for a period of time. The two systems are operated side by side, and users must work with both the old and new systems. For example, if a new accounting system is introduced with a parallel conversion style, data must be entered into both systems. Output from both systems is carefully compared to ensure that the new system is performing correctly. After some period (often 1–2 months) of parallel operation and intense comparison between the two systems, use of the old system is discontinued.

Parallel conversion reduces risk by providing the organization with a fallback position if major problems are encountered with the new system. It adds expense, however, as users are required to do their job tasks twice: once with each system that performs the same function.

**Conversion Locations** The new system can be introduced to different parts of the organization at different times, or it can be introduced throughout the organization at the same time. A *pilot conversion* selects one or more locations (or units or work groups within a location) to be converted first as a part of a pilot test. If the conversion at the pilot location is successful, then the system is installed at the remaining locations.

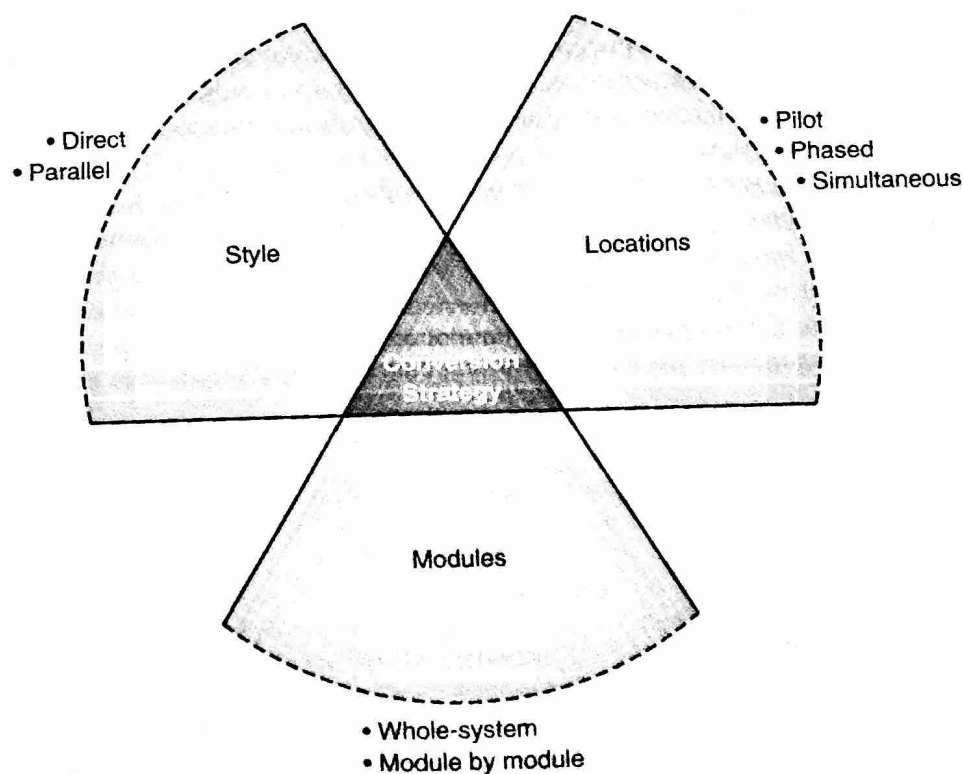


FIGURE 13-3  
Conversion Strategies



Pilot conversion has the advantage of limiting the effect of the new system to just the pilot location. In essence, an additional level of testing is provided before the new system is introduced organizationwide. This type of conversion can be done only in organizations that can tolerate different locations using different systems and business processes for a certain length of time. It also obviously requires a considerable time before the system is installed at all organizational locations.

In some situations, it is preferable to introduce the system to different locations, in phases. With *phased conversion*, a first set of locations is converted, then a second set, then a third set, and so on, until all locations are converted. Sometimes there is a deliberate delay between the phases, so that any problems with the system are detected before too much of the organization is affected. In other circumstances, the project team may begin a new phase immediately following the completion of the previous phase.

Phased conversion has the same advantages and disadvantages as pilot conversion. It also involves a smaller set of people to perform the actual conversion (and any associated training) than if all locations were converted at once.

It may be necessary to convert all locations at the same time, suggesting the need for *simultaneous conversion*. The new system is installed at all locations at once, thus eliminating the problem of having different organizational units using different systems and processes. The drawback of this option is that there must be sufficient staff to perform the conversion and train the users at all locations simultaneously.

**Conversion Modules** Although we typically expect that systems are installed in their entirety, this is not always the case. It may be desirable to decide how much of the new system will be introduced into the organization at a time. When the modules within the system are separate and distinct, organizations may convert to the new system one module at a time, using *modular conversion*. Modular conversion requires special care in developing the system (and usually adds extra cost), because each module must be written to work with both the old and the new systems. When modules are tightly integrated, this is very challenging and is therefore seldom done. When the software is written with loose association between modules, however, it becomes easier.

Modular conversion reduces the amount of training needed for people to begin using the new system, since users need to be trained only for the new module being implemented. Modular conversion does require significant time to introduce each module of the system in sequence.

*Whole-system conversion*, installing the entire system at one time, is most common. This approach is simple and straightforward and is required if the system consists of tightly integrated

## CONCEPTS

### IN ACTION

#### 13-A Converting to the Euro (Part 1)

When the European Union decided to introduce the euro, the European Central Bank had to develop a new computer system (called Target) to provide a currency settlement system for use by investment banks and brokerages. Prior to the introduction of the euro, settlement was performed between the central banks of the countries involved. After the introduction, the Target system, which consists of 15 national banking systems, would settle trades and perform currency conversions for cross-border payments for stocks and bonds.

Adapted from: "Debut of Euro Nearly Flawless," *Computerworld*, 33(2), p. 16, January 11, 1999, by Thomas Hoifman.

#### Question

Implementing Target was a major undertaking for a number of reasons. If you were an analyst on the project, what kinds of issues would you have to address to make sure the conversion happened successfully?

modules. If the system is large and/or extremely complex, however (e.g., an enterprise resource planning system such as SAP or Oracle), the whole system may prove too difficult for users to learn in one conversion step.

**Evaluating the Strategy Choices** Each of the segments in Figure 13-3 are independent, so a conversion strategy can be developed by combining any of the options just discussed.

For example, one commonly used approach is to begin with a pilot conversion of the whole system, using parallel conversion in a handful of test locations. Once the system has passed the pilot test at these locations, it is then installed in the remaining locations by phased conversion with direct cutover. There are three important factors to consider in selecting a conversion strategy: risk, cost, and the time required (Figure 13-4).

**Risk** The introduction of the new system exposes the organization to risk associated with problems and errors that may impede business operations. After the system has passed a rigorous battery of unit, integration, system, and acceptance testing, it should be bug free—maybe. Because humans make mistakes, undiscovered bugs may exist. Depending on the choices made, the conversion process provides one last step in which bugs can be detected and fixed before the system is in widespread use.

The parallel conversion strategy is less risky than direct conversion because of the security of continuing to operate the old system. If bugs are encountered, the new system can be shut down and fixed while the old system continues to function. Converting a pilot location is less risky than phased conversion or simultaneous conversion because the effects of bugs are limited to the pilot location. Those involved, knowing the installation is a pilot test, expect to encounter bugs. Finally, converting by modules is less risky than simultaneous conversion. The number of bugs encountered at any one time should be fewer when a few modules at a time are converted, making it easier to deal with problems as they occur. If numerous bugs are experienced together during simultaneous conversion, the total effect may be more disruptive than if the bugs were encountered gradually.

The significance of the risk factor in selecting a conversion strategy depends on the system being implemented. The team must weigh the probability of undetected bugs remaining in the system against the potential consequences of those undetected bugs. If the system has undergone extensive methodical testing, including alpha and beta testing, then the probability of undetected bugs is lower than if testing were less rigorous. There remains the chance, however, that mistakes were made in analysis and that the new system may not properly fulfill the business requirements.

