CHAPTER 4

Supporting Processes

Certification, Rationalization, and Presentation

The Cost of Saying "No"

One of the major IT issues in corporations today is the cost of saying no. Most organizations have designed their IT systems to respond to business and user needs. This is the right way to do it, but in an enterprise system, it is sometimes important to be able to say no to certain requests, especially if they seem unreasonable.

In determining the cost of saying no, you must review who has the right to deny requests. If this right is given to a technician within the corporation, does the proper level of authority come with it? When the technician says no, does the user accept this decision or does he or she "go up the ladder" until someone finally says yes and tells the technician to perform the work anyway?

The cost of saying no is evaluated by determining when the no is not accepted and how people try to countermand the decision. When putting in place an EMF, the entire enterprise must learn the value of no.

So far, several of the key processes required to put the EMF in place have already been introduced: standardization, rationalization, and certification. Standardization will be discussed in more detail in Chapter 5, because it covers the administrative processes required for the new IT environment.

This chapter focuses on the different solutions that are needed to support the EMF. If rationalization principles are not already in place in your organization, you're paying the cost of supporting diverse systems. But, before you can reap its benefits, you must understand why rationalization is required. For this, you need an understanding of software lifecycle management.

The QUOTE System's Understand Phase

We continue the Understand Phase with a look at the pillars supporting the EMF. The supporting processes we examine here will form the final elements of the initial solution we must put in place.

In addition, because we must all live with some degree of software diversity no matter how stringently we apply rationalization principles, you'll need a backup plan: the certification program. This is the program that will provide the most stability on all SPA objects.

And finally, because stability, reliability, performance, and productivity are the ultimate goal, we will cover the Presentation layer in more detail. This will help you understand how you can use active technologies to deliver and manage personalized environments on every SPA object. After all, meeting user needs to fulfill business goals is the first and foremost requirement of any IT system.

Software Rationalization

For an example of the rationale used during this process, see the Case Study on PC software rationalization at the end of Chapter 2.

Rationalizing the Enterprise

Rationalization is the process of reducing diversity within the IT network. Because the purpose of an IT network is to provide an environment where users can collaborate on projects and share information, it is essential to ensure that information sharing is supported at every level of the enterprise.

Most organizations have already discovered the benefits of a single, unified office automation solution. Why? Because standard software tools can reduce costs dramatically while ensuring maximum content reuse.

Organizations that have standardized on Microsoft Office realize cost savings in automating setups, packaging, deployment, development and enhancements, training, and support. In many cases, organizations can build a single, automated setup file that can be personalized according to regional needs. In fact, with Office 2000 or XP, this is even easier to do than ever before because in these versions Office uses a deployment mode that supports a single source for setup that can be easily modified during installation by a transformation file. This transformation file includes anything that may be particular to a specific group within large organizations such as regional settings, localized dictionaries, group templates, and more.

The question becomes, "If you have standardized on office automation and probably on an operating system, why stop there?"

Microsoft Office Retraining

Microsoft Office owes its success to its uniform look and feel. Because all Office applications use the same menus, the same interface, the same Help engine, and the same keyboard commands, user retraining for each of the applications is minimized. In fact, studies show that almost 40 percent of Office learning is reused when someone goes from Word to Excel or PowerPoint. The rest focuses on the particular features of the product itself. For Word, it is word processing; for Excel, it is calculations; and for PowerPoint, it is illustration and presentations.

With the inclusion of the Microsoft Management Console (MMC)—a unified administration console for all .NET Server products—in Windows 2000 and XP, Microsoft brings the retraining concept from Office to systems administration.

The Effectiveness of Rationalization

If you started using IT technology in the Windows 3.*x* days, you've amassed a plethora of different products within your network. Most organizations have. As people moved from Windows 3.*x* to Windows 9*x* to Windows NT, they often spent an enormous amount of energy making sure these products continued to work in their networks (especially with NT!). Does the corporation profit from the use of a multitude of often incompatible products? Does the cost of supporting often obsolete software warrant its use?

The requirement for older software products often comes from the user community. Because everyone fears change to some degree, people like to continue using the tools they're used to instead of learning new ones. But if the EMF is truly for the enterprise, *everything* must be standardized. Thus, organizations should perform a software rationalization to reduce operating costs. The cost savings inherent in a rationalization soon overcome the costs of replacing and upgrading products.

Following are a few tips for performing a rationalization.

- *Get rid of multiple versions of the same product.* Many organizations use several versions of the same product. Microsoft Office is an example. If you've invested heavily in application development with an earlier version of Office, or worse, if your users have developed their own tools with Office, you often don't consider changing these when upgrading to new versions. Some organizations even hold back Office upgrades because they fear the cost of redeveloping or converting existing applications.
- *Select products by function.* A single product should cover any given IT function. If you want to create technical artwork, use Visio. For illustrations, use CorelDraw or Adobe software. For development, use Visual Studio. Every IT function in your network should be provided by a single and unique product per function, unless a specific business requirement demands otherwise.
- *Provide conversion facilities.* If you are managing your IT investment by reducing diversity, you must provide your users with the ability to convert massive amounts of data.
- *Provide training.* One of the major advantages of rationalization is reduced training costs. By combining the

Standardizing Everything

In a .NET EMF, it is important to standardize as much as possible. .NET infrastructures by their very definition can support diversity within specific boundaries.

For a .NET Enterprise architecture to function at optimum capacity, you need to limit diversity as much as possible—not in the variety of your solutions, but in the tools you use to deliver them.

once diluted support budgets of multiple applications, you can finally provide standard training for nontraditional products.

- *Implement software lifecycle guidelines.* Software must be managed within the network. To do so, use the guidelines outlined in the following sections.

Rationalization Costs

There are costs associated with rationalization. When you convert from four or five drawing programs to a single program, you need to retrain users, convert their data, and acquire new licenses for the selected standard. Each incurs costs, but there are cost-reduction strategies.

- Most manufacturers offer competitive upgrade programs. These enable you to trade in your old product licenses to the new standard at a reduced price. In many cases, the cost savings for license acquisitions can range from 40 to 70 percent.
- Several firms market conversion tools. Most good ones include powerful conversion filters into their products. Two conversion strategies are possible: convert as you go or perform batch conversions. Both have their place, but the selection of one or the other depends on your needs.
- User training programs can be performed at reduced costs. Retraining for new products often implies learning how a particular product works rather than relearning everything about the function of the product. If your training programs are focused on the *difference* in the product interfaces, you can greatly reduce the amount of training required.

Batch File Conversion

Many organizations opt for batch conversions because in many cases it is impossible to search within other document formats. For example, if you convert from Micrografx Draw to Microsoft Visio, it is difficult to use Visio to search within Draw's document format.

If document reuse is a priority, you will need to perform batch conversions.

Software Lifecycle Management

All software has a lifecycle. It begins from the moment the software development project is initiated by a manufacturer until the moment the software is retired from the marketplace. For user organizations, the lifecycle focuses more on when it is acquired, when it is deployed, how it is maintained and supported, and when it is retired from the network. Figure 4.1 illustrates the industry adoption process for the manufacturer's product.

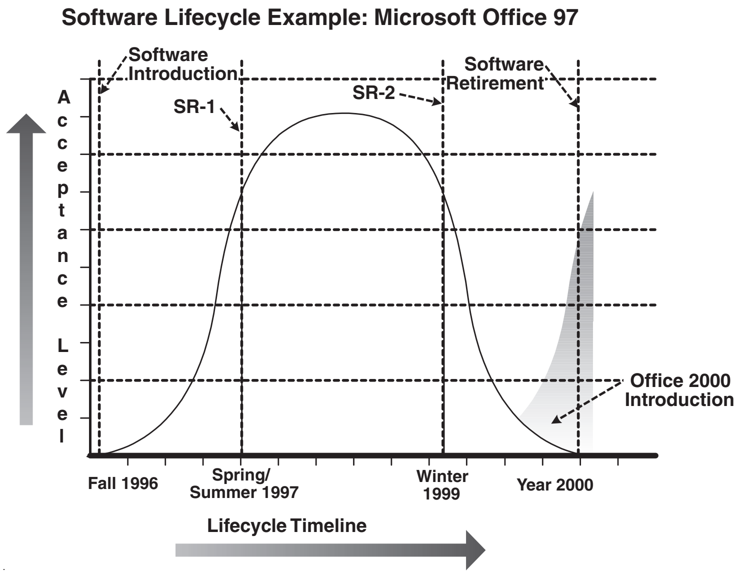


FIGURE 4.1 The Lifecycle of a Software Product. During the lifecycle of a software product, the manufacturer releases several corrective updates. Organizations that adhere to the product at different times during its lifecycle must also manage these updates. Then, once a new version is introduced, the manufacturer slowly retires the product from the market. The corporation must do the same within its own network.

During its lifecycle, most software will require corrections. Manufacturers often call these *service packs* or *service releases*. If an organization adopts a software product before these corrections are released, it will have to deploy the service pack or service release in addition to the deployment of the original product during that product's lifecycle. Figure 4.2 illustrates the corporate software lifecycle.

In addition, software, especially operating systems, tends to be tightly bound to the capabilities of a hardware platform. For example, if you use Windows NT and continue to buy new systems today, you probably realize some of this operating system's limitations. Your new systems come with hot new technologies such as USB or FireWire ports, but you simply can't take advantage of them because NT can't. This alone is a justification for a move to Windows 2000 or XP.

Software tends to grow with hardware capabilities. You need to keep up if you want to make the most of your IT investments.

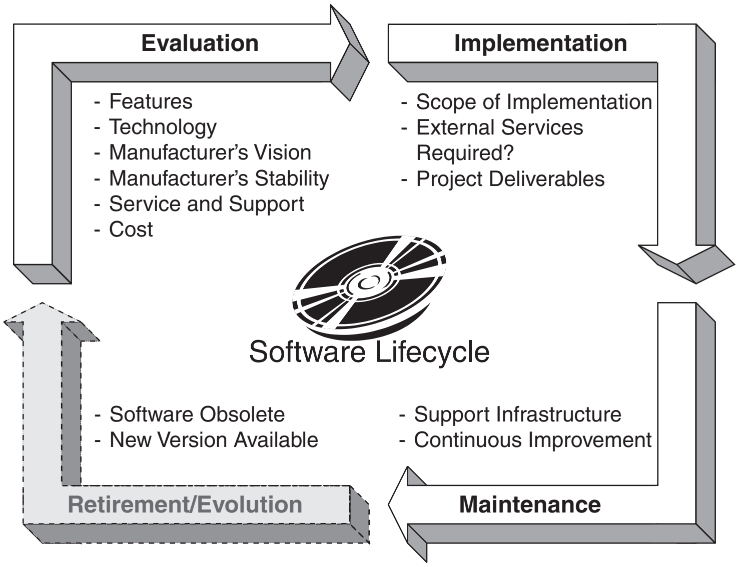


FIGURE 4.2 Corporate Software Lifecycle Management. Corporations must manage an internal software lifecycle that differs, but is related to, the software manufacturer's development lifecycle.

