

Realtime  
publishers

"Leading the Conversation"

*The Definitive Guide™ To*

# Vista Migration

*sponsored by*



altiris®

*Danielle Ruest  
and Nelson Ruest*

---

Chapter 2: Planning the Migration.....	27
Actual Migration Tasks.....	27
Hardware and Software Readiness Assessment .....	28
Personality Capture.....	32
Creating and Deploying Base Images.....	33
Application Packaging.....	35
Software Installation .....	36
Personality Restoration .....	36
Migration Status Reporting.....	37
An Extended View of Migrations.....	37
Using the QUOTE <sup>®</sup> System.....	38
Question Phase — Problem Statement .....	40
Understanding Phase — Exploring the Impacts of Migration.....	42
Organization Phase — Preparing for the Migration .....	46
Transfer Phase — Performing the Pilot Project, then moving on to Deployment.....	51
Evaluation Phase — Moving on to Production.....	54
Creating a Migration Project Plan.....	56
Project Planning Considerations .....	57
Required Teams and Roles .....	59

## Copyright Statement

© 2007 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

## Chapter 2: Planning the Migration

So you've decided to migrate to Windows Vista. Good for you! Your goal now is to design the very best migration possible and to ensure that the network remains stable during and after the migration has been performed. Successful migration projects rely on a proper structure. For this reason, this chapter will focus on how the migration project itself will be organized. To do this, we will begin by looking at the actual tasks required to perform a migration on a PC in order to help you understand what a tangible migration process involves.

But, because you know what to do to migrate one PC doesn't mean that the migration of dozens, hundreds or even thousands of PCs will just work. Migration projects are massive undertakings that require the coordination of hundreds of tasks—tasks that must be correctly orchestrated to be delivered on time, under budget and with a very high level of quality. To meet these requirements, you need a change management strategy, one that will help you blueprint each step of the process and ensure that everything works as planned.

Finally, you need to launch the migration project. Armed with the structured change management strategy, you can begin to lay out the steps you need to undertake to complete this project. That's why we close the chapter with a look at the actual project plan for the migration and the process you need to put it in place.

### Actual Migration Tasks

Migrating a PC and making that migration transparent to a user is hard work. Not that the work is actually hard, but rather, it is the coordination of multiple sequential tasks and the intricate linking of each task that is often difficult. You need to make sure that predecessor tasks are performed accurately, otherwise all of the tasks will fall out of sequence and the result will be less than successful. This is the old computer principle of GIGO or 'garbage in, garbage out' and it applies even more when the migration addresses more than one PC. You need to understand which tasks are required and how they fit together.

When you perform the migration, you need to coordinate seven specific tasks and link them together (see Figure 2.1):

1. Hardware and software readiness assessment
2. Personality capture
3. Creation and deployment of base images
4. Application packaging
5. Software installation
6. Personality restoration
7. Migration status reporting

Each task is performed in sequence and each has its own particularities.

 The PC Migration Cycle was originally based on Altiris' Six Step Migration Process. More on the Altiris process can be found [here](#).

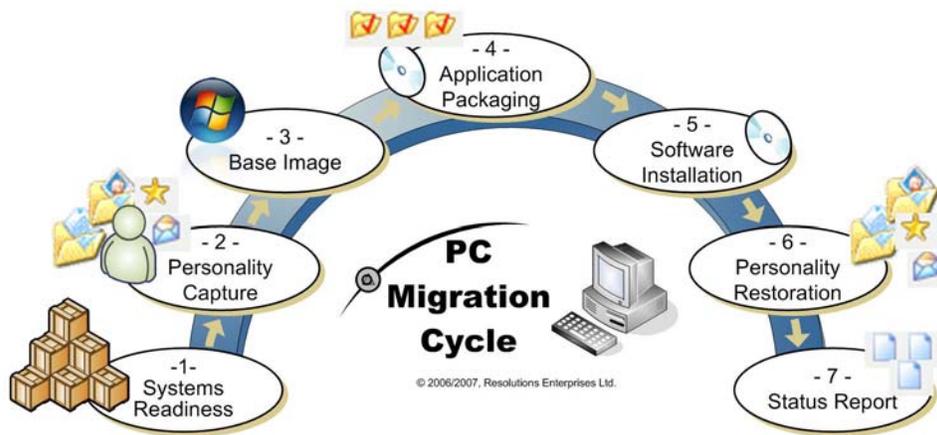


Figure 2.1: The PC migration cycle.

 This guide is about Vista migration. Many organizations will choose to include a new productivity suite such as Microsoft Office 2007 in the migration to a major new OS. Though this guide will not cover the migration to Office 2007, you can rely on the same processes and procedures to add it to your migration.

### Hardware and Software Readiness Assessment

Before you can begin to even consider installing a new operating system (OS) on any PC, you need to assess its current content and structure to first see if it will support the new operating system's requirements, and second, have any issues with hardware drivers, installed applications and expected system purpose. One of the best ways to perform this assessment is to rely on a checklist that leads you through the requirements analysis. If you're performing the migration for a single PC, you'll face a reduced list of questions, but if you're performing the migration for a number of PCs—any number, you'll face additional questions—questions that have to do with volume and address issues of location.

Of course, if you're working on a single PC, all you really need to do is to run a compatibility analyzer on it. As described in Chapter 1, the best way to do this for single PCs is to use the Vista Upgrade Advisor (VUA). This tool runs against your system and provides you with information on the latest requirements and compatible components. But running this on multiple PCs—any more than 10 actually—is less than practical. That's why you'll need to run a system check using either a deployed inventory tool, the Application Compatibility Toolkit version 5.0, or the Windows Vista Readiness Assessment (VRA), both from Microsoft. Note that both VUA and ACT require administrative credentials to install.

 More information about VUA, VRA and ACT can be found at  
<http://www.microsoft.com/windowsvista/getready/upgradeadvisor/default.aspx>;  
<http://www.microsoft.com/technet/windowsvista/deploy/readassess.aspx>;  
<http://www.microsoft.com/technet/prodtechnol/windows/appcompatibility/default.aspx>.

If you have an existing system management tool, you can use its own readiness assessment reports to determine whether your systems are up to snuff. For example, tools such as Altiris Inventory Solution or LANDesk Inventory Manager will include updated reports that allow you to assess both hardware and software readiness in one single pass.

 More information on Altiris Inventory Solution can be found at:  
<http://www.altiris.com/Products/InventorySolution.aspx>

 More information on LANDesk Inventory Manager can be found at:  
<http://www.landesk.com/Products/Inventory/>

Whichever tool you choose to rely on, you should concentrate on the following questions:

1. Do the PCs meet the system requirements for the new operating system? System requirements for Windows Vista were outlined in Chapter 1.
  - a. Which proportion of your systems meets these requirements?
  - b. Which proportion can meet them with minimal component changes?
  - c. Which components require changes—random access memory (RAM), video cards, displays, printers, others?
  - d. Would you rather change components or replace the system in order to aim for a standardized set of PCs in the network? Is this within budget?
  - e. How many systems do not meet the minimum requirements? Are you ready to replace them? Is this within budget?
2. Are there any driver issues on any system?
  - a. Which components are missing drivers?
  - b. Do the manufacturers of these components offer updated drivers?
  - c. If not, which do?
  - d. Are there compatible drivers you can rely on until the manufacturer updates theirs?
3. If hardware upgrades or replacements are required, should the upgrades be sized to meet any other project, present or future, requirements since change is required?
4. If PCs are to be replaced, are they in the same location or do you need to cover multiple locations?
5. Which operating systems are running on your systems? Are they homogeneous or heterogeneous—will you have to upgrade from many different operating systems?

 Note that Windows XP, Windows Media Center Edition and Windows XP Tablet Edition will all upgrade to Windows Vista as there is no longer a distinction between these operating systems as far as the Vista upgrade process is concerned. Of course, you will need the appropriate edition of Vista—Business, Enterprise or Ultimate—to perform the upgrade.

6. Are there any licensing costs associated with the migration? If yes, how will they be addressed? Remember that Vista uses a completely new licensing model (discussed further on) that requires you to put in place internal validation systems if you don't want all of your computers to be activated through the Microsoft Web site.

 More on this licensing model will also be covered in Chapter 5 as we discuss infrastructure changes to support the migration.

