Cutover Strategies for System Migration

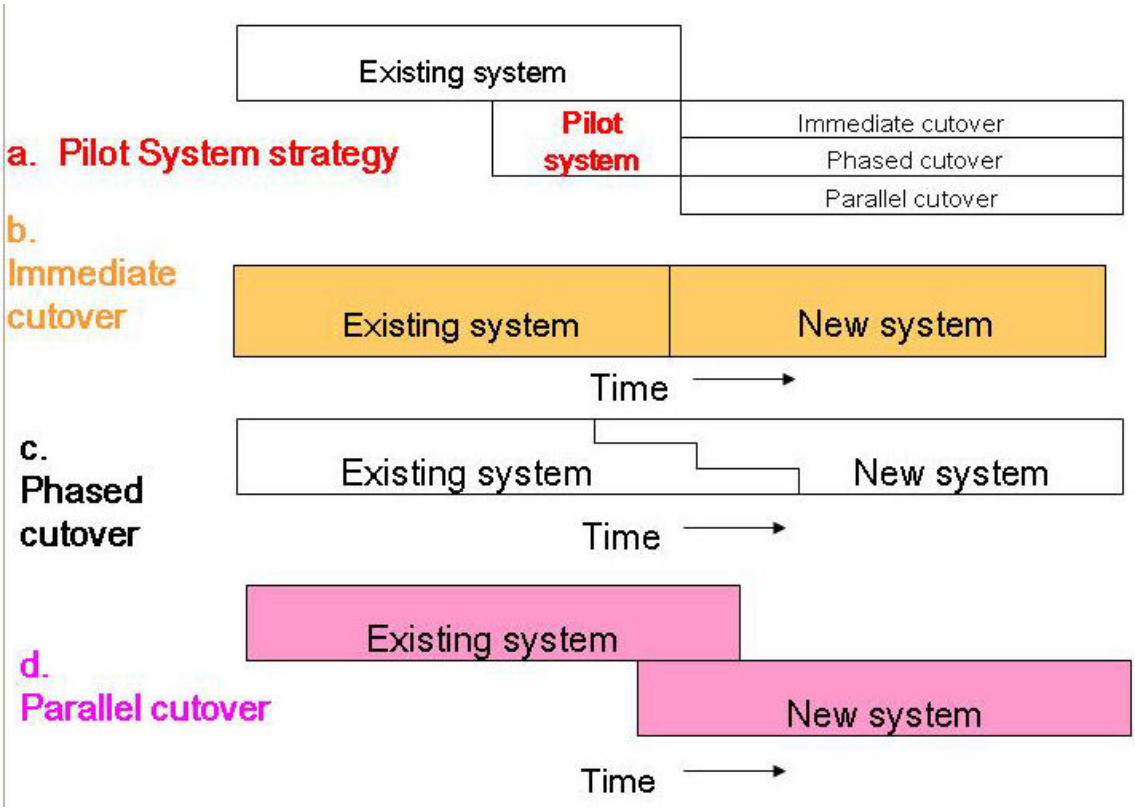
The final stage of the system development life cycle is devoted to migrating the new system from a development-testing-staging environment to the production environment and turning over control to the user. The main goals of migration are to meet the system requirements and to complete the project within constraints of time and cost.

The two basic migration steps are to (1) design and perform final system test and user acceptance tests, and (2) transfer system control to the users.

When the user accepts the new system as it is constructed, the developers turn the system over to the user and the system goes into production. This process may be called cutover, publishing, or deploying, depending on the context.

Cutover to the New System

There are four basic cutover strategies: (a) pilot system, (b) immediate cutover, (c) phased cutover, and (d) phased cutover.



Each of the three main strategies incurs different migration costs and a different degree of risk. Immediate cutover offers low cost yet high risk. Parallel cutover is a high-cost strategy but features low risk. Phased cutover offers a compromise in terms of medium cost and risk. Each can be an effective strategy, depending on the nature of the system being installed.

Pilot System

A **pilot system** is one that is installed in only one part of the firm's operations as a way to measure its impact. Once the pilot performs satisfactorily, the system is installed in the remainder of the firm in an immediate, phased, or parallel manner. A pilot system is like a test marketing activity that a firm can follow to evaluate the reception that consumers will give a new product. As an example, a soft drink firm can use such a city as Phoenix to test the appeal of a new drink.

Immediate Cutover

When **immediate cutover** is followed, the old system is dismantled and the new system is put into operation simultaneously. In other words, the old system is shut down and immediately replaced by the new system, with no transition period during which parts of both systems are live.

Because of its high risk, the immediate cutover strategy requires careful consideration and planning. Installing a system with the immediate cutover strategy is somewhat like performing a high-wire act without a safety net. If problems arise or errors are encountered, there is no net to fall into or old system to fall back on. Another factor making immediate cutover a risky strategy is that it does not allow you to verify the operations of the new system by comparing its output to the old system's output.[1]

Even so, the immediate cutover strategy is sometimes a good—even only—choice. For example, installing a new communications network usually requires the disconnection of the old communications system. Since the new system can be tested thoroughly before cutover and since operating the two systems concurrently is not feasible, immediate cutover is an appropriate strategy. When migration is simple and straightforward or when any other migration strategy would be too costly or too complicated, immediate cutover may be the only feasible choice.