Windows 3.*x*, Windows 9*x*, Windows NT, Windows 2000, and Windows XP all include different capabilities in terms of system integration. Software products built for these systems take advantage of these capabilities. This becomes important when you discover that products designed for Windows 2000/XP fully support the 32-bit capabilities of advanced Pentium processors and can simply not be installed on any older version of Windows.[1] Lifecycles between operating systems and application software are closely related. Figure 4.3 illustrates this relationship.

Defining Software Obsolescence

One major part of the software lifecycle is the definition of its state of obsolescence. For corporations, *obsolescence* should be defined as "a software product that is no longer supported by the manufacturer, that is no longer available

[1] Though most manufacturers create a single installation tool for multiple operating systems, it is not the same product that is installed on NT, 95, or 2000.

on the market, and/or that is no longer required by corporate business processes.”

In a Software Lifecycle Management Model, when a software product is obsolete, it must be either retired or upgraded. Upgrades can take the form of either the purchase of a new version of the same product or the deployment of a replacement product. As a result, an important part of the software management process is the software upgrade cycle.

To support the software upgrade cycle, you need additional processes and principles. You need to put in place standard methods that ensure you properly gauge value and benefits from a product upgrade versus its implementation costs. Cost/benefit analyses are crucial at this stage.

But these are not the only reasons to upgrade. User expectation is also a major factor. Software manufacturers do a good job of creating user desire through product advertising. IT managers are then faced with user requests for new versions of products. The Software Evaluation Model must also take these expectations into account.

The Cost of Obsolescence

Maintaining obsolescence bears its own costs. If an organization continues to use a product despite its obsolescence, it often faces hidden costs.

- Using a product for which the manufacturer has removed support means assuming these support costs internally.
- Older versions of software do not take advantage of newer operating system capabilities. If an organization decides to keep and maintain an older product, it is potentially limiting productivity.
- Older products lack functionality. They are often not integrated with newer companion products.
- Older versions can limit growth. If a corporate system is built around an obsolete product, the corporation can do little to improve its capabilities.

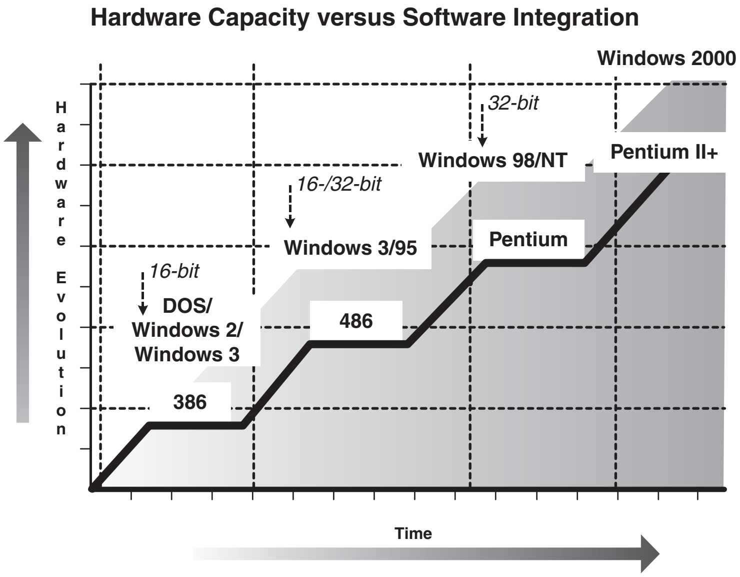


FIGURE 4.3 Hardware and Software Relationships. Software and operating system capabilities are closely related to hardware platforms.

Guidelines for Software Selection During Rationalization

Several factors influence the selection of software during a rationalization process, including

- *Purpose and scope:* What is the purpose of the product? Does it provide a unique functionality? How many users require the product?
- *Security and error protection:* How stable is the product? How secure is its data?
- *User-friendliness:* What is the product's learning curve? Does its interface operate like other products already in use? How much training is required?
- *Functionality:* What are the product's features? How easy are they to use? Do they respond to your functional needs?
- *Performance:* What is the speed of operation of the product? How realistic are the manufacturer's suggested minimum requirements?
- *Versatility:* How does the product compare to its competitors?
- *Compatibility:* How compatible is the product with older versions, with other products already in use, and with hardware platforms?
- *Installation:* How easy is the software installation process? Is it similar to other products in the network?

Positive response to all of these factors means that the product should be retained by the organization. Negative responses must be explained and documented.

Guidelines for Software Lifecycle Management

During the software lifecycle, four factors will initiate the software review process.

- *Software obsolescence:* The product is being retired by the manufacturer.
- *Projects:* A product is required to support business objectives and a project is launched to implement it.
- *New product versions:* New, feature-rich versions incite the organization to upgrade.
- *Corrective maintenance:* The current version has bugs that are repaired with a new version.

Given that these situations initiate a software upgrade, organizations should arm themselves with processes that give them a measure of control over the software maintenance cycle.

Software Obsolescence Rules The first thing to do when implementing a software lifecycle management program (SLMP) is to outline the rules or guidelines that will provide a framework for the program. These should include the following:

Rule 1: Coverage

Every software program that is in use in the network and that has a clientele of more than ten users[2] should be covered by the software lifecycle.

Rule 2: Software Owners

Every software program in use should have a designated software owner. This owner is responsible for managing and maintaining the SLMP for the product.

Rule 3: Exceptions

Every software program that is included in the SPA kernel is treated in an exceptional way because its evolution is tied to the evolution of the entire kernel.

Rule 4: Obsolescence

Software obsolescence occurs when

- The manufacturer does not support the software anymore.
- Two newer versions than the one in use in your network are available on the market.
- The manufacturer announces that it is retiring a product from the marketplace.
- Upgrades for the SPA infrastructure require an upgrade of the product.
- The software product does not work on a new operating system.

The software upgrade cycle is mandatory for any software product that is deemed obsolete.

[2] Ten users is the limit where cost exceeds benefits from the SLM program, though some corporations opt to include *every* software product into the program, no matter what the cost.

Rule 5: New Versions

Every new release of a product that fits within the guidelines of Rule 1 must undergo a software evaluation process. This process includes the following:

- Evaluation of software reviews published in IT industry publications
- Evaluation of the recommendations for use from consulting firms such as Giga Group, Gartner Group, and others
- Evaluation of the product's life expectancy predictions
- Preparation of an evaluation report recommending whether the software upgrade cycle should continue for this product

Rule 6: Software Update Request

Every software update request from any source must be accompanied by a valid justification for the business value of the move.

Managing Kernel Software For the SPA kernel software, there are three additional rules.

Rule 7: Kernel Upgrades

Because the kernel is treated as a single, unified whole, upgrades or corrections for the kernel should be packaged together and released on a regular basis. The kernel deserves an approach that is different from other applications because it is critical to the business.

Rule 8: Corrective Maintenance

Because the kernel is critical to the operation of the network, service packs released for any product contained within the kernel *must* be applied. Before its application, though, the corporation should wait at least two months to ensure its stability as witnessed by the industry and through internal testing. If it is deemed satisfactory, this service pack should be included within the next kernel upgrade.

Rule 9: Software Interdependence

Every new software product released in the network must undergo a coherence evaluation. This ensures that the product is coherent with the components of the kernel and will not destabilize it.

Managing Nonkernel Software Three final rules apply to nonkernel software products.

Rule 10: Delivery Management

Because the introduction of a new product or a new version of a product affects the network, it should always be treated as a project with supporting budgets and processes.

Rule 11: Software Certification

Every software product that has a user base of greater than ten must undergo the software certification process. This process is described in the following section.

Rule 12: Software Lifecycle Management Program Evolution

The SLMP must be managed and periodically reviewed to ensure that its evolution keeps pace with software market trends.

Software Certification

For an example of the benefits reaped from the software certification process, see the Case Study at the end of Chapter 3.

Certification Processes

Now that you have reduced the diversity in your network, you must still arm yourself with processes that ensure stability. No matter how stringent your guidelines, you will always be faced with a certain amount of diversity. This is simply due to the breadth of functional requirements within any IT network. This is where the certification process comes into play.

The Evolution of Software

Software has changed over the years. In the days of DOS and Windows 3.*x*, software often consisted of a single, unified executable that simply required a copy from the installation disks to the hard drive.

Today, software is much more complex. Software manufacturers have changed their product design from single executables to the creation of *master programs* that orchestrate the operation of all of a product's components.

This evolution to master programs is an important development. In order to manage memory more effectively, software manufacturers have divided their products into a series of subcomponents loaded into memory only when users require it, which frees up memory for important tasks.

For example, when you're working with a word processing program, printing is not required all the time. So

Remember DOS?

DOS was a very limited OS. People who produced software for DOS had to worry about every single component of the computer—keyboard drivers, screen drivers, mouse drivers, and so on. This was because DOS provided very limited services to manage the PC. Software manufacturers had to provide every function if they wanted their product to work.

Windows changed all of this (though it wasn't the first program to do this on the PC—remember GEM?) by providing a single, unified environment to manage all of a PC's processes. With Windows, software manufacturers only had to ensure their programs worked with Windows interfaces. This enabled them to concentrate on their product's features.

Unfortunately, this caused another problem: DLL hell.

the printing module (often called a *dynamic link library,* or *DLL)* is only loaded when you want to print. Then, when printing is complete, the DLL is removed from memory to clear it up for more important functions.

In addition, because the OS provides most functions, software applications must often call upon system functions to work. Printing, for example, is provided by the OS, whereas the product only needs to know how to call upon the function (see Figure 4.4).

Facing DLL Hell

The problem with the DLL[3] approach is that it is now very difficult for software manufacturers to ensure that all the components that ship with a product are up-to-date. When software is installed on a system, the components it includes often replace existing components that may be more recent. This causes instability within the system.

For example, say Company A decides in 1998 to develop a new product. The product development cycle will take two years to complete before the product reaches the market. They ensure that in 1998 they have the latest and greatest systems and that every OS component is up-to-date.