7. Which applications are deployed on your network?
  - a. Will they run on Windows Vista?
  - b. Better yet, are they actually in use? Do you have metering information to tell you if an installed application is actually being used by that PC's principal user?
  - c. If you are also considering moving to the x64 platform, will your current applications run on it?
  - d. Are there any applications that should be deployed to every single user? If so, should these applications be included in the PC image you will prepare?
8. What about user data?
  - a. Is it already located on servers and protected?
  - b. Is it located on each PC and will it need to be protected during the OS replacement?
  - c. Is there any data that can be filtered out, especially non-corporate data, to speed the replacement process?
9. Are any infrastructure changes required for the migration?
  - a. Is the project running on its own or will your organization include other changes—changes such as deploying a desktop deployment or management suite, moving to a new directory structure, including mergers and acquisitions—with the migration?
  - b. Will the migration only include the operating system or are some core applications—Microsoft Office 2007, for example—also included?
  - c. Is the network able to support in-place migrations? Are additional migration servers required?
  - d. If not, will you use a PC rotation process to bring each PC into a staging area and upgrade or replace it?
  - e. Which is the best schedule for the migration? If multiple locations are involved, will the change be location-based or based on lines of business?
10. Which non-IT processes will be required?
  - a. Do you need to start a communications plan for the migration?
  - b. Is training required along with the OS deployment?
  - c. Are any other administrative processes impacted by the migration?

The purpose of this questionnaire is to determine what is required to properly plan the migration. You might have a series of other questions to address your own particular situation, but this list is a good starting point. It should provide you with the financial, technical, administrative and organizational answers that will form the nucleus of the project.

Much of this information is gathered in inventories—inventories that need to be automated. Two inventories are required. The first occurs at the very beginning of the project and is deemed a cursory inventory. Here you're focused on numbers of PCs, numbers of users, PC types and configurations, locations of each PC and a general inventory of the software in your network. The purpose of this inventory is to give you a broad general idea of the scope of the project, allow you to generate a budget, and produce the business case that was discussed in Chapter 1.

The second inventory is much more detailed and is focused on the content of each individual PC. This content, detailing software, data, utilities and optional hardware components, will be used to actually replace the operating system on the PC and return it into the same state or rather, as similar a state as possible, once the migration is complete. In order to reduce the migration effort, this detailed inventory needs to be validated. Validation can occur through metering data if it is available. This will tell you if the software that is installed on the PC is actually used by the principal user of that PC. If you do not have access to metering data, then the inventory needs to be validated with the users themselves. This is a tough and time-consuming job.

Traditionally, organizations do not manage PCs very well. They tend to load programs on each PC based on the requirements of the principal user; but, as users evolve in the organization and move, add, changes (MAC) are performed on the PC, repurposing them for new users, software keeps being piled onto the PC. After a period of time, the PC begins to contain products that are not required by its current principal user. This can cause a lot of issues, especially in terms of licensing. Most software licenses are based on installation state: if a piece of software is installed, it requires a license, whether it is in use or not. Yet, so many organizations do not remove unused software when they repurpose the PC.

Beyond the licensing and non-compliance issues, having unused software on a PC becomes a real problem when it is time for an OS migration. That's because the project needs to take the time to validate the presence of each piece of software or utility on a PC with the principal user. This is never easy as users often have no idea what the software or utility actually does and cannot confirm whether they use it or not. We have been in countless migration projects where software inventories were validated with users and software removed from the migrated system only to find out that they actually did use the software but didn't know the name of the product or even that they were using it. This is usually because they address the software through other interfaces or through the performance of another task in their work.

It is extremely important to validate these detailed inventories by actually walking through a typical day's activities with the user instead of just producing a list of items and asking them if they use it or not.

 More on inventories and detailed validations will be covered in Chapters 5 and 9 as we discuss how to perform them and how to transfer the validated inventory to the deployment or release manager.

So, in the end you'll want to produce automated inventories as much as possible. Make sure you use a good inventory tool. Too many products produce only a list of all application components, registered dynamic link libraries (DLL), executables and so on, delivering an incomprehensible list of content for each PC. For this reason, make sure you look to the right requirements if you need to evaluate an inventory tool.

 For a review of asset management products, look up "[Cover your Assets](#)", an article examining four asset management products which can help with your readiness assessments.

## Personality Capture

When inventories are complete, you'll want to move on to the capture of the personality of each PC. PC personalities include the personal settings of each user that logs onto the computer, including their profile, favorites, data, application settings, desktops, mapped drives and so on. You also want to include any custom application properties such as custom dictionaries, templates, email data files and so forth.

This information is vital to the success of the migration project. For a user, there is nothing worse than having to recreate this personality once the OS migration has been performed. This step is related to the quality of service or service level that the project will provide. An OS migration is an IT task—a task that should be as transparent as possible to the end user. After all, it is IT that is responsible for all computing processes, not users. For users, a migration project or the delivery of a new tool should be business as usual and the disruption of business processes should be as minimized as possible. For this reason, it is so important to properly design computer systems, clearly identifying what belongs to users, what belongs to the corporation and what IT is responsible for. It is the portion that belongs to the user that makes up the personality capture.

 This division of responsibility will be covered in Chapter 7 as we discuss building locked-down computer images.

In addition, personality captures must be evaluated for each user profile you find on the computer. Most organizations assign a PC to a principal user—at least it is easier to manage PCs when they are assigned to one single principal user. So, during the personality capture, you need to evaluate if existing profiles are in use or not, rationalizing unused profiles so that you do not restore 'garbage' to the new PC.

Some organizations however do not have the luxury of assigning a principal user to PCs. These include organizations that have 24/7 service offerings or organizations that are in the manufacturing industry and have shifts that work around the clock. In these cases, several users access the same PC. Organizations often use generic accounts for this so that only one profile needs to be captured. This may change with Vista since it supports Fast User Switching, even when the PC is joined to a domain. Generic accounts are a security risk because you can never guarantee you know who is logged on with the account. Organizations should look to the removal of generic accounts from their production networks.

Several tools are available for personality captures. And of course, the best tools will also identify the validity of a profile and whether it should be moved or not. The best policy is to capture all profiles, store them on the network, archive them after the project and restore only those profiles that are actually in use. This way, you have access to old profiles that you may have unwittingly marked as obsolete when they were not. This can happen when users are on prolonged leaves of absence or on vacation during the project timeframe. This policy costs the project in storage space and network bandwidth, but provides the very best quality of service. We have seen projects where vital personalities have been lost—personalities of users with several years' worth of data stored locally—because this service level was not put in place. Best to be safe than to be sorry.

 Many tools will be discussed in Chapter 8 as we cover the capture and replacement of personalities on computer systems.

Of course, the very best policy is to rely on network features such as folder redirection to map user profiles and data to networked drives where they can be protected at all times. This will be one of the considerations for infrastructure modifications in preparation of the migration.

 For more information on personality captures for Windows Vista, look up [Migrating to Windows Vista through the User State Migration Tool](#). Captures can also be performed with a number of third-party tools such as [Altiris PC Transplant Solution](#), [LANDesk Management Suite](#) or [CA Unicenter Desktop DNA](#).

### **Creating and Deploying Base Images**

The very best value for organizations relying on PC-based technologies is to reduce diversity, or reducing the numbers of hardware configurations, buying PC systems in lots with identical configurations, reducing the number of operating systems in use and rationalizing everything from applications to utilities to external devices. Too many organizations, especially organizations that are brought together through acquisitions and mergers, have several different antivirus systems, several different deployment systems, or several different monitoring technologies. The principle of standardization can never be overused when it comes to IT and systems management. Many organizations find themselves very happy to deal with heterogeneous systems, but usually, these organizations are driven by IT, not by business requirements.

Today's organizations need to drive this point home with their IT groups: business is the ultimate driver for the use of technology, nothing else. That's why you should push for a single worldwide global image for your computer systems. With Vista, this is not a dream, but a reality. Vista no longer has issues with different hardware abstraction layers (HAL)—the component that is particular to each PC configuration and is absolutely required to run Windows. Vista can now apply different HALs at load time from a single system image, removing the need to create and maintain one image per HAL. Of course, today, it is possible, through scripts and other mechanisms, to create a single disk-based system image and deploy it to multiple computer configurations, but few organizations take the time to make this effort. With Vista, this effort will be greatly reduced.

The PC system image needs to be designed from the ground up. It should contain everything that is generic to a PC and should ensure that the PC is secure and protected at first boot. You need to determine if you will create an image with the OS only or whether it should contain all generic applications as well. There is a strong argument for both positions and in the end, the most determining factors will be how you stage systems and which tool you use to do it. For example, if your staging tool supports multi-casting—sending one single stream of data over the network to multiple endpoints and thereby greatly reducing network traffic when deploying multiple computers—you'll probably look to the 'fat' image or the image that contains as much as possible since deploying one or fifty PCs at once will take the same time. If not, you'll want to stick to 'thin' images as each deployment will take considerably more time.

 Chapter 7 will provide a detailed discussion on thick versus thin images.