Assessing the consequences (or cost) of a bug is challenging. Most analysts and senior managers are capable of making a reasonable guess at the relative significance of a bug, however. For example, it is obvious that the importance of negative consequences of a bug in an automated stock market trading system or a medical life-support system is much greater than in a computer game or word processing program (recall Figure 12-1). Therefore, risk is likely to be a very important factor in the selection of a conversion strategy if the system has had



| Characteristic | Conversion Style  |                     | Conversion Location |                   |                         | Conversion Modules      |                    |
|----------------|-------------------|---------------------|---------------------|-------------------|-------------------------|-------------------------|--------------------|
|                | Direct Conversion | Parallel Conversion | Pilot Conversion    | Phased Conversion | Simultaneous Conversion | Whole-System Conversion | Modular Conversion |
| Risk           | High              | Low                 | Low                 | Medium            | High                    | High                    | Medium             |
| Cost           | Low               | High                | Medium              | Medium            | High                    | Medium                  | High               |
| Time           | Short             | Long                | Medium              | Long              | Short                   | Short                   | Long               |

FIGURE 13-4 Characteristics of Conversion Strategies

limited testing and/or if the significance of bugs is high. If the system has been thoroughly tested and/or the cost of bugs is not too high, then risk becomes less important to the conversion strategy decision.

**Cost** The various conversion strategies have different costs. These costs can include salaries for people who work with the system (e.g., users, trainers, system administrators, external consultants), travel expenses, operation expenses, communication costs, and hardware leases. Parallel conversion is more expensive than direct cutover because it requires that two systems (the old and the new) be operated at the same time. Employees must now perform twice the usual work and also cross-check the results of the two systems.

Pilot conversion and phased conversion have somewhat similar costs. Simultaneous conversion has higher costs because more staff are required to support all the locations as they simultaneously switch from the old to the new system. Modular conversion is more expensive than whole system conversion because it requires more programming. The old system must be updated to work with selected modules in the new system, and modules in the new system must be programmed to work with selected modules in both the old and new systems.

**Time** The final factor is the amount of time required to convert between the old and the new system. Direct conversion is the fastest because it is immediate. Parallel conversion takes longer because the full advantages of the new system do not become available until the old system is turned off. Simultaneous conversion is fastest because all locations are converted at the same time. Phased conversion generally takes longer than pilot conversion because usually (but not always), once the pilot test is complete, all remaining locations are simultaneously converted. Phased conversion proceeds in waves, often requiring several months before all locations are converted. Likewise, modular conversion takes longer than whole-system conversion because the modules are introduced one after another.

### Preparing a Business Contingency Plan

It is tempting to believe that doing careful and thorough work in analysis and design and managing the IT project correctly will produce a successful system implementation. It is common for the team to view their prospects for success with optimism. With new systems, however, it may be more appropriate to always expect the worst. Keeping small technology glitches in the new system from turning into major business disasters is known as *business contingency planning*. Contingency plans help the business withstand relatively small problems with the new system so that major business disruptions are prevented.

Some might say that business disasters are prevented with good project management and migration planning; therefore, developing contingency plans to cope with disasters is unnecessary. Large projects spanning multiple business processes and involving huge amounts of code, however, provide numerous combinations of relatively small technical problems that together can have devastating consequences. Enterprise resource planning software projects

## YOUR TURN

### 13-1 Developing a Conversion Strategy

Suppose that you are leading the conversion from one word processor to another at your university. Develop a conversion strategy. You have also been asked to develop a conversion strategy for the university's new

Web-based course registration system. How would the second conversion strategy be similar to or different from the one you developed for the word processor?

## CONCEPTS

## IN ACTION

## 13-B US Army Installation Support

Throughout the 1960s, 1970s, and 1980s, the US Army automated its installations ("army bases," in civilian terms). Automation was usually a local effort at each of the more than 100 bases. Although some bases had developed software together (or borrowed software developed at other bases), each base often had software that performed different functions or performed the same function in different ways. In 1989, the army decided to standardize the software so that the same software would be used everywhere. This would greatly reduce software maintenance and also reduce training when soldiers were transferred between bases.

The software took four years to develop. The system was quite complex, and the project manager was concerned that there was a high risk that not all requirements of all installations had been properly captured. Cost and time were less important, since the project had already run four years and cost \$100 million.

Therefore, the project manager chose a modular pilot conversion using parallel conversion. The manager selected seven installations, each representing a different type of army installation (e.g., training base, arsenal, depot) and began the conversion. All went well, but several new features were identified that had been overlooked during the analysis, design, and construction. These were added and the pilot testing resumed. Finally, the system was installed in the rest of the army installations using a phased direct conversion of the whole system. *Alan Dennis*

## Questions

1. Do you think the conversion strategy was appropriate?
2. Regardless of whether you agree, what other conversion strategy could have been used?

are good examples. In 2004, Hewlett-Packard experienced an estimated \$160 million financial impact when a \$30-million SAP project in the Industry Standard Server division experienced relatively minor programming problems. In 2001, Nike experienced small IT problems in an SAP installation that cost the company \$100 million in lost revenue.<sup>2</sup> It may be less risky to plan for how to cope with system failure (contingency plan) than to try to prevent failure purely through project management techniques.

Choosing parallel conversion is one approach to contingency planning. Operating the old and new systems together for a time ensures that a fallback system is available if problems occur with the new system. Parallel conversion is not always feasible, however. Consequently, the worst-case outcome—*no system at all*—should be imagined and planned for, potentially going back to simple manual procedures.

One of the limitations of problem prevention through perfect project management techniques is the constant pressure of budget constraints and limited time that most projects face.

YOUR  
TURN

## 13-2 Comparing Conversion Strategies

- Develop the combination of conversion strategy dimensions that produces the least risk; the most risk.
- Develop the combination of conversion strategy dimensions that produces the least cost; the most cost.

- Develop the combination of conversion strategy dimensions that requires the least time; the most time.

Now, compare these strategies. Do you see any relationships? Based on your analysis, what advice might you give a team selecting a conversion strategy?

<sup>2</sup> Christopher Koch, "When Bad Things Happen to Good Projects," *CIO Magazine*, December 1, 2004.



With no budget or time pressure, it might be possible to prevent problems from occurring, but this is rarely the situation. Therefore, during the development of the migration plan, the project team should devote some attention to identifying the worst-case scenarios for the project, understanding the total business impact of those worst-case scenarios, and developing procedures and work-arounds that will enable the business to withstand those events. Since the contingency plan focuses on keeping the business up and running in the event of IT problems, it will be important to involve key business managers and users in the plan development.

### Preparing the Technology

There are three major steps involved in preparing the technical aspects of the new system for operations: install the hardware, install the software, and convert the data (Figure 13-2). Although it may be possible to do some of these steps in parallel, they usually must be performed sequentially at any one location.

The first step is to buy and install any needed hardware. In many cases, no new hardware is needed, but sometimes the project requires new servers, client computers, printers, and networking equipment. The new hardware requirements should have been defined in the hardware and software specifications during design (see Chapter 8) and used to acquire the needed resources. It is now critical to work closely with vendors who are supplying needed hardware and software to ensure that the deliveries are coordinated with the conversion schedule so that the equipment is available when it is needed. Nothing can stop a conversion plan in its tracks as easily as the failure of a vendor to deliver needed equipment.

Once the hardware is installed, tested, and certified as being operational, the second step is to install the software. This includes the to-be system under development, and sometimes, additional software that must be installed to make the system operational. For example, the Tune Source Digital Music Download system needs Web server software. At this point, the system is usually tested again to ensure that it operates as planned.