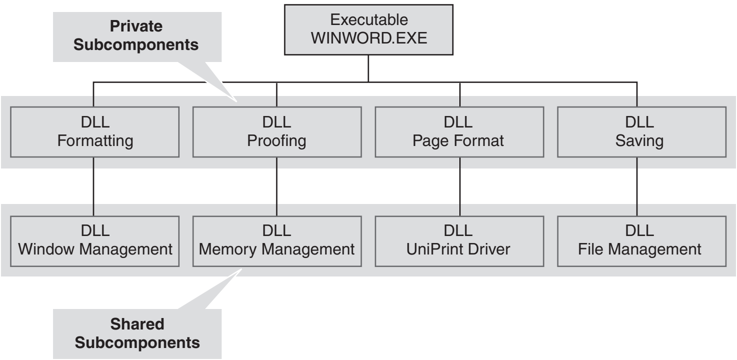


FIGURE 4.4 The Heart of an Executable. Software programs today include a master program component that calls upon either private or shared subcomponents as required during operation. Figure 4.5 illustrates how this interacts with the PC memory management process.

[3] DLLs are not the only components that may cause conflicts and instability, but they are the most well known. Conflict resolution tools can identify all of the other types of components.

The problem arises when, while Company A is in development, operating systems continue to evolve. In 1998, Company A may have been using Windows NT with Service Pack 4. Meanwhile, Microsoft released Service Packs 5, 6, and 6a along with numerous hot fixes.[4] In February 2000, Microsoft released Windows 2000. In June 2000, Microsoft released Service Pack 1 for Windows 2000.

If Company A does not ensure that it keeps all of its systems up-to-date and does not perform compatibility testing of its product when it does upgrade, the product that hits the marketplace in July 2000 will definitely include components that are older than those that may exist on your system. This process is illustrated in Figure 4.6.

The software industry has taken steps to reduce the impact of this problem. For one thing, many manufacturers ensure that all new versions of a DLL support all of the functions of all of the older versions. So even if a product thinks it requires version 1, it will still work with version 2 because the new version supports all of version 1's functions—it is *backward compatible.*

Unfortunately, this does not work with every product. Some manufacturers do not provide backward compatibility for their components. If you want to run two conflicting products on the same operating system, you must figure out a way to make two versions of the same component work on the same system, otherwise known as *side-by-side operation*.

Table 4.1 describes the differences between backward compatibility and side-by-side operation.

As you can see, the DLL problem can become a major headache. It is often this very problem that causes instability in Windows systems.

The Fear of the Unknown

Like most of us, software manufacturers fear the unknown. Because they know their product works with version 1 of DLL A, they will tend to include this DLL in the installation routine of their product.

You *always* face a problem when you have already upgraded your systems to use version 2 of DLL A because during installation their product will replace your version of the DLL with their own.

It gets worse. Given that every product has dependencies on system components, you may have one product that requires version 2 of a DLL and a "new" product that requires version 1. What can you do?

Windows 2000 and DLL Hell

Microsoft is conscious of the DLL problem because they have received numerous support calls relating directly to it. This is why Windows 2000 includes for the very first time several technologies designed to avoid DLL hell, and Windows XP expands on these technologies.

[4] A *hot fix* is a software correction that focuses on a single problem. A *service pack* is a collection of software corrections packaged into a single installation file.

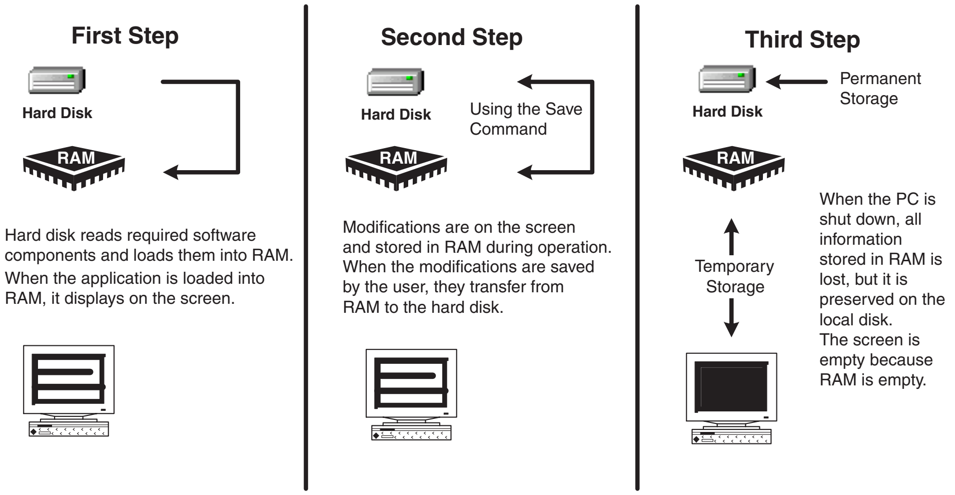

First Step
Hard Disk
RAM
Hard disk reads required software components and loads them into RAM. When the application is loaded into RAM, it displays on the screen.
Second Step
Hard Disk
Using the Save Command
RAM
Modifications are on the screen and stored in RAM during operation. When the modifications are saved by the user, they transfer from RAM to the hard disk.
Third Step
Hard Disk
Permanent Storage
RAM
Temporary Storage
When the PC is shut down, all information stored in RAM is lost, but it is preserved on the local disk. The screen is empty because RAM is empty.

FIGURE 4.5 The PC Memory Management Process. Most users have difficulty understanding how rapid access memory (RAM) is related to permanent storage on the hard disk. Note that software developers use the DLL approach to reduce RAM usage.

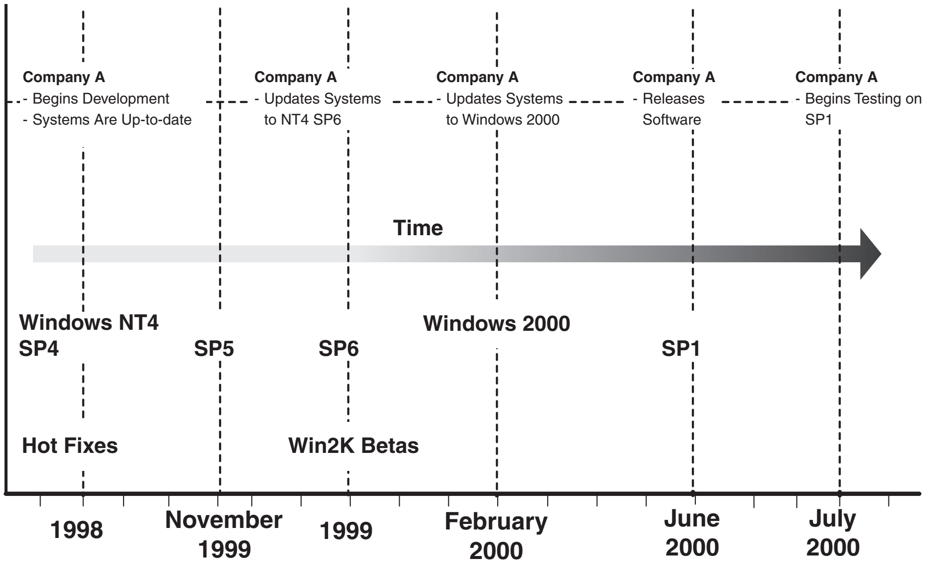


FIGURE 4.6 Development and OS Timelines. Software Company A begins its development cycle with updated systems. As time passes, the company updates its systems, but it is difficult to keep up. One of the ways Software Company A can keep up-to-date is to use beta versions of upcoming systems to test the compatibility of its solution.

TABLE 4.1 Backward Compatibility versus Side-by-Side Operation

Backward Compatibility	Side-by-Side Operation
Requires consistency of behavior and interface over time.	Interface and behavior change over time.
All versions can and must share state.	Requires isolation of state by version.
All applications use the "latest version" of the component.	Applications all use their own version of the component.
Applications support the activation of the component by its absolute name.	Applications call specific component names.
Service is straightforward.	Service is complex because care must be taken when changing system components.

System Security

Like Windows NT, Windows 2000 uses a 32-bit file system called NTFS. The advantage of this system over its predecessors is that every object stored in the system includes *attributes*. These attributes can contain security features—security features that are different for users, power users, and administrators. The greatest limitations are applied to users. Because users only operate the system, they only need read and execution permissions for every system component. In this way, NTFS "protects" system and application files by restricting access to those files.

In Windows NT, users were given too much leeway because software integration was not controlled effectively. Many software products would install into and require constant read and write use of the system directories. Giving users these rights would open the system to potential damage.

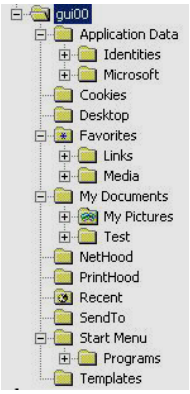
Realizing this, Microsoft released the Zero Administration Kit for Windows NT. This kit provided corporations with the tools to increase system lockdown to further limit user access. But this system was complex to use, and organizations often had to invest in its management.

With Windows 2000, Microsoft changed the nature of the NTFS system lockdown. They added further restrictions

Blocking Software Installations by Users

The security features of NTFS enable administrators to block users from installing software. Many users see this as a limitation of their "rights" on a PC. But this is not the case. Because a PC is a corporate asset that provides support for business processes, it must be stable. If a corporation puts in place a strategy for managing software stability within its network, it must restrict users from installing uncontrolled components on their PCs. This is simply because these uncontrolled components will most likely damage critical system components and destabilize PCs.

Conversely, users must have the right to choose the products they need for their job functions. By providing a self-serve center for approved and controlled products, the corporation meets both user needs and the corporate software management strategy.



- gui00
  - Application Data
    - Identities
    - Microsoft
  - Cookies
  - Desktop
  - Favorites
    - Links
    - Media
  - My Documents
    - My Pictures
    - Test
  - NetHood
  - PrintHood
  - Recent
  - SendTo
  - Start Menu
    - Programs
  - Templates

In Windows 2000/XP, the user profile includes every element that is modifiable by the user.

to users and changed the way applications worked with the OS. As a comparison, users in Windows NT have the same rights that power users do in Windows 2000. Today, users have significant restrictions within the operating system directories and within application directories.

Software that is designed for Windows 2000/XP does not install any component in the system directories. All of its components go into its own application directory. In addition, every component that is modifiable by a user (configuration settings, user preferences, and so on) is stored within the directories containing the user profile. Here users rule and can read and write to their heart's content. Moving modifiable components to the user profile is a good strategy because critical system and application files are protected for all users. If users damage something within their own profile, it simply needs to be erased and re-created.[5]

Side-by-Side DLL Management

In addition, Windows 2000 and Windows XP include side-by-side DLL management. This means that if two applications require the use at the same time of two different versions of the same DLL, Windows 2000/XP can load both into memory but into separate memory spaces. Side-by-side DLL management avoids conflict and relieves organizations from having to manage this process.