Windows Vista introduces new concepts and new ways to work with system images based on three installation scenarios: upgrade, refresh and replace (see Figure 2.2). In the upgrade scenario, you can now use Vista to actually perform a non-destructive upgrade of the system, replacing the complete contents of the operating system without affecting user data or application settings. This is the first time that the upgrade scenario actually works, but it remains to be seen if organizations are ready to move to this installation format.

In the refresh scenario, you capture personality data, wipe or erase the computer's disk drive and reload a brand new operating system, reload applications and restore personality data. This is the scenario most organizations are familiar with. There is a lot to be said for it as it reformats the disk drive and creates a new, pristine environment. But, with Vista, reformatting the hard disk drive may no longer be necessary. Because it is installed as an image by default, Vista has the ability to non-destructively replace an operating system either 32- or 64-bit without destroying existing files. Then, because it includes an automated disk defragmenter, it will automatically begin to defragment the hard disk drive using low priority I/O once the installation is complete. It may be worth examining this scenario as it offers the ability to use local disk drives for personality captures and restores and would save time by cutting down network traffic.

In the replace scenario, you capture data from an old computer and then perform a bare metal installation on a new computer—a computer with no preexisting OS—reloading applications and data once the installation is complete. This is also a favorite among organizations and IT professionals. Whichever scenario you decide to use, make sure your original system image is properly designed and ready for prime time before you begin testing with it. Finally, you'll need an image management strategy, allowing you to update the image as new patches and hotfixes are provided for the components it contains.

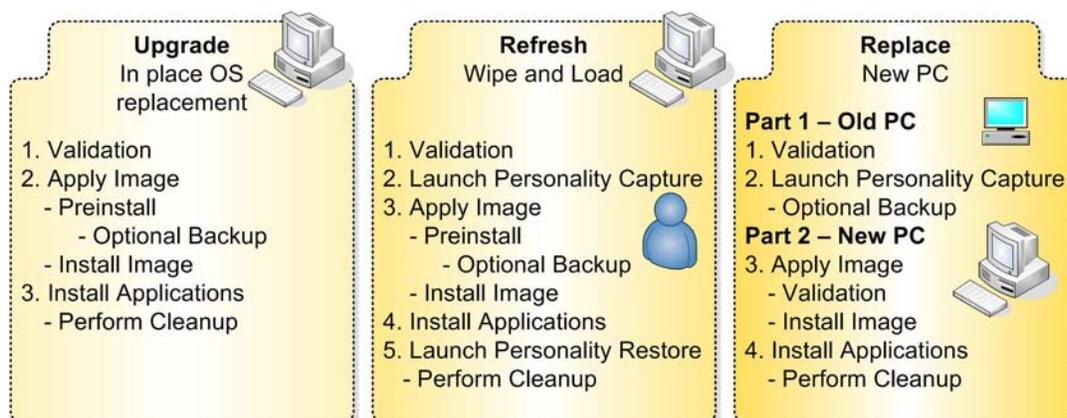


Figure 2.2: Windows Vista supports three migration scenarios.

## Application Packaging

Inventories will be very useful not only to determine how but also which applications will be deployed to end systems once the OS is loaded. Ideally, all applications will be rationalized, removing duplicates from the network and of course, removing any application that is made obsolete by the features of the new OS you are deploying, leaving you with a reduced list of applications to deploy. Deployment methods will be discussed in the next step, but before you can move on to deployment, you need to package all applications for automation.

Software packaging is one of the most important aspects of any deployment or migration project, yet it is often overlooked or addressed in cursory manners. Application management is a lifecycle which involves several steps, one of which is the management of all automated installations of applications. After all, if users are focused on the business, you don't want them to lose productivity by trying to run software installations on their systems. Besides in a locked-down environment, they just won't be able to. So you need to automate installations, and these automations need to be standardized. You just can't afford to deploy software packages that run several different installation methods. If you decide to install applications locally on each system, then you'll want to package for the Windows Installer Service (WIS) as it provides a single installation strategy for all locally installed software in Windows Vista.

You can also choose this migration project as a vehicle for change in the way applications are managed on desktops. If you do, then you should consider application virtualization—a process that installs all applications within a sandbox on the PC and protects the operating system at all times from any changes required by the application. Software virtualization offers many advantages, the least of which is the ability to run conflicting applications on the same PC. Since all applications are sandboxed both from the OS and from each other, two versions of the same application can run on the same system at the same time. More on this will be covered as we discuss application preparation, but for now, the important aspect of this topic is that software needs to be packaged through standard processes—processes that need to be transferred to normal production operations once the project is complete.

 For more information on software packaging in general, visit the [Articles](#) page on the Resolutions Web site and look for the **Application Lifecycle Management** section.

For more information on software virtualization, look up our white paper on [Software Virtualization](#).

### **Software Installation**

Software installation refers to the deployment of any software that is required by a user to complete their function within the organization. Once again, PC design is very important here. If you choose to rely on thin images, you will need to deploy more software to your users. If you choose to rely on thick images, fewer applications will require deployment. Whichever model you decide to use, you'll need to properly identify the targets for software deployment. Ideally, these targets will group users into role-based categories, letting you deploy appropriate software based on the role the user plays within the organization. This way, finance users will receive all of the financial applications; manufacturing users will receive manufacturing applications and so on. Grouping applications doesn't only help in deployment, it also helps in software management and package preparation as you only have to make sure that the applications in a given group will work together and don't need to verify the application against any others. You inevitably have to deploy applications on an ad hoc basis as well, but this should be to a greatly reduced group of users.

One caveat: if you want to minimize application management tasks, you'll have to make sure that your deployment tool supports the automated removal of applications, not only when the application is upgraded from one version to another, but also when the application falls out of the scope of deployment. For example, if you want to avoid finding applications that don't belong to a user role on a system and you want to make sure you have a tight control over your licensing costs, you'll want to make sure you implement a software management system that automates the process of changing a PC role, removing obsolete applications and applying appropriate applications automatically.

### **Personality Restoration**

Once the applications have been restored to a system along with the new OS, you're ready for the return of the system's personality. Ideally, this process will be automated along with the preceding steps. Remember that profiles also need to be rationalized and only valid profiles should be restored. You'll probably want to change the way the personality is designed on user's systems. For example, Vista completely changes the way user information is stored on the PC, no longer relying on the Documents and Settings folder, but now focusing more on a new Users folder. In addition, because of the built-in search engine in Vista, users should no longer rely on desktops cluttered with unused icons, but should rely on more efficient access methods for applications and documents.

If you do change the design or the way anything in the personality functions, make sure that you communicate all changes to users. There is nothing worse for a user to find something missing without any explanation.

### **Migration Status Reporting**

The last step in the actual migration process is status reporting. If you're working on a single system, you'll want to make sure that it is stable, up-to-date in terms of security patches and running properly. You'll want to verify that all applications have been installed and that the user's personality has been restored as it should be.

If, on the other hand, you're working with a multitude of systems, migrating several locations in parallel and managing all of the deployments from one central location, you'll want to make sure that each and every success is properly reported back to central administration for the project. You also want to track any issues that may arise, any migration failures and any unsatisfied users. Ideally, you will have prepared a special help desk team for the support of the migration and you may even have gone through the trouble to provide local coaching resources for migrated users. You'll want to track all of these issues if you want to control project costs.

In addition, you want to be able to have a single view of the progress of the project at all times. How many PCs are deployed? How many are left? How many are being deployed right now? Are we on target? Do we need to modify our schedules? Deployment management requires intricate planning and lots of administrative hard work as users move and shift in the organization during the deployment. Having the right deployment reporting tool is one of the most important aspects of project management for migrations. Make sure you get the right information and have access to it when you need it.

You'll also want to rely on this information for the project post-mortem. This will help you generate return on investment (ROI) reports and demonstrate the value of the massive effort your organization just undertook.

### **An Extended View of Migrations**

The migration of an actual PC is a seven-step process, but managing change on this scale in organizations of any size is much more than just running through the seven steps described previously. The project requires proper structure and organization. Few sources exist for this type of information—information that guides you through the change management process implied in OS migrations. Because of this, Microsoft has produced the Solution Accelerator for Business Desktop Deployment 2007 (BDD).

 The Microsoft BDD can be found [here](#).