The third step is to convert the data from the as-is system to the to-be system. Data conversion is usually the most technically complicated step in the migration plan. Often, separate programs must be written to convert the data from the as-is system to the new formats required in the to-be system and store it in the to-be system files and databases. This process is often complicated by the fact that the files and databases in the to-be system do not exactly match the files and databases in the as-is system (e.g., the to-be system may use several tables in a database to store customer data that was contained in one file in the as-is system). Formal test plans are always required for data conversion efforts (see Chapter 12).

### Preparing People for the New System

In the context of a systems development project, people who will use the new system need help to adopt and adapt to the new system. The process of helping them adjust to the new system and its new work processes without undue stress is called *change management*.<sup>3</sup> There are three key roles in any major organizational change. The first is the *sponsor* of the change—the person who wants the change. This person is the business sponsor who first initiated the request for the new system (see Chapter 1). Usually the sponsor is a senior manager of the part of the organization that must adopt and use the new system. It is critical that the sponsor be active in the change management process, because a change that is clearly being driven by the sponsor, not by the project team or the IS organization, has greater legitimacy in the eyes of the users. The sponsor has direct management authority over those who will adopt the system.

<sup>3</sup> Many books have been written on change management. Some of our favorites are the following: Patrick Connor and Linda Lake, *Managing Organizational Change*, 2nd ed., Westport, CT: Praeger, 1994; Douglas Smith, *Taking Charge of Change*, Reading, MA: Addison-Wesley, 1996; and Daryl Conner, *Managing at the Speed of Change*, New York: Villard Books, 1992.



The second role is that of the *change agent*—the person(s) leading the change effort. The change agent, charged with actually planning and implementing the change, is usually someone outside of the business unit adopting the system and therefore has no direct management authority over the potential adopters. Because the change agent is an outsider from a different organizational culture, he or she has less credibility than do the sponsor and other members of the business unit. After all, once the system has been installed, the change agent usually leaves and thus has no ongoing impact.

The third role is that of *potential adopter*, or target of the change—the people who actually must change. These are the people for whom the new system is designed and who will ultimately choose to use or not use the system.

In the early days of computing, many project teams simply assumed that their job ended when the old system was converted to the new system at a technical level. The philosophy was “build it and they will come.” Unfortunately, that happens only in the movies. Resistance to change is common in most organizations. Therefore, the change management plan is an important part of the overall migration plan that glues together the key steps in the change management process. Successful change requires that people want to adopt the change and are able to adopt the change. The change management plan has four basic steps: revising management policies, assessing the cost and benefit models of potential adopters, motivating adoption, and enabling people to adopt through training (Figure 13-2). Before we can discuss the change management plan, however, we must first understand why people resist change.

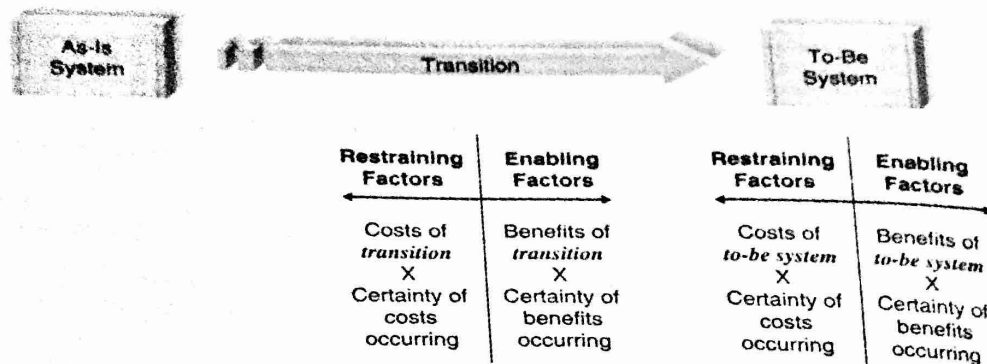
### Understanding Resistance to Change

People resist change—even change for the better—for very rational reasons.<sup>4</sup> What is good for the organization is not necessarily good for the people who work there. For example, consider an order-processing clerk who used to receive orders to be shipped on paper shipping documents, but now uses a computer to receive the same information. Rather than typing shipping labels with a typewriter, the clerk now clicks on the print button on the computer and the label is produced automatically. The clerk can now ship many more orders each day, which is a clear benefit to the organization. The clerk, however, probably doesn't really care how many packages are shipped. His or her pay doesn't change; it's just a question of whether the clerk prefers a computer or typewriter. Learning to use the new system and work processes—even if the change is minor—requires more effort than continuing to use the existing, well-understood system and work processes.

So why do people accept change? Simply put, every change has a set of costs and benefits associated with it. If the benefits of accepting the change outweigh the costs of the change, then people change. And sometimes the benefit of change is avoidance of the pain that you would experience if you did not adopt the change (e.g., if you don't change, you are fired, so one of the benefits of adopting the change is that you still have a job).

In general, when people are presented with an opportunity for change, they perform a cost-benefit analysis (sometimes consciously, sometimes subconsciously) and decide the extent to which they will embrace and adopt the change. They identify the costs of and benefits from the system and decide whether the change is worthwhile. However, it is not that simple, because most costs and benefits are not certain. There is some uncertainty as to whether a certain benefit or cost will actually occur; so both the costs of and benefits from the new system will need to be weighted by the degree of certainty associated with them (Figure 13-5). Unfortunately, most humans tend to overestimate the probability of costs and underestimate the probability of benefits.

<sup>4</sup> This section benefited from conversations with Dr. Robert Briggs, research scientist at the Center for the Management of Information at the University of Arizona.



**FIGURE 13-5**  
The Costs and Benefits of Change

There are also costs and benefits associated with the actual transition process itself. For example, suppose that you found a nicer house or apartment than your current one. Even if you liked it better, you might decide not to move, simply because the cost of moving outweighed the benefits of the new house or apartment itself. Likewise, adopting a new computer system might require you to learn new skills, which could be seen as a cost by some people, but as a benefit by others who perceive that those skills may somehow provide other benefits beyond the use of the system itself. Once again, any costs and benefits from the transition process must be weighted by the certainty with which they will occur (Figure 13-5).

Taken together, these two sets of costs and benefits (and their relative certainties) affect the acceptance of change or resistance to change that project teams encounter when installing new systems in organizations. The first step in change management is to understand the factors that inhibit change—the factors that affect the *perception* of costs and benefits and certainty that they will be generated by the new system. It is critical to understand that the “real” costs and benefits are far less important than the perceived costs and benefits. People act on what they *believe* to be true, not on what *is* true. Thus, any understanding of how to motivate change must be developed from the viewpoint of the people expected to change, not from the viewpoint of those leading the change.

### Revising Management Policies

The first major step in the change management plan is to change the management policies that were designed for the as-is system to new management policies designed to support the to-be system. *Management policies* provide goals, define how work processes should be performed, and determine how organizational members are rewarded. No computer system will be successfully adopted unless management policies support its adoption. Many new computer systems bring changes to business processes; they enable new ways of working. Unless the policies that provide the rules and rewards for those processes are revised to reflect the new opportunities that the system permits, potential adopters cannot easily use it.

Management has three basic tools for structuring work processes in organizations.<sup>5</sup> The first is the *standard operating procedures* (SOPs) that become the habitual routines for

<sup>5</sup> This section builds on the work of Anthony Giddons, *The Constitution of Society: Outline of the Theory of Structure*, Berkeley: University of California Press, 1984. A good summary of Giddons' theory that has been revised and adapted for use in understanding information systems is an article by Wanda Orlikowski and Dan Robey, “Information Technology and the Structuring of Organizations,” *Information Systems Research*, 1991, 2(2):143–169.