Windows System File Protection

Windows 2000 also includes System File Protection (SFP). This feature stores a backup copy of critical system files (within the DLL Cache subdirectory). A special agent is constantly watching the system directories. If, during the installation of a new application, a critical system DLL is replaced and the original system DLL is crushed, this agent automatically replaces the new file with the original and proper file.

In addition, SFP enables only the OS to update files within these directories.

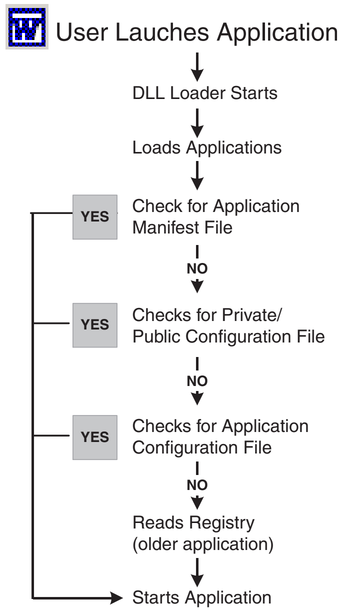
Windows Installer

For software applications, Microsoft introduced the Windows Installer service.[6] This system service is designed to

Windows XP and DLL Hell

With Windows XP, Microsoft enhances the DLL hell management process with a new folder to store side-by-side DLLs and virtual DLL redirection. DLLs no longer need to register within the Windows Registry. Applications designed for Windows XP now include descriptive files that are read by XP's DLL Loader before loading the application. This means that in a pure XP environment, DLL hell may become a ghost of the past, but it will be some time before organizations can move to pure XP environments.

In addition, because it is no longer based on DOS, XP includes new virtual Windows and DOS compatibility environments, supporting even more applications than Windows 2000.


User Lauches Application
DLL Loader Starts
Loads Applications
Check for Application Manifest File
YES
NO
Checks for Private/ Public Configuration File
YES
NO
Checks for Application Configuration File
YES
NO
Reads Registry (older application)
Starts Application

Windows XP goes beyond the capabilities of Windows 2000 to manage the DLL process as is demonstrated in this flowchart.

[5] Care must be taken during this operation because the profile also stores user preferences. These must be recovered after the profile has been re-created.

[6] This service was first delivered with Office 2000, but it was refined with Windows 2000 and later, Windows XP.

assist in the control and management of the software lifecycle on Windows 2000/XP systems. For stability purposes, Windows Installer has the capability to provide the same type of protection to application files that the SFP provides for system directories.

Applications whose installation is integrated with Windows Installer include an installation database. This database is stored on the system during installation. Then, every time an application is launched, Windows Installer verifies its consistency based on the database settings. If everything is okay, the application starts normally. But if critical components are missing or damaged, Windows Installer searches the database for the location of the original installation file and repairs or replaces the component, and then launches the application normally. This process is called *self-healing*.

Windows Installer is not perfect, but it does provide some measure of protection for applications. But beware: For self-healing to work, installation files must *always* be available to the Windows Installer service. This means that installation sources must permanently remain available in the network.

Certified for Windows 2000/XP

With all of these changes, Microsoft has implemented a new approach for software manufacturers to integrate their products with Windows 2000/XP: the Certified for Windows 2000/XP initiative. A product that is certified for Windows 2000/XP supports the following features:

- Its DLLs are designed to be backward compatible.
- All of its operation files are stored within an application directory in the Program Files folder.
- Its installation process is integrated with Windows Installer (the extension of the installation file is then .MSI).
- It can be delivered to SPA objects (PCs or servers) through Windows 2000/XP's Active Directory service.
- All user-modifiable files are stored in the user's profile directories (within the Documents and Settings folder).

In fact, when a product is certified for Windows 2000/XP, the product will provide as much stability as is available from the OS.[7]

Viewing System File Protection at Work

To view the System File Protection (SFP) feature at work, log on to Windows 2000 as an administrator. Navigate to the WINNT\SYSTEM32 directory. Locate CALC.EXE (the calculator) and delete it. Empty the Recycle Bin.

SFP should replace it within seconds.

Windows Installer

Windows Installer does much more than just provide self-healing capabilities. It is a complete system designed to manage software on a SPA object. It can provide deployment through Active Directory, Windows 2000's central system and user management service. It can personalize installation settings through transformation files applied during setup. It can provide clean uninstallation of a software product. It can support the application of corrective maintenance (service packs) during the product's lifecycle.

[7] "Certified for Windows 2000" means much more than just the features outlined here. For a complete description of the requirements, go to http://msdn.microsoft.com/certification/appspec.asp.

Stabilizing Windows Technologies

The problem is that not all applications on the market today comply with Windows 2000/XP standards. Until they do, you will need to have pre-Windows 2000 applications coexisting with Windows 2000/XP–certified applications.

This may be cause for concern, but not if you apply a software management strategy within your network. Because the problem has been defined, you now know where to look for application and system instability issues.

Certifying Software: Managing DLL Conflicts

To face the DLL problem, you can implement a software certification process within your corporation. This process is based on the identification of all of the components you install on your systems and the repair of damaging applications. If an application includes a DLL that is older than the one stored in the system, you can take the following actions.

- Upgrade the DLL during the installation of the product. This works if the DLL is backward compatible.
- Move the DLL to another directory. This will take advantage of Windows 2000's side-by-side operational capabilities.

Other conflicts can arise, but their impact is not critical. They can be repaired on an as-needed basis. The entire DLL management process is outlined in Figure 4.7. Wise Solutions, Inc.,[9] a software manufacturer specializing in software delivery technologies, was the first to formalize this process.

Because the SLMP requires all products that have a user base larger than ten to be managed centrally, corporations must use automated techniques for installation. Installation automation is required because the corporation does not want users to be faced with configuration decisions during installation. Most products aimed at the corporate marketplace include technologies that enable preconfiguration of installation parameters. Windows 2000/XP applications au-

Hardware Certification

Windows 2000/XP includes a hardware certification program. You can find most of the products that have been certified for Windows 2000 in the HCL available at http://www.microsoft. com/windows2000/server/howtobuy/upgrading/compat/search/devices.asp. If the product you want is not on this list, ensure that the manufacturer provides a Windows 2000 driver for the product; otherwise, you may destabilize your systems.

Running Pre-Windows 2000 Software

Pre-Windows 2000 software works on Windows 2000. But, because of the changes in NTFS security and because these applications have a tendency to store components within system directories, you will need to "loosen" system security. Otherwise, users will not have the ability to execute these applications, because in Windows 2000, they do not have write access to these directories.

To run most pre-Windows 2000 applications with user accounts, you must apply the Compatibility Security Kit to your systems. To do so, run the following command:

```
secedit /configure /cfg ↵
compatws.inf /db ↵
compatws.sdb
```

This command will modify the security level of Windows 2000 to make it operate like Windows NT.[8]

[8] The ↵ symbol represents word wrap on the command.

[9] You can find more information on Wise Solutions at http://www.wisesolutions.com/.

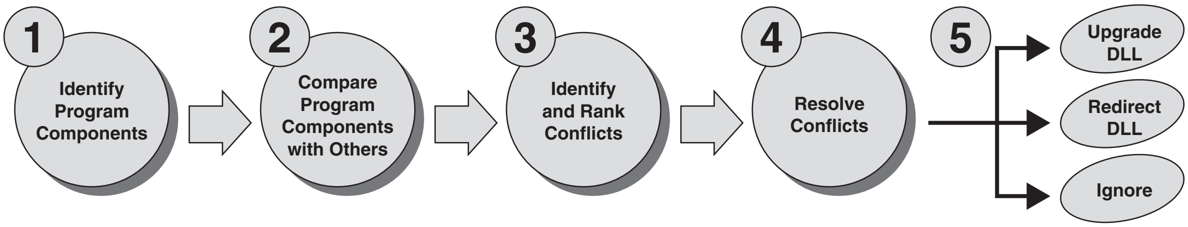

1
Identify Program Components
2
Compare Program Components with Others
3
Identify and Rank Conflicts
4
Resolve Conflicts
5
Upgrade DLL
Redirect DLL
Ignore

FIGURE 4.7 The DLL Conflict Management Process. Managing DLLs involves identifying conflicts and taking proper action.

tomatically include such a system because it is one of the functions of the Windows Installer.

In some cases, corporations will need to deploy and manage software that does not include automated setup technology. This is generally referred to as *software repackaging*. Software repackaging consists of taking a "snapshot" of the system before installation, performing and customizing the installation, and taking a snapshot after installation. A software-repackaging program records all of the changes and inserts them into an executable that automates setup.

These tools are useful, especially when the corporation needs to deploy internally developed software.

It is during repackaging that conflict detection can occur. Here the packaging tool inventories all of the packaged components. This inventory is stored in a database containing all of a system's components. The database is then checked for conflicts. If conflicts are detected, action can be taken to repair damaging components.

Proactive and Reactive Conflict Management Strategies

It is clear that with its new features, Windows 2000/XP improves stability. But for applications, it does so only for those that are certified for Windows 2000/XP. Older applications are not integrated with Windows Installer because their setup is not integrated into this service.

It is possible, though, to integrate older applications into the Windows Installer service at the repackaging phase. Every application that is repackaged using a Windows Installer–compatible packager will gain some, but not all, of the features of a Windows 2000–ready product. It will, for example, have self-healing, but it will not operate with simple user rights. It will also take advantage of Windows

Wise Solutions' Conflict Manager

Wise Solutions, Inc., offers a product called InstallManager. This product is designed to perform installation automation. Within InstallManager is a module called Conflict Manager. Conflict Manager uses a database to store all component information and also provides a Conflict Resolution Wizard to repair damaging components.

Lanovation Prism Pack Enterprise Edition

Lanovation was the first to introduce a single, integrated product that managed both DLL conflicts and created Windows Installer (MSI) installation files. Their product is very simple to use. There is very little training required, so long as technicians understand the concepts of DLL management and Windows Installer integration.

This product is definitely a boon to the software lifecycle and certification processes. More information is available at http://www.lanovation .com.

Managing Older Applications

Most organizations moving to Windows 2000/XP today will need to live with three types of applications.

- 8- and 16-bit applications from DOS and Windows 3.*x*
- 32-bit applications that are not ready for Windows 2000/XP
- Windows 2000/XP–certified applications

The latter are by far the most stable, but you need to wait for manufacturers to update their products.