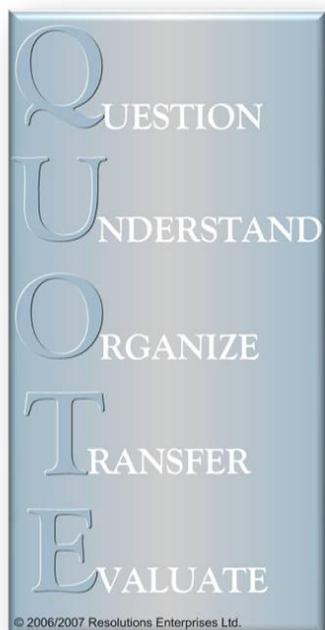
The documentation contained in Microsoft's BDD attempts to put each and every step of the deployment program into perspective. It relies on the Microsoft Solutions Framework (MSF) to do so, but unfortunately, MSF is a framework that is designed for software development, not systems integration. It makes sense that Microsoft would rely on MSF since it is an organization whose focus is nothing else but software development.

Also, the BDD guidance is divided into multiple feature teams. In our experience, there are really two major feature teams: one who is in charge of preparing all aspects of the desktop and its construction, and one who is in charge of preparing all aspects pertaining to server components for the deployment. Within these two teams, you'll find applications specialists, security experts, management gurus, and other technical staff with all sorts of expertise; in fact, the broader the experience of your teams, the better. Managing two teams makes it easier to control the outcome of the project and keeps everyone in better contact.

But, nevertheless, BDD is a very valuable source of information on deployments and there is no doubt that the tools it includes are a great help for organizations of all sizes. We'd like to supplement its guidance with our own deployment management strategy—a strategy which is focused on the integration, not development, tasks that make an OS migration project work. This system has been in use for over eight years and has been successfully adopted by a multitude of firms. It helps identify the critical path for any migration project.

### Using the QUOTE<sup>®</sup> System

The QUOTE<sup>®</sup> System is a structured approach for change management which is based on five phases (see Figure 2.3):



**Figure 2.3:** The QUOTE<sup>®</sup> system.

- Phase 1, Question: This is the diagnostics or discovery phase. It is focused on a situation review, followed by a needs analysis. You need to identify the factors of the change situation. First is the fact that change is coming. Second is the scope of the change. Third is where and how the change originated. Fourth is what you already have in place and how it will be affected. Question phases always involve an inventory of some sort.
- Phase 2, Understand: This is the planning phase and focuses on initial solution development. Now that you have all of the information on the change in hand, you can start planning how to influence the change. If the change is technological in nature—for example, a desktop deployment—it is essential to fully understand how the new technology feature set will impact and improve the elements and processes already in place within the organization. This stage may include one or several proofs of concept (POC)—working mock-ups of the solution to come.

These POCs will help flesh out the details of the solution you will deploy because they provide additional, and often non-technical, evaluations of the way the solution is structured. This phase also finalizes the scope of the project and its structure.

- **Phase 3, Organize:** This begins the execution portion of the project. It is focused on preparing the deployment mechanisms and the deployed content; in fact, all of the engineering tasks of the project. It includes everything from detailed conception of the solution to complete validation, testing all of the facets of the solution to ensure quality of service for the next phase. Since changes such as OS deployments tend to affect the organization as a whole, it is vital to validate all the elements of the solution before they are implemented.
- **Phase 4, Transfer:** This is the second part of the execution portion of the project and focuses on manufacturing processes or massive operations because of the repetitive nature of deployments. All systems are “Go!” and a production environment is used to deploy the solution on a massive scale. Begin with the pilot project to validate that everything works as planned, and if not, make final modifications. Then, when all solutions have been fully tested and adapted, you begin to transfer them to the sections of the organization that are impacted by the change.
- **Phase 5, Evaluate:** Finally, you need to evaluate the change you implemented in order to best view how it can continue to evolve within your organization. In addition, the project’s performance as a whole is evaluated against its original objectives. Rules are derived for future projects and adjustments are made to future approaches. This is also the stage where the solution passes into ongoing operations. You need to transfer project mechanisms into production, finalize all support training and transfers of knowledge and report on your successes. Because change works in cycles, you’ll also need to re-initiate the discovery phase to ensure the continued evolution of the solutions you deployed. This re-initiation is crucial if you want to fully benefit from the change you implemented. Many projects fail because this re-initiation is not performed and the project is closed off once it is complete.

This approach can be used in almost any change situation, personal or organizational. Using this approach within organizations ensures that organizational and especially, IT change is always structured. It greatly simplifies and reduces the impact of change in organizations of all sizes. In Chapter 1, we argued that operating system deployments are part and parcel of PC lifecycles. In IT, change is constant as manufacturers constantly improve on existing products and new products abound. You should rely on the QUOTE System to manage all IT change.

#### **Forklift versus Hardware Refresh Migration Strategies**

As discussed in Chapter 1, there are several different strategies for migrating to Vista. Most organizations will use one of two main strategies: forklift—migrating all systems at once; or hardware refresh—migrating systems as they are replaced through purchasing programs. Both methods are valid, but in order to make your migration a success, you’ll need to properly prepare the design and engineering aspects of the project. This means that no matter which migration method you use, you’ll still need to run through the Question, Understand, and Organize phases of the QUOTE. Where the strategies differ is in the Transfer and Evaluate phases. With forklift migrations, the Transfer phase is a massive effort affecting all PCs in the organization. With hardware refresh migrations, the Transfer phase actually carries on for several years until each and every PC is replaced.

Whichever one you use, don’t make the mistake of not being thoroughly prepared before delivering the new OS to your users.

Because of its nature, the QUOTE System is ideal for the preparation and review of deployment projects. Here’s how it works.

## Question Phase — Problem Statement

As mentioned earlier, the Question Phase relies on inventorying the organization (see Figure 2.4). You need to collect and review several different inventories and refine them if the information you need isn't available. If you're already running a modern infrastructure, you'll always be ready for change because you'll already have this information in hand and if you don't, you'll already have deployed the tools that can provide this information to you. If you don't have an up-to-date infrastructure yet, then you'll have to determine how you're going to generate these inventories. And, make sure you take advantage of the project to put in place an automated inventory tracking process, one that will be part of your default infrastructure after your deployment is complete.

### The Question Phase

This phase includes:

- Situation Review including:
  - Internal Inventories/Problem Identification
  - Hardware Readiness Assessment
  - Software Readiness Assessment
  - Infrastructure Readiness Assessment
  - Organizational Readiness Assessment
- Needs Analysis
  - Initial Solution Design
  - Selection of New Technologies
  - Review of New Technology Features
- Initial Project Evaluation Including:
  - Potential Benefits
  - Initial Budget Estimates (~25% error)
  - Initial Project Scope
- Cost/Benefit Analysis
- Production and Presentation of the Business Case
- Go/No Go for the Project

In deployment projects, the Question Phase is usually initiated by a series of potential situations:

- Software Product Owners—if you have them—have identified that new versions of the technologies they are responsible for are available and meet the organization's criteria for project initiation.
- Users have identified that new technologies are available, are using them at home and they want to take advantage of them at work.
- Business objectives or business processes have changed and new technologies are required to support them.
- The organization is ready for change and requires the introduction of new technologies to promote it.
- A major manufacturer or software publisher has released a new version of their product.
- A major manufacturer or software publisher has announced the end of life for a specific product you are currently using and will no longer provide support for that product.

In the case of a Vista migration, the project initiation can originate from a number of these factors, especially end-of-life or user demand. It is good practice for you to initiate the project *before* these factors come into play. When you do so, you'll want to begin with a situation review and then follow with a needs analysis to identify the benefits that could be acquired through the integration of Vista's host of new features. The situation review needs to focus on readiness assessments—hardware, software, infrastructure and organizational—that will indicate whether your assets are compliant with Vista's requirements.

Then, the needs analysis will focus on a list of potential benefits including existing problems that could be solved by the technology. Remember to include the Help Desk database in your assessments; it is a very valuable source of information for this type of project. The needs analysis should include a scope definition—who will be affected, what the impact of the change will be and how you wish to approach it—and initial budget estimates with plus or minus 25 percent error margin. This is enough to generate the relative size of the budget you will require.

You should also generate an initial solution design, outlining how the new technology will improve and address issues you are facing today and how these benefits can turn into real value for the organization. Keep an eye on the potential return on investment your organization will reap when it moves to the proposed solution. This will help you create first, a risk analysis, then, a cost/benefit analysis to ensure that the organization will profit from the change.