### 13-3 Standard Operating Procedures



Identify and explain three SOPs for the course in which you are using this book. Discuss whether they are formal or informal.

how work is performed. The SOPs are both formal and informal. Formal SOPs define proper behavior. Informal SOPs are the norms that have developed over time for how processes are actually performed. Management must ensure that the formal SOPs are revised to match the to-be system. The informal SOPs will then evolve to refine and fill in details absent in the formal SOPs.

The second aspect of management policy is defining how people assign meaning to events. What does it mean to "be successful" or "do good work"? Policies help people understand meaning by defining *measurements* and *rewards*. Measurements explicitly define meaning because they provide clear and concrete evidence about what is important to the organization. Rewards reinforce measurements because "what gets measured gets done" (an overused, but accurate, saying). Measurements must be carefully designed to motivate desired behavior. The IBM credit example ("Your Turn 3-3" in Chapter 3) illustrates the problem when flawed measurements drive improper behavior. (When the credit analysts became too busy to handle credit requests, they would "find" nonexistent errors so that they could return the requests unprocessed.)

A third aspect of management policy is *resource allocation*. Managers can have a clear and immediate impact on behavior by allocating resources. They can redirect funds and staff from one project to another, create an infrastructure that supports the new system, and invest in training programs. Each of these activities has both a direct and a symbolic effect. The direct effect comes from the actual reallocation of resources. The symbolic effect shows that management is serious about its intentions. There is less uncertainty about management's long-term commitment to a new system when potential adopters see resources being committed to support it.

### Assessing Costs and Benefits

The next step in developing a change management plan is to develop two clear and concise lists of costs and benefits provided by the new system (and the transition to it), compared with the as-is system. The first list is developed from the perspective of the organization, which should flow easily from the business case developed during the feasibility study and refined over the life of the project (see Chapter 1). This set of organizational costs and benefits should be distributed widely so that everyone expected to adopt the new system clearly understands why the new system is valuable to the organization.

The second list of costs and benefits is developed from the viewpoints of the different potential adopters expected to change, or stakeholders in the change. For example, one set of potential adopters may be the front-line employees, another may be the first-line supervisors, and yet another might be middle management. Each of these potential adopters or stakeholders may have a different set of costs and benefits associated with the change—costs and benefits that can differ widely from those of the organization. In some situations, unions may be key stakeholders that can make or break successful change.

Many systems analysts naturally assume that front-line employees are the ones whose set of costs and benefits are the most likely to diverge from those of the organization and thus are the ones who most resist change. However, these employees usually bear the brunt

of problems with the current system. When problems occur, they often experience them firsthand. Middle managers and first-line supervisors are the most likely to have a divergent set of costs and benefits; therefore, they resist change because new computer systems often change how much power those individuals have. For example, a new computer system may improve the organization's control over a work process (a benefit to the organization), but reduce the decision-making power of middle management (a clear cost to middle managers).

An analysis of the costs and benefits for each set of potential adopters or stakeholders will help pinpoint those who will likely support the change and those who may resist the change. The challenge at this point is to try to change the balance of the costs and benefits for those expected to resist the change so that they support it (or at least do not actively resist it).

This analysis may uncover some serious problems that have the potential to block the successful adoption of the system. It may be necessary to reexamine the management policies and make significant changes to ensure that the balance of costs and benefits is such that important potential adopters are motivated to adopt the system.

Figure 13-6 summarizes some of the factors that are important to successful change. The first and most important is a compelling personal reason to change. All change is made by individuals, not organizations. If there are compelling reasons for the key groups of individual stakeholders to want the change, then the change is more likely to be successful. Factors such as increased salary, reduced unpleasantness, and—depending on the individuals—opportunities for promotion and personal development can be important motivators. If the change makes current skills less valuable, however, individuals may resist the change because they have invested a lot of time and energy in acquiring those skills and anything that diminishes those skills may be perceived as diminishing the individual (because important skills bring respect and power).

There must also be a compelling reason for the organization to need the change; otherwise, individuals become skeptical in regard to whether the change is important and are less certain that it will in fact occur. Probably the hardest organization to change is an organization that has been successful, because individuals come to believe that what worked in the past will continue to work. By contrast, in an organization that is on the brink of bankruptcy, it is easier to convince individuals that change is needed. Commitment and support from credible business sponsors and top management are also important in increasing the certainty that the change will occur.

The likelihood of successful change is increased when the cost of the transition to individuals who must change is low. The need for significantly different new skills or disruptions in operations and work habits may create resistance. A clear migration plan developed by a credible change agent who has support from the business sponsor is an important factor in increasing the certainty about the costs of the transition process.

## Motivating Adoption

The single most important factor in motivating a change is providing clear and convincing evidence of the need for change. Simply put, everyone who is expected to adopt the change must be convinced that the benefits from the to-be system outweigh the costs of changing.

There are two basic strategies to motivating adoption: informational and political. Both strategies are often used simultaneously. With an *informational strategy*, the goal is to convince potential adopters that the change is for the better. This strategy works when the cost-benefit set of the target adopters has more benefits than costs. In other words, there really are clear reasons for the potential adopters to welcome the change.



|                          | Factor                                         | Examples                                                                                                       | Effects                                                                                                                                                                    | Actions to Take                                                                                                                                                             |
|--------------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Benefits of to-be system | Compelling personal reason(s) for change       | Increased pay, fewer unpleasant aspects, opportunity for promotion, most existing skills remain valuable       | If the new system provides clear personal benefits to those who must adopt it, they are more likely to embrace the change.                                                 | Perform a cost-benefit analysis from the viewpoint of the stakeholders, make changes where needed, and actively promote the benefits.                                       |
|                          | Compelling organizational reason(s) for change | Risk of bankruptcy, acquisition, government regulation                                                         | If adopters do not understand why the organization is implementing the change, they are less certain that the change will occur.                                           | Perform a cost-benefit analysis from the viewpoint of the organization and launch a vigorous information campaign to explain the results to everyone.                       |
| Certainty of benefits    | Demonstrated top management support            | Active involvement, frequent mentions in speeches                                                              | If top management is not seen to actively support the change, there is less certainty that the change will occur.                                                          | Encourage top management to participate in the information campaign.                                                                                                        |
|                          | Committed and involved business sponsor        | Active involvement, frequent visits to users and project team, championing                                     | If the business sponsor (the functional manager who initiated the project) is not seen to actively support the change, there is less certainty that the change will occur. | Encourage the business sponsor to participate in the information campaign and play an active role in the change management plan.                                            |
|                          | Credible top management and business sponsor   | Management and sponsor who do what they say instead of being members of the "management fad of the month" club | If the business sponsor and top management have credibility in the eyes of the adopters, the certainty of the claimed benefits is higher.                                  | Ensure that the business sponsor and/or top management has credibility so that such involvement will help; if there is no credibility, involvement will have little effect. |
| Costs of transition      | Low personal costs of change                   | Few new skills needed                                                                                          | The cost of the change is not borne equally by all stakeholders; the costs are likely to be higher for some.                                                               | Perform a cost-benefit analysis from the viewpoint of the stakeholders, make changes where needed, and actively promote the low costs.                                      |
| Certainty of costs       | Clear plan for change                          | Clear dates and instructions for change, clear expectations                                                    | If there is a clear migration plan, it will likely lower the perceived costs of transition.                                                                                | Publicize the migration plan.                                                                                                                                               |
|                          | Credible change agent                          | Previous experience with change, does what he or she promises to do                                            | If the change agent has credibility in the eyes of the adopters, the certainty of the claimed costs is higher.                                                             | If the change agent is not credible, then change will be difficult.                                                                                                         |
|                          | Clear mandate for change agent from sponsor    | Open support for change agent when disagreements occur                                                         | If the change agent has a clear mandate from the business sponsor, the certainty of the claimed costs is higher.                                                           | The business sponsor must actively demonstrate support for the change agent.                                                                                                |

FIGURE 13-6 Major Factors in Successful Change



## CONCEPTS

### IN ACTION

#### 13-C Managing Global Projects

Shamrock Foods is a major food distributor centered in Tralee, Ireland. Originally a dairy cooperative, Shamrock branched into various food components (dried milk, cheese solids, flavorings (or flavourings, as the Irish would spell it)) and has had substantial growth in the past 10 years, most of which came by way of acquisition of existing companies or facilities. For example, Iowa Soybean in the United States is now a subsidiary of Shamrock Foods, as is a large dairy cooperative in Wisconsin.