In order to properly manage these applications in the network, use the following three installation directories:

- APPS16 (for DOS and Windows 3.*x*)
- PreWin2K (for older, 32-bit software)
- Program Files (for certified products)

Then migrate applications from the two first directories as you update them. You will be able to destroy these directories when they are empty.

Veritas WinInstall LE

WinInstall LE is located on the Windows 2000 Professional installation CD. It comes with a 60-day trial license.

Component Inventories

Eventually, all applications that want to make it in the Windows market will be designed for Windows 2000/XP. As a result, repackaging may no longer be necessary. But it will always be vital for organizations to inventory components. Nothing is better than knowing exactly what is out there. Then, if problems arise, you will at least know what you have.

Installer's reactive DLL management techniques (if the DLL is the same as another already loaded, load this one into a different memory space).

Several software manufacturers offer repackaging tools—for example, Wise for Windows Installer from Wise Solutions, Inc.; WinInstall LE or WinInstall 2000 from Veritas, Inc.; InstallShield for Windows Installer from InstallShield Corporation; and Prism Pack from Lanovation, Inc.

Using both repackaging for Windows Installer and repackaging conflict detection techniques enables corporations to be both proactive and reactive when designing Windows 2000/XP networks. These techniques proactively repair applications *before* they are deployed in the network. And, if some conflicts are missed and deployed anyway, they reactively support conflicts *after* deployment through the Windows Installer service.

A Certification Approach: Complete DLL Management

Introducing a certification process is not much more costly than managing software in traditional ways. Most medium- to large-sized organizations already have software repackaging facilities because they want to take advantage of central deployment strategies.

Identifying and correcting DLL conflicts while packaging means adding two additional processes to the traditional software repackaging approaches.

- Inventory components to identify conflicts
- Repair damaging components

These two processes fit very well into most packaging methodologies because testing is already covered (if a software is repackaged, its installation must always be tested before deployment). With conflict detection, testing changes slightly to verify if either the backward compatible or the side-by-side strategy is required. Figure 4.8 illustrates this process.

Using a Certification Center

In this process, the software repackaging center becomes a certification center. Its nature is permanent and must be staffed as such. Certification activities will include kernel

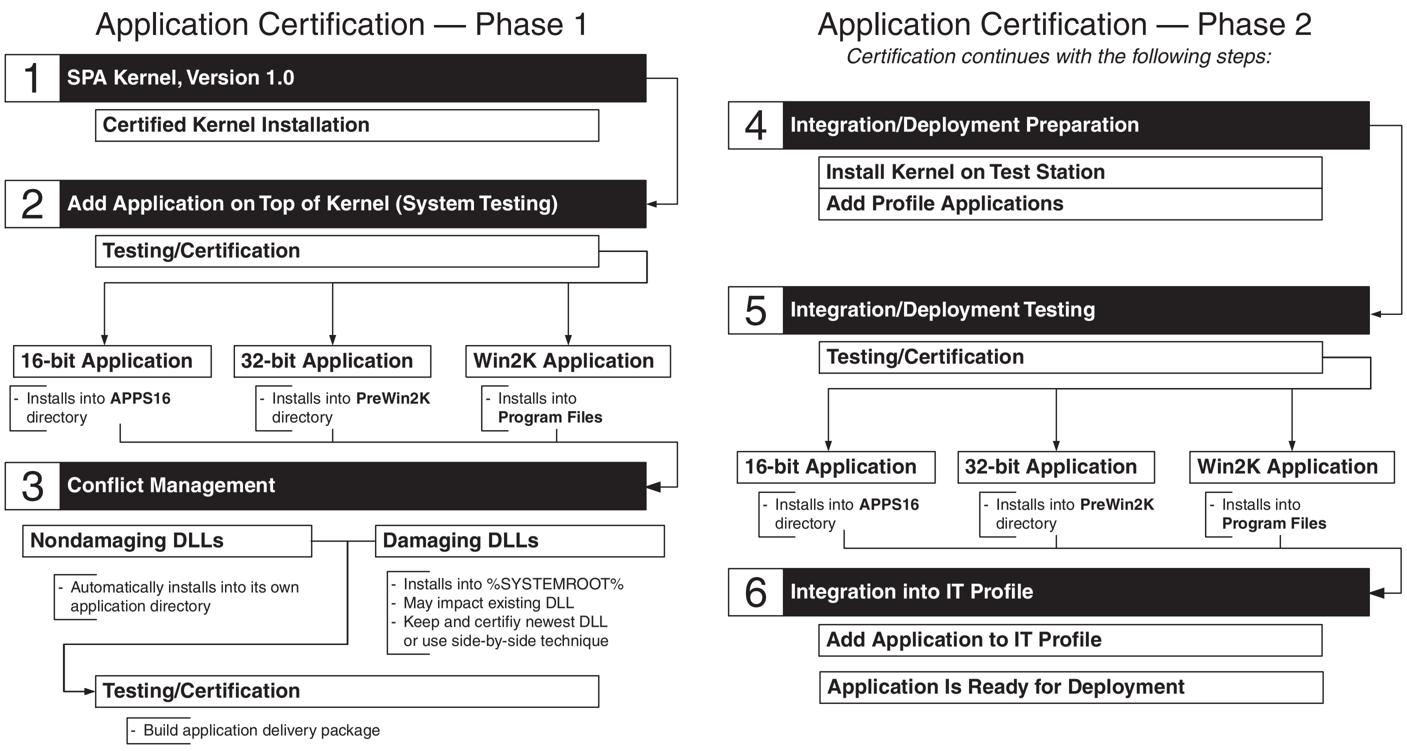

Application Certification — Phase 1
1 SPA Kernel, Version 1.0
Certified Kernel Installation
2 Add Application on Top of Kernel (System Testing)
Testing/Certification
16-bit Application
- Installs into APPS16 directory
32-bit Application
- Installs into PreWin2K directory
Win2K Application
- Installs into Program Files
3 Conflict Management
Nondamaging DLLs
- Automatically installs into its own application directory
Damaging DLLs
- Installs into %SYSTEMROOT%
- May impact existing DLL
- Keep and certifiy newest DLL or use side-by-side technique
Testing/Certification
- Build application delivery package
Application Certification — Phase 2
Certification continues with the following steps:
4 Integration/Deployment Preparation
Install Kernel on Test Station
Add Profile Applications
5 Integration/Deployment Testing
Testing/Certification
16-bit Application
- Installs into APPS16 directory
32-bit Application
- Installs into PreWin2K directory
Win2K Application
- Installs into Program Files
6 Integration into IT Profile
Add Application to IT Profile
Application Is Ready for Deployment

FIGURE 4.8 The Software Certification Process. The application certification process involves six major steps.

management and software lifecycle management. Every software addition into the network must pass through this center. No deviation is allowed. The return on investment for the certification center is provided by the increased stability of software in the network. Payback is also evident in a significant decrease of support calls related to nonfunctioning software.

Certification Center Processes

Several processes support the certification center. The first is the software certification process. It includes the following steps:

1. *System testing:* Prepare the application for automated installation.
   a. Perform software configuration and installation process validation to determine if the installation fits within standards.
   b. Perform repackaging or package analysis. If the application is packaged to Windows 2000/XP standards, proceed with component inventory and conflict detection and resolution. If the application is not ready for Windows 2000/XP, repackage it, inventory components, and perform conflict detection and resolution.
   c. Finalize the package.
   d. Move on to system testing.
   e. Perform system tests.
   f. Document the installation.
2. *Integration/deployment preparation:* Prepare and test the remote installation of the application.
   a. Update the deployment mechanisms.
   b. Deploy on test systems along with profile contents.
   c. Update the documentation.
3. *Integration testing*: Test application cohabitation.
   a. Test the operation of the application within a complete user profile.
   b. Update the documentation and have it approved by the software owner.
4. The application is ready for deployment.[10]

Deploying MSI Software in the Enterprise

Software deployment in a large enterprise is difficult because of all the different source locations you need to manage. It becomes even more difficult when you include self-healing capabilities. Self-healing means that the source of the installation must always be available. Each different site must have a different source address. This may be simple for SPA objects that are always linked to the network, but it is more complex for portables because they must carry the source files with them.

One of the best methods is to use a variable on the SPA object. Here's how it works:

1. Prepare all of your installation packages using a variable such as %Source%.
2. On the preparation systems, define this variable as the source of the installation (Set SOURCE=\\Server\Folder).
3. Define a variable on the SPA object. This variable can map to different locations on different SPA objects, but your installation packages all point to the same destination: %Source%.

For example, on a portable this variable can be set to a local, hidden partition on the hard disk; in a region, it can be a local server; in the certification lab, it can be the staging server; and at headquarters, it can be a central server.

However, if you are using Windows XP, all you need to do is redirect the software source URL.

[10] There is an additional acceptance testing phase that will be covered later.

The next process is the kernel update process. This process singles out the kernel from any other software that is in use within the network. This is because the kernel is the only globally used component in the network. Because every user requires the kernel, its level of stability must go beyond that of other applications.

This means that the kernel must be updated on a constant basis. Because there are a number of different products covered, a change request and evaluation process must be put in place. This change request process uses the rationalization guidelines outlined previously to evaluate the validity of the request. If approved, the change is delivered in the next kernel update.

Given the rate of change in the software industry and the number of different software components that often reside in the kernel, it is essential to manage this process on a four-month basis at the very most. Waiting longer for update collection creates packages that are very unruly. Deploying more frequently is too expensive.

Figure 4.9 illustrates the kernel update process.