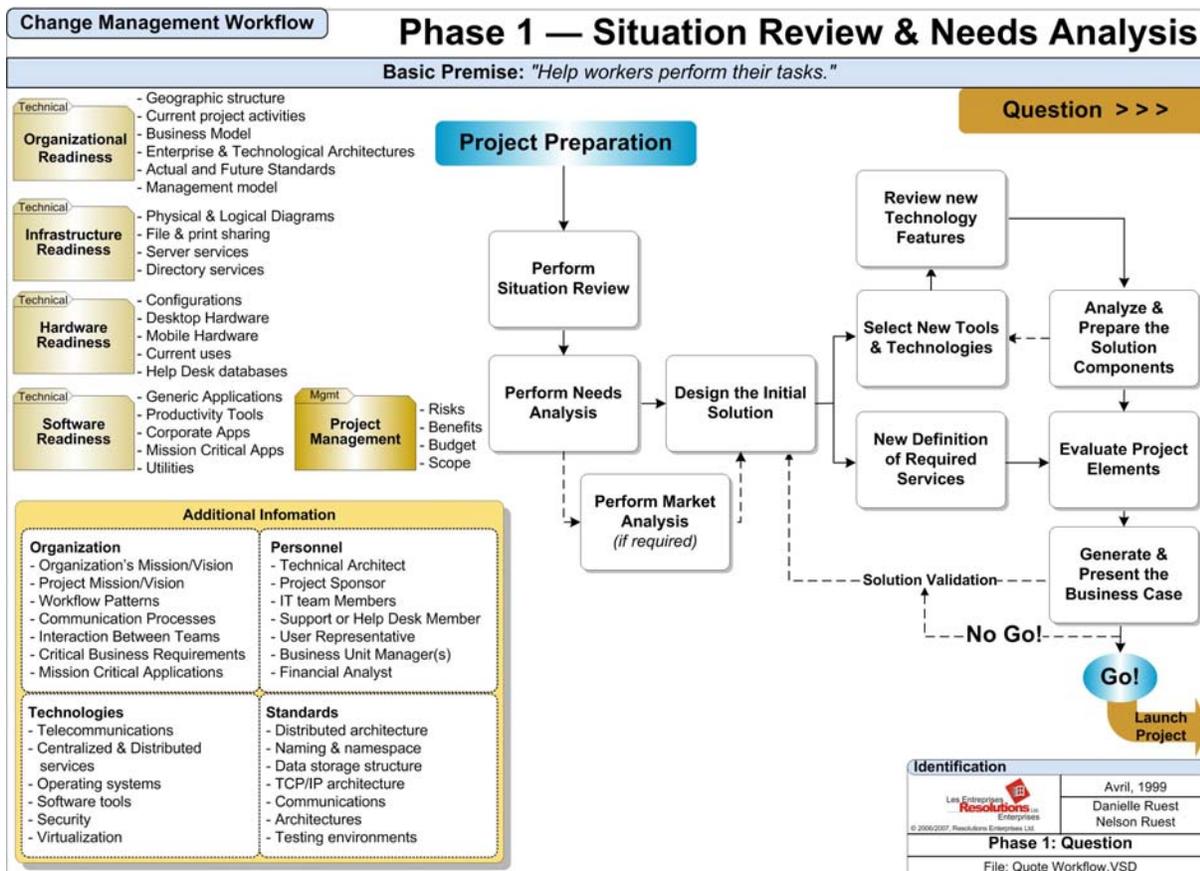


Figure 2.4: The Question Phase focuses on inventories and initial solution design.

### Where to Spend your Dollars

Research has shown that those who spend little time engineering their solutions before moving on to deployment, have much more painful experiences than those who do. Remember the 80-20 Rule, spend lots of time getting all of the engineering right and your project will run more smoothly.

### The Value of Automation

Make sure you aim for a fully-automated deployment. In our experience, costs for manual deployments average between \$500 and \$1,000 per PC. Semi-automated or Lite Touch deployments, to use a term from the BDD, average about \$350 per PC. Microsoft's goal for fully automated or Zero Touch deployments is less than \$50 per PC. That's why you should aim for a fully-automated deployment strategy. That's also why this is the strategy recommended by this guide.

All of the previous elements will become part of one major deliverable: the business case. Prepare your business case with care and have it validated by the other members of your initial team. Then, when you feel that you have everything in place, present the business case to the stakeholders. This phase culminates in a Go/No Go decision for the project.

 Remember to go to the DGVM's [companion Web site](#) to obtain the tools provided in Chapter 1: the **Business Case Template** and the **List of New Features for Vista**. These tools will greatly help your own project's Question Phase.

## ***Understanding Phase — Exploring the Impacts of Migration***

The Understand Phase is used to begin mastering the new technologies to be implemented and identify solutions to known problems. The first step is to list the features of Windows Vista which you have decided to implement. Then, using the results of the Needs Analysis created in the previous phase, you can map features to current issues. This will serve to refine the initial solution design that was created in the previous phase.

Next is project definition. Once again, you refine the initial project plan and scope prepared in the previous phase. While the objective of the previous phase was to get the business case approved, the objective of this phase is to refine all aspects of the project so that it can move forward on to the other three phases.

Refining the project definition means preparing and generating all of the elements of the project: Vision, Objectives, Scope, Structure, Teams, Tools and Timelines; all this in order to create the definitive project plan. This time, the project definition should include a refined budget estimate to within plus or minus an error margin of 10 percent. This becomes the project's official budget.

This budget should be supplemented by two additional documents: an evaluation of the potential impacts of the project on all aspects of your business and the migration strategy. The impact evaluation should cover items such as work patterns and operational processes, user interaction with IT Systems, system changes, operational changes, structural changes, and so on. The summary migration strategy that was presented in the business case is refined to identify the strategies that will be used for systems deployment, interactions with users during the deployment, training strategies, remote and local deployments and so on.

### The Understand Phase

This phase includes:

- Secondary Review of the features of the new technology including:
  - Identification of features to be implemented
  - Final mapping of features to known issues
- Project Definition including:
  - Vision/Objectives
  - Scope
  - Structure
  - Teams
  - Tools
  - Budgets & Impact Analysis
  - Timelines
  - Project Plan
  - Project Organization Manual
- Initial Communications Program
- Logical Solution Definition including:
  - Naming and other standards that apply to the solution
  - Identification of Problem Resolution Strategies
  - Logical Solution Architecture
  - Initial Training Solution
  - Detailed Deployment Strategy
- Initial Acquisitions
- Initial Laboratory Creation including:
  - Operating and Testing Standards
  - Preparation and Testing Environments
  - Initial Tests
  - Solution Mock-up
  - Technical Training
- Proof of Concept
- Go/No Go for the Project

Also, now that the project is officially in place, the project's communications program needs to be designed and initiated. Its scope depends on the scope of the change the organization will bear. In the case of Vista deployments, you should make sure you invest in communications early. It's never too early to promote comprehensive change projects like an OS migration. Your objective is to promote good will towards the project and its deliverables. There are several ways to do this. *Don't make the mistake of underestimating the importance of the communications plan in this project!* Lack of proper communications programs can make even the best engineered solutions fail because users will resist the change. One good example of the kind of help the communications program can provide to the project is through the creation of a project 'welcome kit'—a tool that outlines all of the major points of the project, its structure, its objectives and the role each person will play in it. With projects of this scale, this kit will quickly become a vital tool as personnel rotate through the teams.

 For a detailed discussion on the requirements of a communications program for your Vista project, see **Communications Plans: a Sample Chapter from Preparing for .NET Enterprise Technologies** at <http://www.reso-net.com/documents/ruestch08.pdf>.

You also need to launch the initial training program and training analysis. Little training is required at this time and is often limited to basic technical training for the core project team, but this is the time to begin examining all of the training options that will be required by the project—end user, technical, support teams, and so on. As with communications, training programs are critical to the success of a Vista project. Actual user training does not need to be overwhelming; it needs to be direct and to the point. Often simple presentations outlining new ways to do things and offered to end users are all you need. Technical training needs to be more comprehensive, but can once again, be very creative, running in-house or Web-based programs to cover just what you need and nothing else. The training program needs to be tightly integrated to the solution you design. Make sure you budget for this very important task.

At the same time, you can begin to refine the initial solution you outlined in the business case. This will include the definition of the standards that will guide the architecture of solution. It will rely on these standards to identify problem resolution strategies and eventually culminate into the Solution Architecture. This architecture serves as input into both the communications and training programs and will provide direction for the engineering work that lies ahead.

Because you need to refine the solution, you'll need to begin the creation and preparation of the laboratory you'll require to support the preparation of the technical elements or engineering work of the project. You may only need to adapt existing laboratories if they are already in place; if not, you'll have to create it from scratch. In either case, you may need to perform some initial acquisitions in order to populate this laboratory. And, since this laboratory will be used for the duration of the project, you will need to define its operational and testing standards.

 Chapter 3 will outline how to go about preparing the laboratory for the Vista migration. It will focus on the use of virtualization technologies as these technologies are ideal for the support of laboratory efforts.