Shamrock has processing facilities in over 12 countries and distribution and sales in over 30 countries. With the rapid growth by acquisition, the company has generally adopted a "hands-off" policy keeping the systems separated and not integrated into a unified ERP system. Thus, each acquired company is still largely autonomous, although it reports to Shamrock Foods and is managed by Shamrock Foods.

This separation concept has been a problem for Conor Lynch, CFO of Shamrock Foods. The board of directors would like some aggregated data for direction

and analysis of acquired businesses. Conor has the reports from the various subsidiaries, but has to have his staff convert the figures reported in them to a consistent basis (generally, either Euros or American dollars).

#### Questions

1. When should a multinational/multisite business consolidate data systems?
2. There are costs associated with consolidating data systems that have a variety of hardware and software systems. For example, the various acquired companies already had their own functioning accounting systems. What justification should Conor use to push for a consolidated, unified ERP system?
3. At times, Conor has to deal with incomplete and incompatible data. For instance, inventory systems might be FIFO for some of the subsidiaries and LIFO for other subsidiaries. How might a CFO with multinational interests deal with incomplete and incompatible data?

Using this approach, the project team provides clear and convincing evidence of the costs and benefits of moving to the to-be system. The project team writes memos and develops presentations that outline the costs and benefits of adopting the system from the perspective of the organization and from the perspective of the target group of potential adopters. This information is disseminated widely throughout the target group, much like an advertising or public relations campaign. It must emphasize the benefits as well as increase the certainty in the minds of potential adopters that these benefits will actually be achieved. In our experience, it is always easier to sell painkillers than vitamins; that is, it is easier to convince potential adopters that a new system will remove a major problem (or other source of pain) than that it will provide new benefits (e.g., increase sales). Therefore, informational campaigns are more likely to be successful if they stress the reduction or elimination of problems, rather than focus on the provision of new opportunities.

The other strategy to motivate change is a *political strategy*. With a political strategy, organizational power, not information, is used to motivate change. This approach is often used when the cost-benefit set of the target adopters has more costs than benefits. In other words, although the change may benefit the organization, there are no reasons for the potential adopters to welcome the change.

## YOUR TURN

#### 13-4 Overcoming Resistance to a New Executive Information System

How would you motivate adoption if you were the developer of a new executive information system designed to

provide your organization's top executives with key performance measures and economic trend information?

The political strategy is usually beyond the control of the project team. It requires someone in the organization who holds legitimate power over the target group to influence the group to adopt the change. This may be done in a coercive manner (e.g., "adopt the system or you're fired") or in a negotiated manner, in which the target group gains benefits in other ways that are linked to the adoption of the system (e.g., linking system adoption to increased training opportunities). Management policies can play a key role in a political strategy by linking salary to certain behaviors desired with the new system.

In general, for any change that has true organizational benefits, about 20% to 30% of potential adopters will be *ready adopters*. They recognize the benefits, quickly adopt the system, and become proponents of the system. Another 20% to 30% are *resistant adopters*. They simply refuse to accept the change, and they fight against it, either because the new system has more costs than benefits for them personally or because they place such a high cost on the transition process itself that no amount of benefits from the new system can outweigh the change costs. The remaining 40–60% are *reluctant adopters*. They tend to be apathetic and will go with the flow to either support or resist the system, depending on how the project evolves and how their coworkers react to the system. Figure 13-7 illustrates the actors who are involved in the change management process.

The goal of change management is to actively support and encourage the ready adopters and help them win over the reluctant adopters. There is usually little that can be done about the resistant adopters because their set of costs and benefits may be divergent from those of the organization. Unless there are simple steps that can be taken to rebalance their costs and benefits or the organization chooses to adopt a strong political strategy, it is often best to ignore this small minority of resistant adopters and focus on the larger majority of ready and reluctant adopters.

### Enabling Adoption: Training

Potential adopters may want to adopt the change, but unless they are capable of adopting it, they won't. Adoption is enabled by providing employees the skills needed to adopt the change through careful *training*. Training is probably the most self-evident part of any change management initiative. How can an organization expect its staff members to adopt a new system if they are not trained? We have found that training is one of the most commonly overlooked parts of the process, however. Many organizations and project managers simply expect potential adopters to find the system easy to learn. Since the system is presumed to be so simple, it is taken for granted that potential adopters should be able to learn with little effort. Unfortunately, this is usually an overly optimistic assumption.

Every new system requires new skills, either because the basic work processes have changed (sometimes radically in the case of business process reengineering (BPR); see Chapter 3) or because the computer system used to support the processes is different. The more radical the changes to the business processes, the more important it is to ensure that the organization has the new skills required to operate the new business processes and supporting information system. In general, there are three ways to get these new skills. One is to hire new

| Sponsor                                | Change Agent                              | Potential Adopters                                                                                                                                   |
|----------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| The sponsor wants the change to occur. | The change agent leads the change effort. | Potential adopters are the people who must change.<br>20–30% are ready adopters.<br>20–30% are resistant adopters.<br>40–60% are reluctant adopters. |

**YOUR  
TURN****13-5 Developing a Training Plan**

Suppose that you are leading the conversion from one word processor to another in your organization. Develop an outline of topics that would be included in the training. Develop a plan for training delivery.

employees who have the needed skills that the existing staff does not. Another is to outsource the processes to an organization that has the skills that the existing staff does not. Both of these approaches are controversial and are usually considered only in the case of BPR when the new skills needed are likely to be the most different from the set of skills of the current staff. In most cases, organizations choose the third alternative: training existing staff in the new business processes and the to-be system. Every training plan must consider what to train and how to deliver the training.

**What to Train** What training should you provide to the system users? It is obvious: how to use the system. The training should cover all the capabilities of the new system, so that users understand what each module does, right?

Wrong. Training for business systems should focus on helping the users to accomplish their jobs, not on how to use the system. The system is simply a means to an end, not the end in itself. This focus on performing the job (i.e., the business processes), not using the system, has two important implications. First, the training must focus on those activities around the system, as well as on the system itself. The training must help the users understand how the computer fits into the bigger picture of their jobs. The use of the system must be put in the context of the manual and computerized business processes, and it must also cover the new management policies that were implemented along with the new computer system.

Second, the training should focus on what the user needs to do, not on what the system can do. This is a subtle—but very important—distinction. Most systems will provide far more capabilities than the users will need to use (e.g., when was the last time you wrote a macro in Microsoft Word?). Rather than attempting to teach the users all the features of the system, training should instead focus on the much smaller set of activities that users perform on a regular basis and ensure that users are truly expert in those. When the focus is on the 20% of functions that the users will use 80% of the time (instead of attempting to cover all functions), users become confident about their ability to use the system. Training should mention the other little-used functions, but only so that users are aware of their existence and know how to learn about them when their use becomes necessary.

One source of guidance for designing training materials is the use cases and use scenarios. The use cases and use scenarios outline the common activities that users perform and thus can be helpful in understanding the business processes and system functions that are likely to be most important to the users.

**How to Train** There are many ways to deliver training. The most commonly used approach is *classroom training*. This has the advantage of training many users at one time with only one instructor and creates a shared experience among the users.