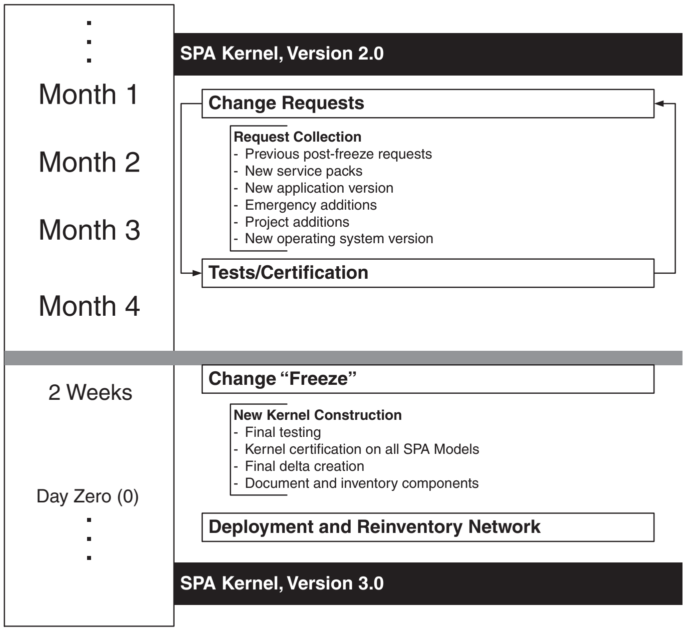
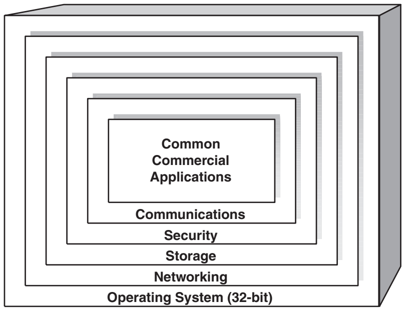


FIGURE 4.9 The Kernel Update Process. The SPA kernel update process should occur every four months. One month before release, a freeze is applied to change requests in order to allow update preparation.



The kernel includes all of the components required for the operation of a basic SPA object. It is discussed in more detail in Chapter 5.

Rate of Updates

Three updates a year has proven to be optimal for the kernel in all of the organizations using this process to date.

Certification by Profile

Even with rationalization practices, organizations often have several hundred software products to support and manage in the network. Therefore, it is important to regroup applications by user type. These IT profiles[11] will include every application that is common to a group of users performing the same type of task. This regrouping by IT role or function simplifies certification testing (only those applications most likely to cohabit on a system will be tested) and deployment (applications are automatically deployed to users that fit the IT profile).

In addition to kernel updates, the certification center is responsible for software update validation and preparation. This means applying the same processes used with the kernel to every managed software product.

Though 90 percent of applications are managed because they have a population greater than ten users, it is also necessary to put in place standard procedures for the other 10 percent. Applications with less than ten users do not warrant certification because if they have conflicts, they do not impact many people. In addition, so long as they are designed to work with Windows 2000/XP, corporations can use the reactive approach to DLL management. Their installation can be manual, but it should be structured. Specific procedures for manual software installations should be put in place and monitored by the certification center. This is called the “less than ten” management process.

Finally, the certification process must support two additional situations: emergency updates and project requests. Both situations require a shorter timeline for deployment. If a fatal bug has been missed during testing and has been deployed by mistake, it is vital to perform an emergency update deployment. This deployment must have a very short timeframe because vital operations are nonfunctional.

In addition, when projects are launched, they often require newer versions of components or components that are not already in use in the network. This situation also requires a shorter timeframe. Figure 4.10 displays these processes.

All of the processes in the certification center help maintain stability within your network. These and the software rationalization processes must become part of your everyday operations if you want to profit from their benefits.

Certification and Consultant PCs

One of the questions you will be faced with when you implement your certification program is “What do you do with systems that operate in your network but don’t belong to you?”

Continued next page

[11] In some cases, user application profiles are called *software configurations* or *software configuration management*.

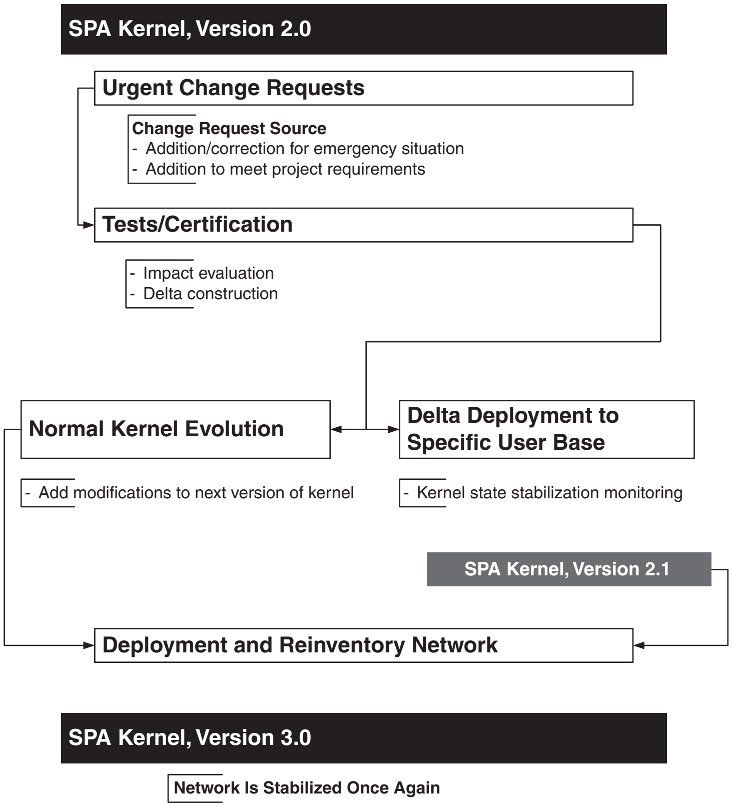

SPA Kernel, Version 2.0
Urgent Change Requests
Change Request Source
- Addition/correction for emergency situation
- Addition to meet project requirements
Tests/Certification
- Impact evaluation
- Delta construction
Normal Kernel Evolution
Delta Deployment to Specific User Base
- Add modifications to next version of kernel
- Kernel state stabilization monitoring
SPA Kernel, Version 2.1
Deployment and Reinventory Network
SPA Kernel, Version 3.0
Network Is Stabilized Once Again

FIGURE 4.10 Emergency or Project Certification. The emergency or project deployment leads to incremental version updates of the kernel.

Concluded

It all depends. If the system belongs to a consultant and is used only as a productivity tool, the issue has less importance. If consultants want to manage their own PCs without certification, that is their issue and not yours.

But if the computer is connected to the network, you have some degree of control. The least you can ask for is that the computer use a secure OS that conforms to your corporate security guidelines.

The certification problem is compounded when consultants use their own systems to perform development for your network. If this is the case, these consultants are in a situation where they introduce components into your network. As such, their systems should conform *completely* to your certification standards.

Presentation Layer Management

The third supporting process is the Presentation layer management process. Because improved user experience is one of the goals of the IT Model for Service Points of Access, it is important to identify the technologies and systems that define the Presentation layer.

Defining the Presentation Layer

Recall that the Presentation layer is the interface layer; for users, it most often means the desktop. For IT, it means the administrative interfaces for all PCs, servers, and services.

Desktop layer ownership focuses on three owners: users, IT, and the corporation. This means that the desktop must be divided into specific areas, or *zones of ownership.* But before these zones can be designed, several principles must be defined.

Principle 1: Default Corporate Desktop

The default corporate desktop should offer a single, unified interface to all users. This desktop should be the default desktop created at any new logon—when new users first log on to their PCs, or when current users use another PC, they should be presented with a desktop that meets every basic requirement and that provides access to all of the productivity components[12] of the SPA kernel.

Windows 2000/XP has the capability to store an editable default desktop and generate it for every new user. Corporations should update this default desktop with their own customized versions before deployment.

Reduced Training Costs

Providing a common desktop for all PCs uses the same principle that made Microsoft Office so popular. If the basic desktop interface is exactly the same for all users, retraining costs as users move from position to position within the corporation are greatly reduced.

Windows XP brings the power of desktop design to bear with a new interface that is completely customizable to organizational needs.

Principle 2: Personal versus Corporate Zones

Because the PC is a corporate asset, it belongs to the corporation. The corporation needs to keep a measure of control over it. But because users are people and people have a sense of propriety toward their tools, users also need to have control over some portions of the desktop. Negotiations for this shared ownership are performed through extensive communications programs outlining the reasons for this division of responsibilities.

Communications Program

Chapter 8 discusses the specifics of a communications program.

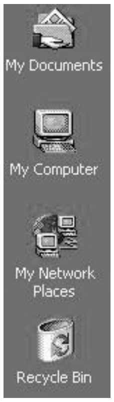
[12] For administrators, these should be all the administrative components of the kernel in addition to the productivity components.

Principle 3: Virtual Desktop

Using Windows 2000/XP's Active Desktop technology,[13] corporations can create and present *virtual* desktops to their users. In addition to conventional components on the desktop (icons, tool bars, menus), users can have direct access to intranet components based on who they are, which department they belong to, and which IT role they occupy.

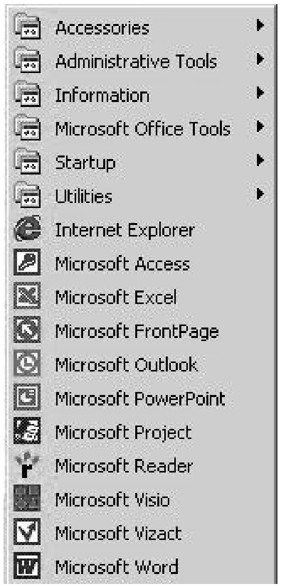
Every department in the corporation can design a standard desktop page that includes tools, news, information access, and more, and have it directly displayed on each user's desktop when he or she logs on to his or her system. The desktop is deemed virtual because the content of these pages can change at a moment's notice and be automatically updated on all targeted desktops at the next automatic refresh sequence.

Now that you have the basic principles in place, you can begin the desktop zone design process. This process should aim to cover the following requirements:



An example of the corporate tools zone.

- *Corporate tools zone:* The desktop should not be cluttered. Because Windows 2000/XP includes other, more practical areas for the location of shortcuts, only the most basic shortcuts should be on the desktop. These should include My Documents, My Computer, My Network Places, and Recycle Bin.
- *Personal tools zone:* Users should have the right to place additional shortcuts on the desktop because these are stored within their own profiles and do not affect others. But they should not be able to destroy any of the corporate items placed on the default desktop.
- *Common application access menus:* The Programs menu in the Start menu should be as clean as possible. Software tools should be grouped into types, and shortcut folders should be redirected during installation to these tool-type menus. All extraneous shortcuts, such as Uninstall or Register Now, should be removed. The menu should be clean because its role is to give quick access to common tools and stored access to utilities.
- *Quick Launch area:* Next to the Start menu, Windows 2000/XP offers the Quick Launch area. This area pro-



An example of a corporate Programs menu.

[13] Active Desktop displays Web content on the desktop. In addition, it enables users to display many more image types on the desktop, something all users appreciate.

Tool Groupings

Chapter 6 covers the methods used to identify tool ranking in your corporation.

vides one-click access to the tools for which it stores shortcuts. In addition, because this area is stored within the Windows taskbar, it is always available. This should be the preferred method of access for all the tools located on the system. For users, the Quick Launch area should include access to productivity tools. For IT staff, it should provide access to administrative tools.