The risk of immediate cutover can be reduced by conducting a final test of the installed system during a slow period or after working hours. In the case of the communication system mentioned above, the network would probably be closed for several hours to all but a few users involved in the final test. This test would be conducted at night or on a weekend so that very few users would be inconvenienced. After the low-volume test, the network would be opened to all users. Similarly, a department store might make a final check of its installed point-of-sale system by conducting a rehearsal run while the store is closed to customers.

Parallel Cutover

With the **parallel cutover** strategy, the new system is run simultaneously with the old until the accuracy and reliability of the new system can be verified. The greatest advantage of this strategy is that it is very low risk; its greatest

disadvantage is that it is very high cost and often requires complicated procedures.

Parallel cutover reduces risk in several ways:

- It allows you to verify the new system's outputs by comparing them to the old system's outputs.

- It builds user confidence in the new system by allowing users to verify it in a production environment.

- It builds user comfort with the new system by enabling the users to become accustomed to its operations.

- It provides a safety net—the still functioning old system—in case the new system fails. This feature is often called **fallback** because it allows you to fall back to the old system if necessary.

The high costs and complicated logistics associated with parallel cutover may make it infeasible. Running both systems simultaneously strains the organization's resources; the same number of people who ran only the old system before must now oversee the operation of both systems. When a manual system is being replaced by an automated one, users must continue to perform all of the old manual procedures while also performing all of the new procedures of the automated system. The increased workload and mental strain of parallel cutover can quickly take its toll. If both the old and the new system are automated, computer resources may be pushed beyond their limits, resulting in increased downtime and turnaround time. These complications increase the likelihood of errors, in effect increasing the risk of system failure while reducing the difficulty of coping with that failure (by providing the old system as fallback should a failure occur).

In summary, several factors can make parallel cutover prohibitively difficult:

- Too many resources may be required.

- Comparing the new system to the old may be too difficult.

- The required procedures or logistics may be too complicated.

- There may be no equivalent operation in the old system (for example, when a manual ticket reservation system is computerized).

The overall risk reduction achieved through the parallel cutover strategy, however, may justify living with these complications and planning for them.

Another way of dealing with these problems is to perform a **modified parallel cutover** strategy. With this approach, the new system can be tested using old data to simulate a historical parallel run. That is, a new computerized accounting system might go live over a weekend, running old but genuine data from the old manual system. The output from the new system could be compared to that generated by the old system to verify the new system's operations. Furthermore, the users could operate the new system in a no-risk environment, thus avoiding not only the stress of running both systems simultaneously but also the anxiety of using the new system to produce output that affects the organization. Although this alternative is feasible only for small systems with few users, it deserves consideration when circumstances permit.

One question that must be addressed when the parallel cutover strategy is adopted is how long to continue the parallel operation before dismantling the old system. The length of the parallel cutover period will vary depending on the complexity of the new system and the consequences of its failure. A short period of parallel operation, perhaps only a few days, is appropriate when the new system is fairly straightforward and errors in its functions are neither critical nor life-threatening. In contrast, when a large organization installs a new accounting system, it may continue parallel cutover through several accounting cycles because the validity of the system's data and reports is crucial to the organization's survival. The adage about ends justifying means comes to mind—the cost and complexity incurred by a more extended parallel cutover period are justified by the peace of mind and risk reduction achieved.

Phased cutover

A migration strategy that achieves a happy medium on the cost and risk scales is **phased cutover**, in which the system is installed in phases. This is similar to the phased development approach that we have been following for system development. Phased cutover differs from phased (or iterative) development in that the first is a migration strategy whereas the second is a development approach, an SDLC. A system being installed with a phased cutover strategy may have been developed by following any development methodology—traditional life cycle approach, agile, or an iterative SDLC.

The phases of migration can be:

- **Subsystems of the system** Each subsystem is cut over separately. For example, when a manufacturer cuts over to a new accounting system, order entry is cut over first, inventory is cut over next, followed by billing, and so on.

- **Organizational units** The system is put into use in only one part of the firm at a time. The parts can be organizational levels, functional areas, geographic sites, and so on. An example is when the Air Force cuts over to a new aircraft maintenance system. The system is installed airbase by airbase.

The order of functions for phased cutover by subsystem is usually determined either by the needs of the users or by the logical progression of functions. In the first case, system functions are installed one by one on the basis of user priorities. For example, when a new marketing information system is being installed, users may determine that the salesperson territory assignment subsystem should be installed first.

Putting Migration in Perspective

Cutover is an exciting time both for the users, who gain full control of the system, and for the developers, who see months of labor finally brought to fruition. Migration signals the end of the project, but it would not be complete without recognition of the outstanding contributions of the key developers, users, and managers. Everyone will be refreshed by a celebration that alleviates the psychological letdown associated with the end of an intense activity. Although post-project recognition ceremonies or celebrations may seem frivolous, they are valuable motivational tools that formally mark the end of a project, revitalize users and developers, and restore some of the stamina needed to take on tomorrow's projects.

Adapted From:
Systems Development: A Project Management Approach
by Ray McLeod & Eleanor Jordan