It is also possible to provide *one-on-one training* in which one trainer works closely with one user at a time. This is obviously more expensive, but the trainer can design the training program to meet the needs of individual users and can better ensure that the

**CONCEPTS****IN ACTION****13-D Finishing the Process**

As a great analyst, you have planned, analyzed, and designed a good solution. Now you need to implement it. As a part of implementation, do you think that training is just a wasted expense?

Stress is common in a help-desk call center. Users of computing services call to get access to locked accounts, get help when technology is not working as planned, and frequently can become very upset. Employees of the help-desk call center can get stressed out, and this can result in a greater number of sick days, less productivity, and higher turnover. Max Productivity Incorporated (MPI) is a training company that works with people in high-stress jobs. MPI's training program helps employees learn how to relax, how to "shake off" tough

users, and how to create "win-win" scenarios. MPI claims to be able to reduce employee turnover by 50%, increase productivity by 20%, and reduce stress, anger, and depression by 75%.

**Questions**

1. How would you challenge MPI to verify its claims regarding reducing turnover, increasing productivity, and decreasing stress and anger?
2. How would you conduct a "cost-benefit" analysis aimed at deciding whether to hire MPI to do ongoing training for your help-desk call center employees?

users really do understand the material. This approach is typically used only when the users are very important or when there are very few users.

Another approach that is becoming more common is to use some form of *computer-based training* (CBT), in which the training program is delivered via computer, either on DVD or over the Web. CBT programs can include text slides, audio, and even video and animation. CBT is typically more costly to develop, but is cheaper to deliver because no instructor is needed to actually provide the training.

Figure 13-8 summarizes four important factors to consider in selecting a training method. CBT is typically more expensive to develop than one-on-one or classroom training, but it is less expensive to deliver. One-on-one training has the most impact on the user because it can be customized to the user's precise needs, knowledge, and abilities, whereas CBT has the least impact. However, CBT has the greatest reach—the ability to train the most users over the widest distance in the shortest time—because it is so much simpler to distribute, compared with classroom and one-on-one training, since no instructors are needed.

Figure 13-8 suggests a clear pattern for most organizations. If there are only a few users to train, one-on-one training is the most effective. If there are many users to train, many organizations turn to CBT. We believe that the use of CBT will increase in the future. Quite often, large organizations use a combination of all three methods. Regardless of which approach is used, it is important to leave the users with a set of easily accessible materials that can be referred to long after the training has ended (usually a quick reference guide and a set of manuals, whether on paper or in electronic form).



|                 | <b>One-On-One<br/>Training</b> | <b>Classroom<br/>Training</b> | <b>Computer-Based<br/>Training</b> |
|-----------------|--------------------------------|-------------------------------|------------------------------------|
| Cost to develop | Low-medium                     | Medium                        | High                               |
| Cost to deliver | High                           | Medium                        | Low                                |
| Impact          | High                           | Medium-high                   | Low-medium                         |
| Reach           | Low                            | Medium                        | High                               |

**FIGURE 13-8**  
Selecting a Training  
Method

## POSTIMPLEMENTATION ACTIVITIES

The goal of postimplementation activities is to *institutionalize* the use of the new system—that is, to make it the normal, accepted, routine way of performing the business processes. The postimplementation activities attempt to refreeze the organization after the successful transition to the new system. Although the work of the project team naturally winds down after implementation, the business sponsor and, sometimes, the project manager are actively involved in refreezing. These two—and, ideally, many other stakeholders—actively promote the new system and monitor its adoption and usage. They usually provide a steady flow of information about the system and encourage users to contact them to discuss issues.

In this section, we examine three key postimplementation activities; support (providing assistance in the use of the system), maintenance (continuing to refine and improve the system), and project assessment (analyzing the project to understand what activities were done well—and should be repeated—and what activities need improvement in future projects).

### System Support

Once the project team has installed the system and performed the change management activities, the system is officially turned over to the *operations group*. This group is responsible for the operation of the system, whereas the project team is responsible for the development of the system. Members of the operations group usually are closely involved in the installation activities because they are the ones who must ensure that the system actually works. After the system is installed, the project team leaves but the operations group remains.

Providing system support means helping the users to use the system. Usually, this means providing answers to questions and helping users understand how to perform a certain function; this type of support can be thought of as *on-demand training*.

*Online support* is the most common form of on-demand training. This includes the documentation and help screens built into the system, as well as separate web sites that provide answers to *frequently asked questions* (FAQs) that enable users to find answers without contacting a person. Obviously, the goal of most systems is to provide sufficiently good online support so that the user does not need to contact a person, because providing online support is much less expensive than is providing a person to answer questions.

Most organizations provide a *help desk* that provides a place for a user to talk with a person who can answer questions (usually over the phone, but sometimes in person). The help desk supports all systems, not just one specific system, so it receives calls about a wide variety of software and hardware. The help desk is operated by *level 1 support* staff who have very broad computer skills and are able to respond to a wide range of requests, from network problems and hardware problems to problems with commercial software and with the business application software developed in house.

The goal of most help desks is to have the level 1 support staff resolve 80% of the help requests they receive on the first call. If the issue cannot be resolved by level 1 support staff, a *problem report* (Figure 13-9) is completed (often using a special computer system designed to track problem reports) and passed to a *level 2 support* staff member.

- Time and date of the report
- Name, e-mail address, and telephone number of the support person taking the report
- Name, e-mail address, and telephone number of the person who reported the problem
- Software and/or hardware causing the problem
- Location of the problem
- Description of the problem
- Action taken
- Disposition (problem fixed or forwarded to system maintenance)

**FIGURE 13-9**  
Elements of a  
Problem Report



The level 2 support staff members are people who know the application system well and can provide expert advice. For a new system, they are usually selected during the implementation phase and become familiar with the system as it is being tested. Sometimes, the level 2 support staff members participate in training during the change management process to become more knowledgeable with the system, the new business processes, and the users themselves.

The level 2 support staff works with users to resolve problems. Most problems are successfully resolved by the level 2 staff. In the first few months after the system is installed, however, the problem may turn out to be a bug in the software that must be fixed. In this case, the problem report becomes a *change request* that is passed to the system maintenance group.

### System Maintenance

System maintenance is the process of refining the system to make sure it continues to meet business needs. Over a system's lifetime, more money and effort are devoted to system maintenance than to the initial development of the system, simply because a system continues to change and evolve as it is used. Most beginning systems analysts and programmers work first on maintenance projects; usually only after they have gained some experience are they assigned to new development projects.

Every system is "owned" by a project manager in the IS group (Figure 13-10). This individual is responsible for coordinating the systems maintenance effort for that system. Whenever a potential change to the system is identified, a change request is prepared and forwarded to the project manager. The change request is a "smaller" version of the *system request* discussed in Chapters 1 and 2. It describes the change requested and explains why the change is important.

Changes can be small or large. Change requests that are likely to require a significant effort are typically handled in the same manner as system requests: They follow the same process as the project described in this book, starting with project initiation in Chapter 1 and following through installation in this chapter. Minor changes typically follow a "smaller" version of this same process. There is an initial assessment of feasibility and of costs and benefits, and the change request is prioritized. Then a systems analyst (or a programmer/analyst) performs the analysis, which may include interviewing users, and

#### CONCEPTS

#### IN ACTION

### 13-E Converting to the Euro (Part 2)

When the European Union decided to introduce the euro, the European Central Bank had to develop a new computer system (called Target) to provide a currency settlement system for use by investment banks and brokerages. The euro opened at an exchange rate of US \$1.167. However, a rumor that the Target system malfunctioned sent the value of the euro plunging two days later.