As you can see in Figure 4.11, tools located in the Quick Launch area are grouped according to usage. First come the desktop, exploration, browsing, and research tools; then come office automation tools in order of usage; and finally come productivity tools that are required infrequently.

- *Wallpaper and screensavers:* Wallpaper is an area in which there is a lot of contention between corporate standards and personal taste. Dozens of companies have built their success on the production of wallpaper and screensavers for all tastes, good and bad. Corporations should provide some leeway for users here. Wallpaper and screensavers are something that users like to personalize, sometimes every day. But they are also something that the corporation needs to control in a certified network if PCs are to remain stable. A central bank of approved (for stability purposes) screensavers and wallpaper should be made available to all users.
- *Departmental Web zone:* This is an active zone. It can display the message of the day and provide access to mission-critical virtual[14] tools. This zone is controlled through an intranet Web page managed at the departmental level. Departmental zones vary on the desktop because they are displayed according to the department to which a user belongs.
- *Corporate Web zone:* The corporate Web zone includes global messages and access to corporate systems such

The Screensaver Craze

People have always enjoyed using screensavers and wallpaper on their desktop. The advent of the Internet has made a wide variety of screensavers available. The problem is that because the most sophisticated screensavers require additional software to run, some of the products downloaded from the Internet need to install components before they can run. These components are renowned for damaging systems.

In certified environments, corporations cannot allow users to install these faulty products, especially when they require the deactivation of critical components such as virus scanning to work properly. Unfortunately, there are many such screensavers on the market.

Corporations should provide banks of approved screensavers for users to install. This lets users have their fun and maintains system stability.

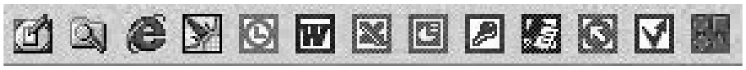


FIGURE 4.11 A productivity Quick Launch area

[14] Virtual tools are Internet-enabled tools. They are deemed virtual because they often do no require the installation of local components to work. If local component installation is required, the intranet Web server can install them automatically on the client.

as human resources, automated document templates, and global intranet research tools. This zone is displayed on all desktops. The Web content of this zone is controlled by IT for the corporation.

- *Project Web zone:* Project teams have special requirements. With new intranet-based project management technologies, such as those of Microsoft Project with Project Central, the inclusion of a project Web zone on the desktop provides support for personnel belonging to project teams. The project team is responsible for the management of this Web zone.
- *IT Web zone:* The last zone on the desktop is the IT zone. This zone can include several IT-related components such as help desk access, productivity support sections such as FAQs, tools and tips, training program booking, and emergency messages about the system. This Web zone is controlled by IT.

With Windows 2000,[15] your corporate desktop can become a very powerful tool if managed properly. The inclusion of Web zones on the desktop is a boon to productivity. With Web technologies, corporations can manage content in real time, and through central policy management technologies, they can choose to activate them at will. Figure 4.12 illustrates a sample desktop design.

Group Customization of the Presentation Layer

Every single network needs to categorize its users. Most do it for reasons of security. Managing access rights within the network is much simpler if all users requiring the same access are grouped within some type of container. These containers have in the past been limited to user groups, but Windows 2000/XP introduces the notion of organizational units as well. Whichever method is used—and in most cases, organizations will choose to use both—it is important that corporations take aspects other than security into consideration when designing user groupings today.

Microsoft has integrated this notion into Windows 2000 as *distribution groups*. These special groups are separate

[15] Windows 2000 is not the first technology to support Web content on the desktop, but it is the first to provide an integrated management approach to all desktop content.

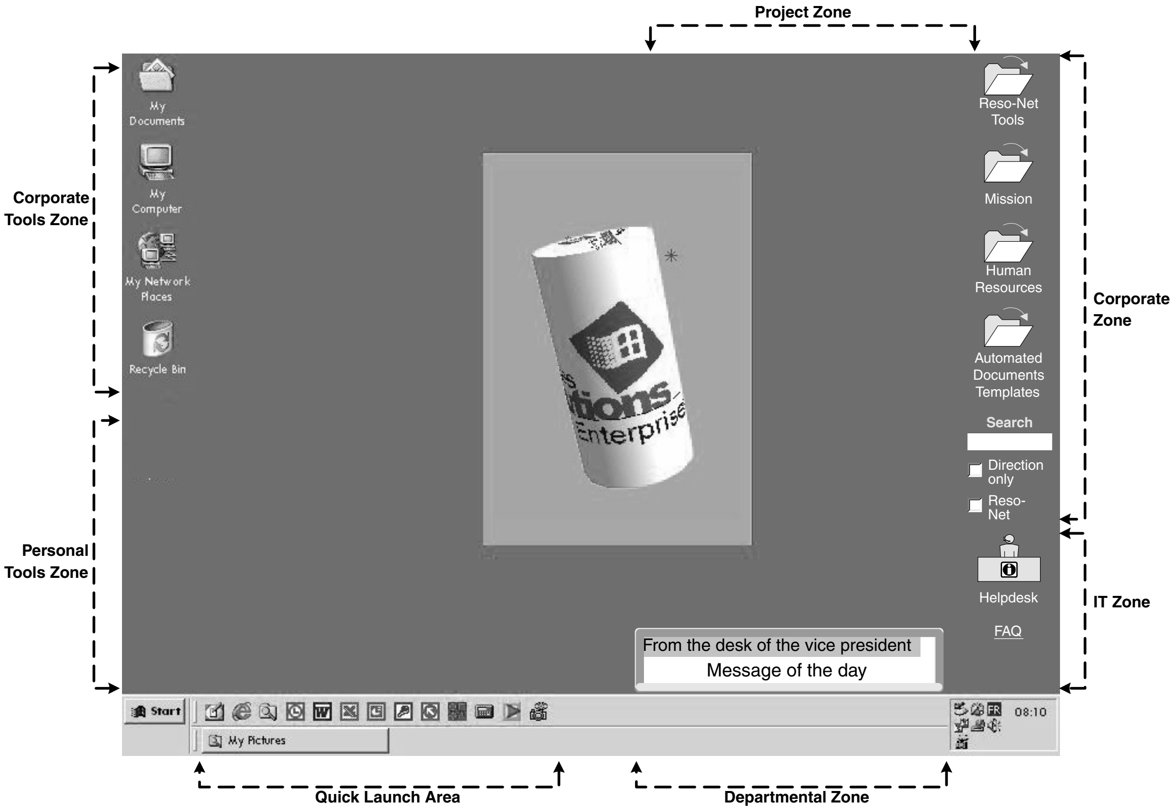


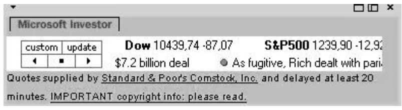
FIGURE 4.12 A Sample Corporate Desktop. The corporate desktop is divided into zones. Ownership of each zone is clearly defined. For users, the Active Desktop offers sophisticated wallpaper and screensaver images.

from security groups and are used for distribution purposes only. With these types of groupings, organizations can ensure that their central systems deliver specific content to specific groups of users.

With technologies such as Windows 2000, this means that corporations can deploy information to users—information that can take the shape of software products, software upgrades, documents, system updates, and Web content.

When designing the corporate desktop, organizations must take into account the granularity they want for specific content delivery. By determining how finely you want to detail your distribution points, you can determine how many levels are required in your distribution groups.

For example, several corporations have opted to divide their groups of users based on three major guidelines:



Dow 10439,74 -87,07 **S&P500** 1239,90 -12,9

$7.2 billion deal ● As fugitive, Rich dealt with pari

Quotes supplied by Standard & Poor's Comstock, Inc. and delayed at least 20 minutes. IMPORTANT copyright info: please read.

Microsoft offers free Web channels. This is an example of the Microsoft Finance channel. It is used to display stock information in real time on the Active Desktop.

- *Department minus 1:* Users are grouped by department and by division. Smaller groupings are not determined. This allows for distribution of line-of-business information and toolsets.
- *IT function:* Users are grouped by user role within the IT framework. This allows for the delivery of the software tools required to support the function.
- *Project groups:* Groups are created for project teams to support their specific requirements.

Remember, groups can be either organizational units or user groups. Use both judiciously. Organizational units are ideal for the department minus 1 guideline because groupings are vertical. User groups are the best for IT function and project groups because both cover the horizontal breadth of the enterprise—both cut across departments.

Managing Web Content on the Desktop

Web content must be managed, so it requires a supporting process. Because Web zone owners are responsible for their content, this process needs to be simple and straightforward. It must also be widely communicated to responsible groups.

An organization that has an intranet has usually provided some form of Web control to each of the responsible groups identified previously. Managing Web content is simply creating and maintaining a special type of Web page: the Web channel. Channels are stored as Channel Definition Format (CDF) files and are displayed within a special frame on the desktop.

On the creation side, a channel is just another Web page. Applying the channel to a desktop is done through a group policy in Windows 2000. This means that a policy enforcing the activation of the channel on given desktops is applied to the group container within Active Directory, Windows 2000/XP's central identity administration environment. Group policies are applied and refreshed on SPA objects (servers or PCs) on a regular basis that can be modified to suit your requirements. (The default is 90 minutes.)

Managing the Web content for the Presentation layer (see Figure 4.13) involves the following steps.

1. Prepare channel content.
2. Create or modify target groups.
3. Target specific groups with a group policy within Active Directory to activate the zone. The zone displays on the desktop.
4. Perform recurrent administration of the zone.