It is a good idea at this stage to create a working mock-up of the solution. This mock-up can greatly assist in budget definitions through cursory testing of existing applications and systems. For example, with a migration to Windows Vista, you should use the lab to identify what the acceptable baseline is for existing hardware. Should this baseline be different than that outlined in the business case, you'll have to refine the budgets for hardware acquisition. The mock-up can take the form of a single virtual machine image that can be used to demonstrate new feature sets to the project stakeholders.

The laboratory serves three major roles: solution preparation, testing, and personnel training. The latter role is as important to the success of the project as the first two. Technical staff requiring retraining should have scheduled access to the lab so that they can begin playing with the technologies. Often the best way to give them this access is to, once again, rely on virtual machines running with the new OS so that they can begin to familiarize themselves with the coming delivery. It is also vital to schedule this training officially, especially with the core project team, otherwise no training will ever take place and the technical training program may have to be outsourced. Keep in mind that one of the main project goals is to ensure that a transfer of knowledge is performed to all technical staff. You want to keep all new skills in-house at the end of this project.

The laboratory is called the certification laboratory because it will be used to certify all engineering aspects of the solution—PC images, application packages, support structures, deployment procedures, personality captures and restores and so on. And, because several different teams will be working in this laboratory, it is a great idea to prepare a laboratory welcome kit that outlines how the lab works and where team members can find the resources they require to perform their tasks.

Finally, it is often a very good idea to perform a proof of concept to clearly demonstrate the benefits brought by the migration to the new OS in order to get refined budgets approved. Once again, the organization must issue a Go/No Go decision for the project since new budget authorizations are requested (see Figure 2.5).

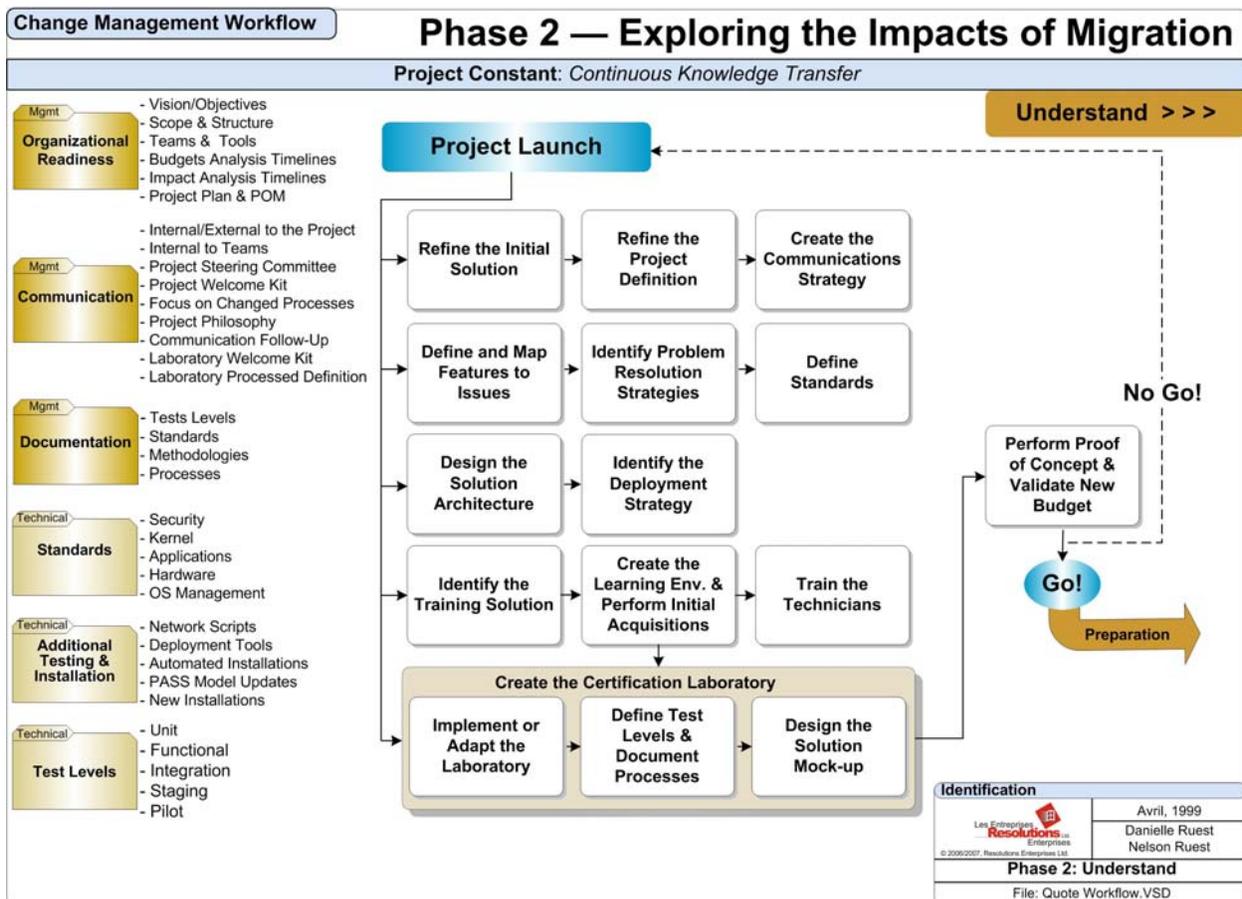


Figure 2.5: The Understand phase focuses on solution refinement and project preparation.

### The Scope of the Proof of Concept

The Proof of Concept (POC) can take a number of different forms. It could be just a mock up demonstration to the Project Steering Committee—the management group that oversees the project. Or, it could take the form of an actual deployment to key team members. In some cases, the POC can eventually cover about one percent of your user population.

You'll probably have several POCs—some focusing on individual features, others on portions of the overall solution, and eventually, a one percent population sample. It is a good idea to treat POCs the same way you treat any engineering project, with Alpha, Beta, and Gold deliveries.

## Organization Phase — Preparing for the Migration

The third phase of the project focuses on finalization of all technical aspects of the deployment and eventually, preparation to move to the pilot project. The stage involves the bulk of the engineering work required to perform the deployment. Everything is refined and tested and then tested over again until it all falls into place as one integrated process. This is the phase that addresses the actual migration tasks described earlier. In fact, the purpose of this stage is to engineer and automate each aspect of these tasks.

### The Organization Phase

This stage includes:

- Finalization of the Solution Architecture
- Solutions Development (if required)
- Solution Integration Programs including:
  - Deployment Infrastructure Preparation
  - Solution Security Elements
  - Application Compatibility Testing and Packaging/Certification
  - Automated Installation Mechanisms
  - User Data or Personality Recovery Mechanisms
  - Solution Certification
- Solution Support Programs including:
  - Refined User and Hardware Inventories
  - Validation of all Inventories with Users
  - Technical Training Program
  - End User Training Program Preparation
  - Special Support Mechanism Preparation
  - Data/Application Conversion Programs
- Continued Communications
- Deployment Strategy Finalization and Approval
- Final Acquisition Launch
- Pass/Fail Acceptance Testing

This phase also addresses the finalization of the architecture for the solution. This architecture may need updates as engineering work proceeds and identifies actual processes and procedures instead of expected logical outcomes.

Solutions focusing on infrastructure changes, such as the Vista deployment, require a lot of integration and very little development, but developers should be part of the team to offer support for any automation request through scripts and other measures. They are also called upon to develop custom components for the solution such as Web tools or documentation libraries. Another reason to include developers at this stage is that they are one of the final clienteles for the deployment and their requirements are considerably different than those of normal users. That's because they need to have some form of access to elevated privileges in order to perform their work. Unlike users who will run with standard privileges, developers need to install and test code as it is being produced. The solution can either be focused on giving them virtual machines or granting them elevated rights in development domains. Whichever you choose, try to avoid giving them access to elevated rights accounts in your production domain because this can lead to both security and administration issues down the road.

This phase is all about the integration programs. These include system image staging mechanisms, application compatibility testing and repackaging, the definition of data recovery mechanisms or personality captures and restores, the definition of all of the security elements of the solution, and the preparation of deployment infrastructures such as mobile staging rooms and training facilities. A lot of work needs to be performed by the various teams that are engaged in this technical aspect of the project. After all, for the deployment to work, these mechanisms must be fully flushed out. Each aspect of these programs is fully tested and certified before it is ready for testing. The bulk of this eBook addresses each of these programs in detail and helps you understand how to go about preparing them.

The preparation of the support programs is also one of the tasks addressed with this phase. Several support programs are required. First, you need to validate all of the inventories of each PC. This validation is essential because you don't want to be deploying unnecessary products to end users. This is an arduous administrative task unless you have software usage or metering data. Quality control is very important at this stage because these validated inventories will become the production request sheets that will be handed off to the release manager for the migration of each and every PC in your network. If mistakes are made here, mistakes will be made during the deployment.