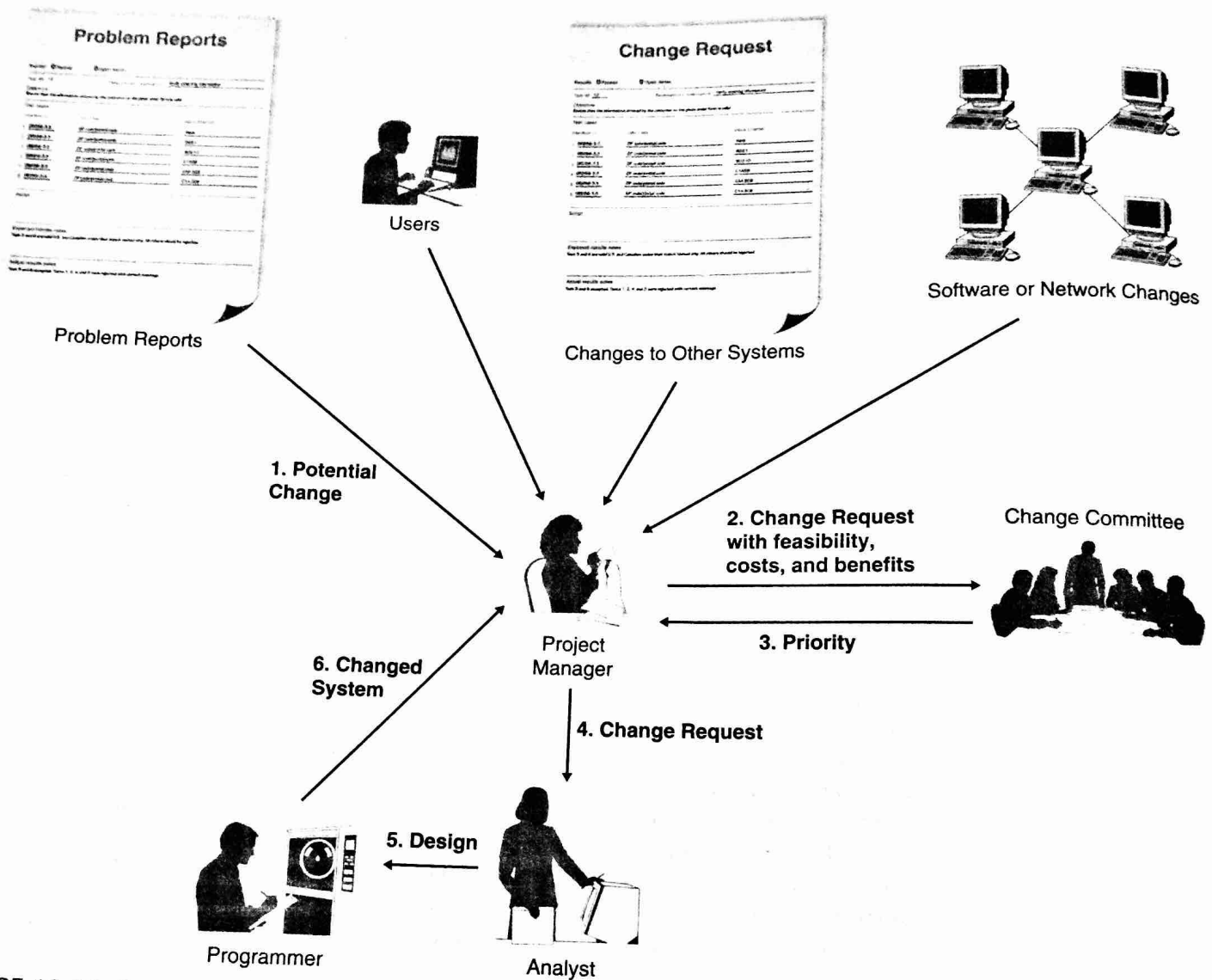
That evening, it was determined that the malfunction was not due to system problems. Instead, operators at some German banks had misunderstood how to use the system and had entered incorrect data. Once the

problems were identified and the operators quickly retrained, the Target system continued to operate and the euro quickly regained its lost value.

Adapted from: "Debut of Euro Nearly Flawless," *Computerworld*, January 11, 1999, 33(2), p. 16, by Thomas Hoffman.

#### Question

Target could be considered a high-risk system because of its effects on the European economy. What kinds of system support activities could be put in place to mitigate problems with Target?



**FIGURE 13-10** Processing a Change Request

prepares an initial design before programming begins. The new (or revised) program is then extensively tested before it is put into production.

Change requests typically come from five sources. The most common source is problem reports from the operations group that identify bugs in the system that must be fixed. These are usually given immediate priority because a bug can cause significant problems. Even a minor bug can cause major problems by upsetting users and reducing their acceptance of and confidence in the system.

The second most common source of change requests is enhancements to the system from users. As users work with the system, they often identify minor changes in the design that can make the system easier to use or identify additional functions that are needed. These enhancements are important in satisfying the users and are often key in ensuring that the system changes as the business requirements change. Enhancements are often given second priority after bug fixes.

A third source of change requests is other system development projects. For example, as part of Tune Source's Digital Music Download project, Tune Source likely had to make some minor changes to its existing Web-based CD sales system to ensure that the two systems would

**CONCEPTS****IN ACTION****13-F Software Bugs**

The awful truth is that every operating system and application system is defective. System complexity, the competitive pressure to hurry applications to market, and simple incompetence contribute to the problem.

Will software ever be bug free? Not likely. Microsoft Windows Group General Manager Chris Jones believes that bigger programs breed more bugs. Each revision is usually bigger and more complex than its predecessor, which means that there will always be new places for bugs to hide. Former Microsoft product manager Richard Freedman agrees that the potential for defects increases as software becomes more complex, but he believes that users ultimately win more than they lose. "I'd say the features have gotten exponentially better, and the product quality has degraded a fractional amount."

Still, the majority of users who responded to our survey said they'd buy a software program with fewer features if it were bug free. This sentiment runs counter to what most software developers believe. "People buy features, plain and simple," explains Freedman. "There have been attempts to release stripped-down word processors and spreadsheets, and they don't sell." Freedman says a trend toward smaller, less-bug-prone software with fewer features will "never happen."

Eventually, the software ships and the bug reports start rolling in. What happens next is what separates the companies you want to patronize from the slackers. While almost every vendor provides bug fixes eventually, some companies do a better job of it than others. Some observers view Microsoft's market dominance as a roadblock to bug-free software. Todd Paglia, an attorney with the Washington, D.C.-based Consumer Project on Technology, says, "If actual competition for operating systems existed and we had greater competition for some of the software that runs on the Microsoft operating system, we would have higher quality than we have now."

Adapted from: "Software Bugs Run Rampant," *PC World*, January 1999 17(1), p. 46, by Scott Spanbauer.

**Question**

If commercial systems contain the amount of bugs that this article suggests, what are the implications for systems developed in-house? Would in-house systems be more likely to have a lower or higher quality than commercial systems? Explain.

work together. These changes, required by the need to integrate two systems, are generally rare, but are becoming more common as system integration efforts become more common.

A fourth source of change requests is those that occur when underlying software or networks change. For example, a new version of Windows often will require an application to change the way it interacts with Windows or enable application systems to take advantage of new features that improve efficiency. While users may never see these changes (because most changes are inside the system and do not affect its user interface or functionality), these changes can be among the most challenging to implement because analysts and programmers must learn about the new system characteristics, understand how application systems use (or can use) those characteristics, and then make the needed programming changes.

The fifth source of change requests is senior management. These change requests are often driven by major changes in the organization's strategy (e.g., the Tune Source Digital Music Download project) or operations. These significant change requests are typically treated as separate projects, but the project manager responsible for the initial system is often placed in charge of the new project.

**Project Assessment**

The goal of *project assessment* is to understand what was successful about the system and the project activities (and therefore should be continued in the next system or project) and what needs to be improved. Project assessment is not routine in most organizations, except for military organizations, which are accustomed to preparing after-action reports. Nonetheless, assessment can be an important component in organizational learning because it helps

organizations and people understand how to improve their work. It is particularly important for junior staff members because it helps promote faster learning. There are two primary parts to project assessment—project team review and system review.