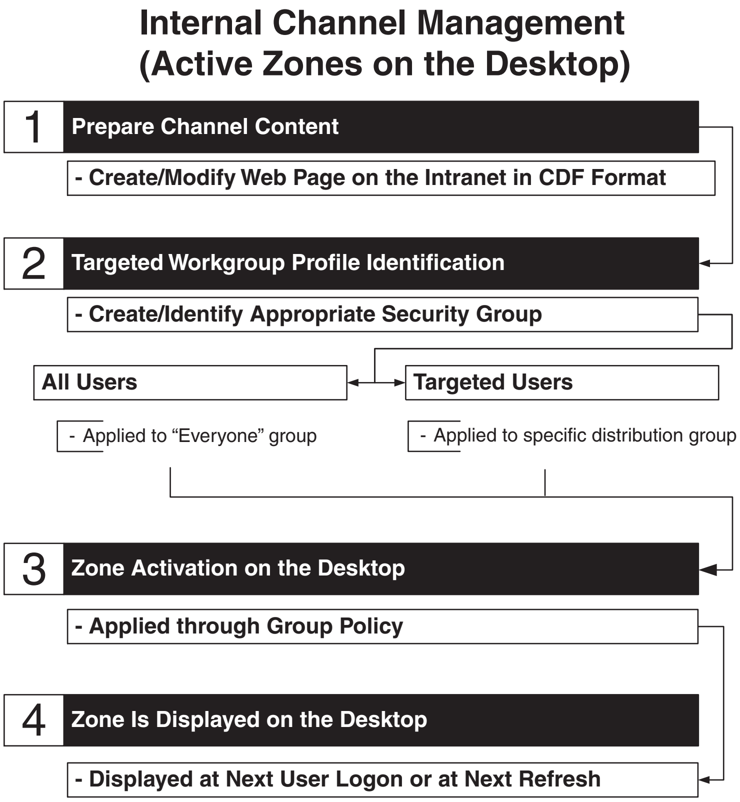

Internal Channel Management
(Active Zones on the Desktop)
1 Prepare Channel Content
- Create/Modify Web Page on the Intranet in CDF Format
2 Targeted Workgroup Profile Identification
- Create/Identify Appropriate Security Group
All Users
Targeted Users
- Applied to “Everyone” group
- Applied to specific distribution group
3 Zone Activation on the Desktop
- Applied through Group Policy
4 Zone Is Displayed on the Desktop
- Displayed at Next User Logon or at Next Refresh

FIGURE 4.13 Presentation Layer Management. Active Desktop zones are managed as Web pages and are applied to workgroups through group policies.

Using the QUOTE System

We have now covered all of the supporting processes required to put in place the EMF. You are now well on your way through the Organize Phase. You know how rationalization, certification, and Presentation layer management work. Before you finish this phase, you need to identify how management and administration of the EMF differs from traditional IT management.

In the next chapter, we will outline traditional IT administrative roles and describe the additional roles required to support the new processes introduced by the EMF.

Then you will be ready to move on to the Transfer Phase, the phase where you can put the new Framework in place.

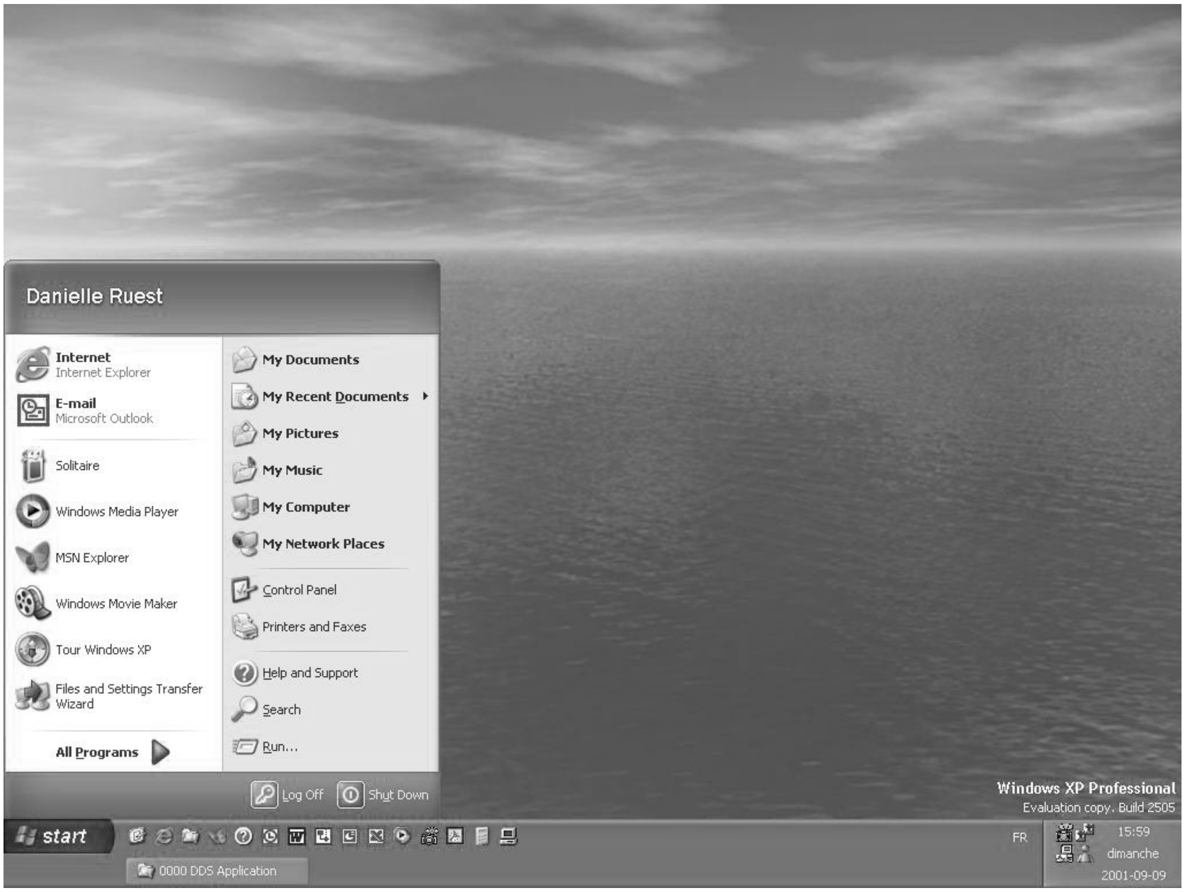
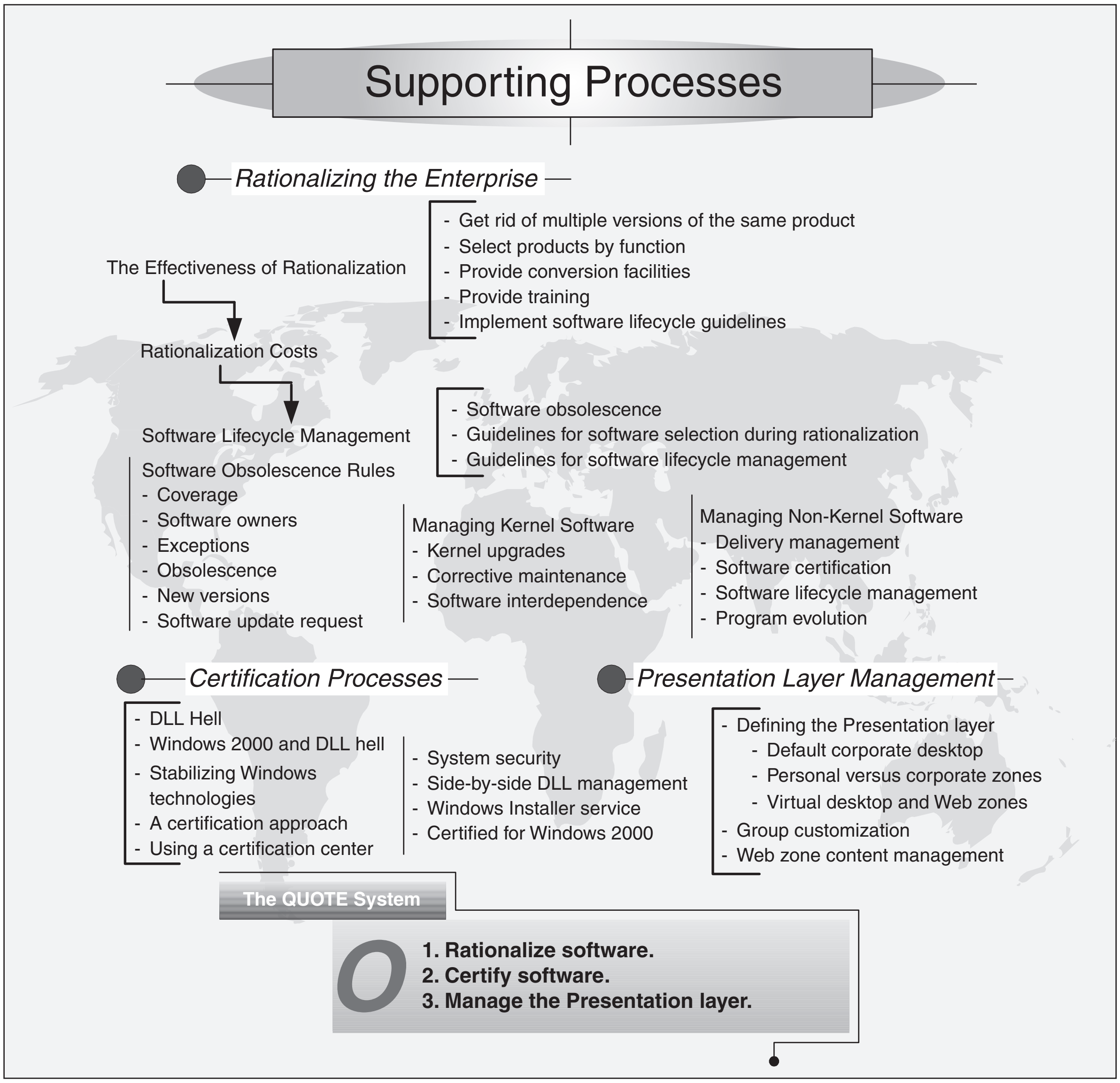


FIGURE 4.14 Windows XP's "Luna" interface supports richer content on the desktop.

CHAPTER ROADMAP

Use this flowchart as a guide to understand the concepts covered in this chapter.



Supporting Processes

Rationalizing the Enterprise

The Effectiveness of Rationalization

- Get rid of multiple versions of the same product
- Select products by function
- Provide conversion facilities
- Provide training
- Implement software lifecycle guidelines

Rationalization Costs

Software Lifecycle Management

- Software obsolescence
- Guidelines for software selection during rationalization
- Guidelines for software lifecycle management

Software Obsolescence Rules

- Coverage
- Software owners
- Exceptions
- Obsolescence
- New versions
- Software update request

Managing Kernel Software

- Kernel upgrades
- Corrective maintenance
- Software interdependence

Managing Non-Kernel Software

- Delivery management
- Software certification
- Software lifecycle management
- Program evolution

Certification Processes

- DLL Hell
- Windows 2000 and DLL hell
- Stabilizing Windows technologies
- A certification approach
- Using a certification center

- System security
- Side-by-side DLL management
- Windows Installer service
- Certified for Windows 2000

Presentation Layer Management

- Defining the Presentation layer
  - Default corporate desktop
  - Personal versus corporate zones
  - Virtual desktop and Web zones
- Group customization
- Web zone content management

The QUOTE System

O

1. Rationalize software.
2. Certify software.
3. Manage the Presentation layer.