This is also the time where the bulk of the technical training program will be performed. Begin with the personnel on the project team and then move on to other, non-project personnel. Now is the time to train them because the deployment is not yet engaged and resources only need to deal with the day-to-day issues they are already used to. Don't wait until the deployment actually starts because then, your resources will more strained as unforeseen issues may arise and project teams are engaged in deployment support activities.

In addition, now is the time to begin the preparation of the end user training program. As the solution coalesces into a final deliverable, you can begin to see which elements will need to be covered by such training. Be creative. One great idea would be to include custom training videos users could view during the deployment and then access later if they need a refresher course.

Another aspect of the project that needs to be covered here is the preparation of special support mechanisms: Will the regular support team absorb support of the deployment, or should a special, separate team be put in place? How long will this special team be required? When does a migrated user move back to the regular support team? What is the project guarantee time period? The answers to all of these questions form the support mechanisms for the project. In most cases, deployment support is offered by the project for a period of 20 working days before the user is returned to recurring support mechanisms. This 20-day period forms the project's deployment support guarantee.

Finally, you need to put in place data conversion programs, especially if you decided to include Microsoft Office into the deployment of your client operating system. This might mean deploying conversion tools to existing PCs as you deploy the new OS and then, performing a final data conversion after the deployment is complete. You may also need to perform application conversions. For example, Microsoft Access databases and custom tools often need to be upgraded when new versions of the suite are deployed. These conversion programs are as important as any other aspect of the project because they deal with your most valued asset: information.

 If you find you need to perform application conversions such as those for Microsoft Access, we highly recommend you redesign these application programs. For more information on gaining control of Microsoft Access applications, see [Decentralized Deployment Strategies](#).

This is also the time to initiate all of the acquisitions for the project; getting the right licensing agreements for Windows Vista and for any other software you might choose to deploy, for example, Microsoft Office Professional 2007. It also means obtaining any hardware upgrade components or new computer systems to be able to supply the deployment rate in the next phase.

### Microsoft Windows Vista Licensing Changes

Microsoft has put in a lot of effort into Vista to protect itself from theft or software pirating. Many of these features are also designed to protect your organization and to ensure that your licensing keys are not being used outside your organization. These issues address the most common problem with Windows XP volume licenses: theft or misuse of the volume licensing key. After all, you don't want your licensing key to be used on computers that no longer belong to you or being used by users who no longer work in your organization. The new licensing changes address just this type of situation.

Organizations have three options for the acquisition of Windows Vista. The first and simplest is to acquire retail licenses of the product either with new PCs or through upgrades. In this case, you'll probably want to acquire Windows Vista Ultimate Edition as it is the only one that includes all of the features that make Vista truly great, but is also the most expensive edition.

Organizations that acquire Vista through volume licensing or software assurance programs have two more options. The first is to deploy a centralized key management service (KMS) which controls the activation of Windows operating systems without requiring individual PCs to connect to a Microsoft Web site. At the time of this writing, KMS can run on either Vista or Windows Server Codenamed "Longhorn" (WSCL). It may be adapted to Windows Server 2003 in 2007.

To run KMS, organizations need to have at least 25 PCs running Vista or five WSCL servers consistently connected to an organization's network to operate—virtual instances of operating systems do not count. KMS can support the activation of hundreds of thousands of PCs from one single KMS device, though organizations should have at least two KMS devices in the network, one main device and a backup system. Location of KMS devices can be performed through auto-discovery relying on the DNS service or through direct connections, entering the machine name and port number for the connection.

Client computers must renew activation by connecting to the KMS device at least once every 180 days. New, un-activated clients will try to contact the KMS every 2 hours (configurable), and once activated will attempt to renew their activation every seven days (configurable) to renew their 180-day lifespan. This 180-day lifespan ensures that any system that leaves your premises with a copy of your license will eventually expire.

Organizations requiring multiple activations, but with fewer than 25 systems or with special situations that does not allow them to connect to a KMS, can rely on Multiple Activation Keys (MAK). MAKs are special activation keys that will support individual PC activation with no time limits or can go through a MAK Proxy to activate several keys at once.

If the copy of Windows Vista becomes deactivated for some reason, the following features will no longer work:

- The Windows Aero user interface will no longer operate.
- Windows Defender will no longer remove non-critical threats.
- Windows ReadyBoost will no longer operate.
- The Windows Update Web site will no longer provide downloads.
- Windows will provide persistent notifications that this copy is unlicensed.

Un-activated or de-activated PCs have a 30-day grace period before requiring re-activation. Copies of Windows that go beyond the grace period enter Reduced Functionality Mode (RFM). In addition to the reduced functionalities listed above, a PC in RFM mode will:

- Run a default Web browser when the user opens a session.
- Run sessions with no Start Menu, no desktop icons, and a black desktop background.
- Log users out after an hour without warning.

Make sure you take these aspects of licensing into account when engineering your deployment.

 Note that Microsoft will be releasing a version of the KMS for Windows Server 2003 in the first half of 2007. You can then run KMS on a server as it should be instead of relying on a Vista PC.

As each of the engineered elements of the solution is released from the solutions integration programs, it needs to undergo certification—arduous testing programs that validate that the new component will work under any condition. If it fails certification, the component must return to the integration center and be modified or updated to meet certification. If it passes certification, then the product passes on to acceptance testing—testing that is performed by appropriate clienteles, such as user representatives, local developers, line of business representatives and so on. These users have a pass or fail authority on each component and will determine if it is ready for prime time. Once all components pass acceptance testing, they are ready for the final test: the pilot project (see Figure 2.6).

 Since change is a constant in IT, it is often better to lease than to buy. Leasing programs include automatic upgrades on a scheduled basis. In addition, they provide consistent costs over the years.

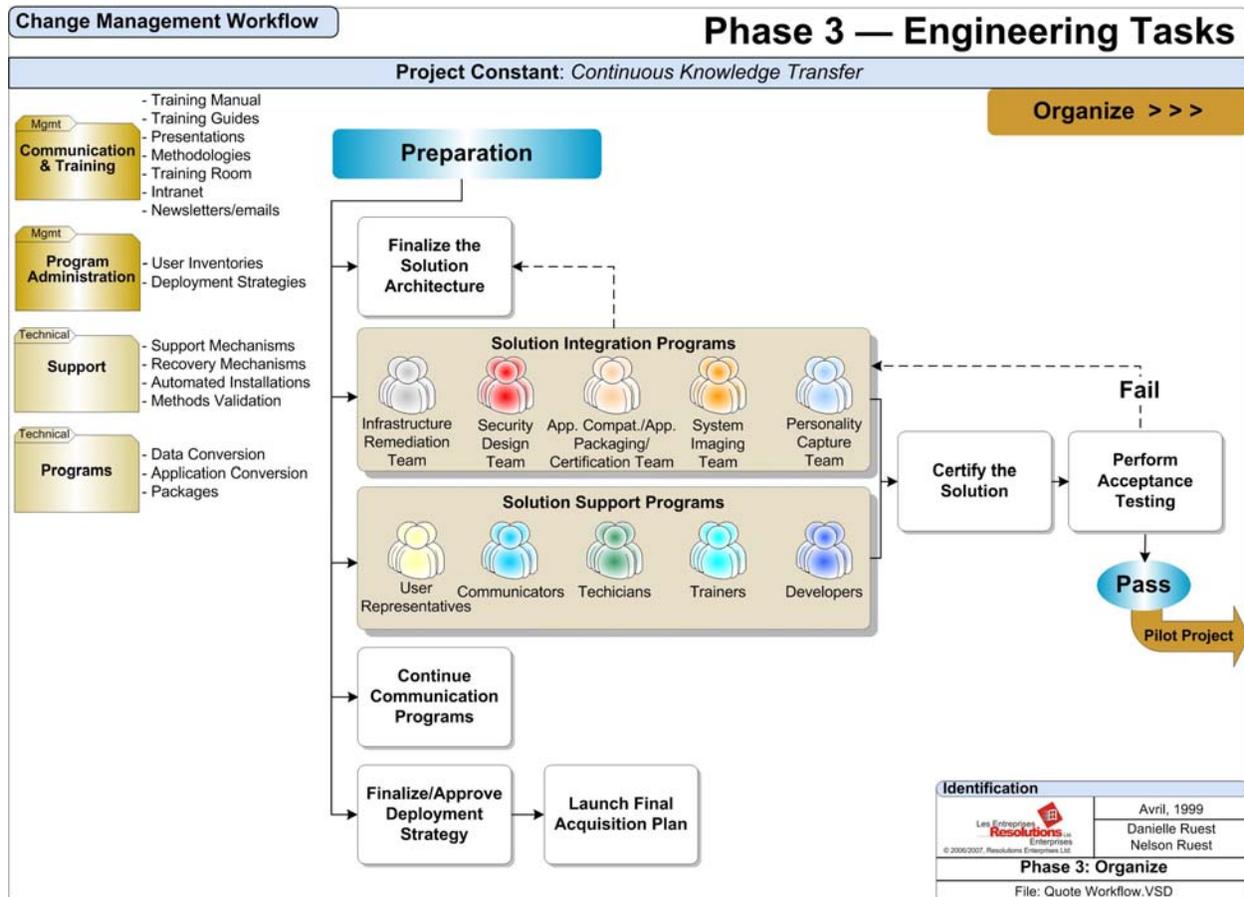


Figure 2.6: The Organize phase concentrates on finalizing the technology transfer methodology.