**Project Team Review** *Project team review* focuses on the way the project team carried out its activities. Each project member prepares a short two- to three-page document that reports on and analyzes his or her performance. The focus is on performance improvement, not penalties for mistakes made. By explicitly identifying mistakes and understanding their causes, project team members will, it is hoped, be better prepared for the next time they encounter a similar situation—and less likely to repeat the same mistakes. Likewise, by identifying excellent performance, team members will be able to understand why their actions worked well and how to repeat them in future projects.

The documents prepared by each team member are assessed by the project manager, who meets with the team members to help them understand how to improve their performance. The project manager then prepares a summary document that outlines the key learnings from the project. This summary identifies what actions should be taken in future projects to improve performance, but is not intended to identify team members who made mistakes. The summary is widely circulated among all project managers to help them understand how to manage their projects better. Often, it is also circulated among regular staff members who did not work on the project so that they, too, can learn from projects outside their scope.

**System Review** The focus of the *system review* is understanding the extent to which the proposed costs and benefits from the new system that were identified during project initiation were actually recognized from the implemented system. Project team review is usually conducted immediately after the system is installed, while key events are still fresh in team members' minds, but system review is often undertaken several months after the system is installed, because it often takes a while before the system can be properly assessed.

System review starts with the system request and feasibility analysis prepared at the start of the project. The detailed analyses prepared for the expected business value (both tangible

## PRACTICAL

### 13-1 Beating Buggy Software



**TIP**

**H**ow do you avoid bugs in the commercial software you buy? Here are six tips:

1. **Know your software:** Find out if the few programs you use day in and day out have known bugs and patches, and track the web sites that offer the latest information on them.
  2. **Back up your data:** This dictum should be tattooed on every monitor. Stop reading right now and copy the data you can't afford to lose onto a CD, second hard disk, or Web server. We will wait.
  3. **Don't upgrade—yet:** It is tempting to upgrade to the latest and greatest version of your favorite software, but why chance it? Wait a few months, check out other users' experiences with the upgrade on Usenet newsgroups or the vendor's own discussion forum, and then go for it. But only if you must.
  4. **Test thoroughly:** If you have a separate system before you install it on all the computers in your home or office.
  5. **Forget the betas:** Installing beta software on your primary computer is a game of Russian roulette. If you really have to play with beta software, get a second computer.
  6. **Complain:** The more you complain about bugs and demand remedies, the more costly it is for vendors to ship buggy products. It's like voting—the more people participate, the better are the results.
- Adapted from: "Software Bugs Run Rampant," *PC World*, January, 1999, 17(1), p. 46, Scott Spanbauer.

Adapted from: "Software Bugs Run Rampant," *PC World*, January, 1999, 17(1), p. 46, Scott Spanbauer.

and intangible), as well as the economic feasibility analysis, are reexamined, and a new analysis is prepared after the system has been installed. The objective is to compare the anticipated business value against the actual realized business value from the system. This helps the organization assess whether the system actually provided the value it was planned to provide. Whether or not the system provides the expected value, future projects can benefit from an improved understanding of the true costs and benefits.

A formal system review also has important behavioral implications for project initiation. Since everyone involved with the project knows that all statements about business value and the financial estimates prepared during project initiation will be evaluated at the end of the project, they have an incentive to be conservative in their assessments. No one wants to be the project sponsor or project manager for a project that goes radically over budget or fails to deliver promised benefits.

## APPLYING THE CONCEPTS AT TUNE SOURCE

Installation of the Digital Music Download system at Tune Source was somewhat simpler than the installation of most systems because the system was new; there was no as-is system. Most changes would be felt by the staff in the stores who might need to assist customers using the in-store kiosks to buy music downloads. Jason did not expect any major problems in this area.

### Implementation Process

The team was faced with the problem of installing the in-store kiosks and training the staff in all 50 of Tune Source's stores. Because the new in-store kiosks were new and not changing any old functions, it was less important to make sure that the conversion was completely synchronized across all stores. Because the risk associated with the system was low, and because both cost and time were somewhat important, Jason chose to use direct conversion with a whole-system conversion, but with a pilot phase first. You will recall from the last chapter that the system was beta tested with employees. Jason chose to use the beta test as the pilot conversion.

The in-store kiosks were installed in the 18 stores in a greater Los Angeles area for the beta test. The manager and the employees in each of these 18 stores were trained in the use of the new in-store kiosks.

Conversion during the pilot phase went smoothly. First, the new hardware was purchased and installed. Then the software was installed on the Web server of the company intranet. Data were loaded into the MySQL database by several temporary employees contracted for the job. Since this system contains new data, there was no data conversion to manage.

At the end of the beta test and pilot conversion phase, the kiosks were then installed in the remaining 32 stores in the Tune Source chain and their staffs were trained. Once the in-store kiosks were operating smoothly, the system was ready for general use by the public on the Internet.

### Preparing the People

There were few change management issues because there were few existing staff members who had to change. New staff were hired, most by internal transfer from other groups within Tune Source. The most likely stakeholders to be concerned by the change would be managers and employees in the traditional retail stores who might see the Digital Music Download system as a threat to their stores. Jason therefore developed an information campaign (distributed through the employee newsletter and internal Web site) that discussed the reasons for the change and explained that the Digital Music Download system should be seen as a complement to the existing stores, not as a competitor. The system was instead targeted at Web-based competitors such as Amazon.com.



The new management policies were developed, along with a training plan that encompassed both the manual work procedures and computerized procedures. Jason decided to use classroom training for the marketing personnel who would be using the promotion system component, because there was a small number of them and it was simpler and more cost effective to train them all together in one classroom session.

### Postimplementation Activities

Support of the system was turned over to the Tune Source operations group, who had hired four additional support staff members with expertise in networking and Web-based systems. System maintenance began almost immediately, with Jason designated as the project manager responsible for maintenance of this version of the system plus the development of the next version. Jason began planning to develop the next version of the system.

Project team review uncovered several key learnings, mostly involving Web-based programming and the difficulties in optimizing searches and providing fast, flawless file transfer protocols. The project was delivered on budget, with the exception that more was spent on programming than was anticipated.

A preliminary system review was conducted after two months of operations. Thanks to an advertising campaign, sales via the Internet system were \$120,000 for the first month and \$162,000 for the second, showing a gradual increase. (Remember that the goal for the first year of operations was \$2.6 million.) Operating expenses averaged \$47,000 per month, a bit higher than the projected average, owing to startup costs. Nonetheless, Carly Edwards, vice president of marketing and the project sponsor, was quite pleased. She approved the feasibility study for the follow-on project to develop the second version of the Internet system, and Jason began the SDLC all over again.

## CHAPTER REVIEW

After reading and studying this chapter, you should be able to:

- ☐ Identify and describe Lewin's three-step model of organizational change.
- ☐ Explain how the activities of systems analysis and design help with the unfreezing process.
- ☐ What is the purpose of the migration plan?
- ☐ Identify and describe the three dimensions of the conversion strategy and how choices made will impact the cost, time, and risk of conversion.
- ☐ Discuss the role and importance of business contingency planning during system migration.
- ☐ Identify and discuss the main reasons underlying people's resistance to change.
- ☐ Identify and describe ways to overcome resistance to change.
- ☐ Discuss the role and purpose of postimplementation support.
- ☐ Describe the sources of change requests.
- ☐ Discuss the best way to provide ongoing system maintenance.
- ☐ Discuss what is gained from conducting a post-project review?

## KEY TERMS

Business contingency plan  
Change agent  
Change management  
Change request

Classroom training  
Computer-based  
training (CBT)  
Conversion

Conversion location  
Conversion modules  
Conversion strategy  
Conversion style

Direct conversion  
Frequently asked questions  
(FAQs)  
Help desk