### Enterprise Project Planning

It is important to identify existing and ongoing projects in preparation of the next phase because yours may conflict with other project schedules. If synchronization is possible in delivery schedules, you should do everything to make it happen. After all, your driving principle is *little or no disruption to business operations*.

### **Transfer Phase — Performing the Pilot Project, then moving on to Deployment**

Once all engineering aspects of the project are complete, you need to take the time to complete the administrative and support aspects of the program before you test actual deployments. This includes the finalization of the end user training program, the communications program, the support program and the deployment schedule. Each of these needs to wait until all engineering is complete before it can be finalized. Don't try to rush things. Much of this effort can be performed in parallel with the engineering tasks, but not all of it. You need to reserve a period of time after engineering finishes, but before the pilot to finalize the content of these programs. After all, it is hard to finish the training kit or the user guide until you are 100 percent sure of what the image will look like once it is deployed.

#### **The Transfer Phase**

This stage includes:

- Finalization of Deployment Procedures Including:
  - Training Program
  - Communications Program
  - Support Program
  - Deployment Schedule
- Pilot Project including:
  - User Communications
  - Solution Deployment
  - Training Programs for Users
  - Training Programs for Technical Staff
  - Solution Support
  - Pilot Evaluation
  - Approach Modifications (if required)
- Deployment Launch including:
  - Deployment Coordination
  - Deployment Risk Management
  - User Communications
  - Massive Solution Deployment
  - User Training Programs
  - Residual Technical Training Programs
  - Deployment Support
  - Hardware Disposal Program
  - Ongoing Program Evaluation

Once all components, administrative and technical, are ready, you can move on to the final test: the deployment of the pilot project before you move on to massive deployment (see Figure 2.7). Pilot projects should address up to 10 percent of the user population and should include at least one remote site if your organization includes remote sites. The purpose of the pilot project is to test all of the facets of the deployment process to ensure quality of service for the actual deployment. The pilot project should represent every deployment activity from user communications warning them of the upcoming deployment to PC and software deployment as well as training. It should deliberately test every deployment mechanism. For example, if the project aims for the replacement of PCs, a select target group of users should be chosen to ‘forget’ applications on purpose during the PC staging process. This allows the testing of the support mechanisms for post-deployment rapid application distribution. You need to choose a select group for this type of test in order to avoid skewing the deployment results which will be used to help determine if you are ready to move on to massive deployment.

#### Using a System Stack

This guide outlines how to use a system construction model to build the PC image. This allows you to structure how applications will be deployed during the project. Generic applications or applications that are targeted to 100 percent of the user population are prepared first, which means that you can begin to deploy to users who only require generic applications as soon as they are ready. This often means as many as 50 to 60 percent of your user population. Then, as you are deploying to these users, your software packaging teams can complete the packaging of applications that are outside of the system stack. These applications should be grouped by role and when a role is complete, then you can include it into your deployment schedule, giving you time to complete the next role and so on. This lets you manage deployment activities in parallel and shorten the actual deployment schedule.

Your pilot should aim to meet the goals of the massive deployment. For example, if your goal is to deploy 50 PCs per day during the massive deployment, then you should aim to meet that same objective during the pilot. Also, in order to capture comments from pilot users, you should include regular meetings with user representatives to determine the ongoing status of the pilot.

#### Marry Training with Migration

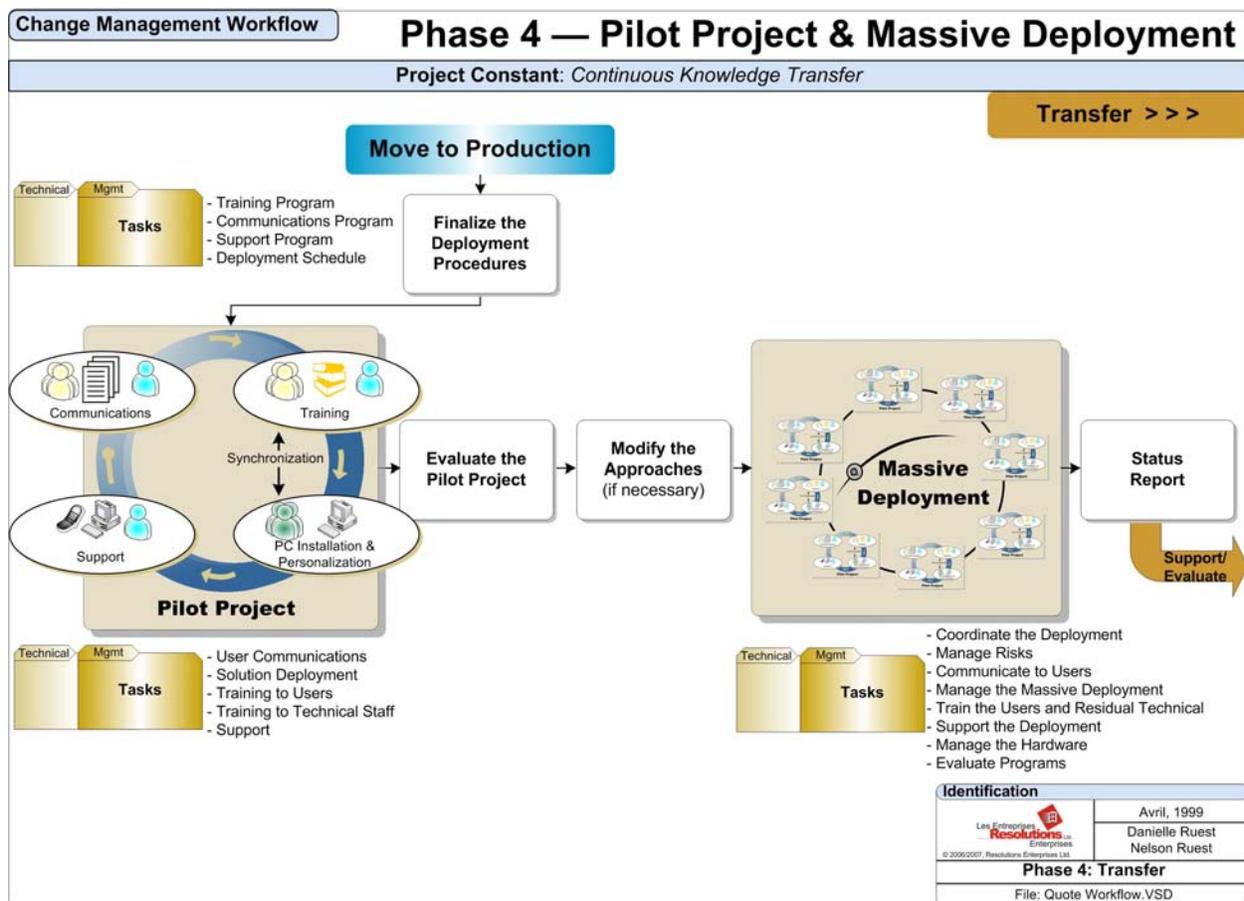
No matter how good your engineering is, migrating each PC will take time. Depending on your approach, it could take between one to three hours. It is always good to estimate half a day for each group of migrated PCs.

Of course, you can't expect users to hang around for that time doing nothing as their PC is out of commission. If you did, you couldn't consider your approach as having no impact on business as usual. One excellent approach is to marry the user training program along with the PC migration. Call users into training, either through Webcasts or in larger presentation rooms, and introduce them to the new desktop and tool set while their PC is being upgraded or replaced. This way, users will find themselves moving to the new OS from the moment they walk into the training room. When they return to their desk after the training program, they find their PC migrated and ready to go.

Give users some time to get used to the new system. Vista is not a major change, but it is different from any previous version of Windows and if you want them to be as productive as possible, let your users have some time to get used to the new system at their own pace.

Since the purpose of the pilot is to validate and refine every aspect of the process, you'll need to perform an evaluation at its end, sort of a pilot post-mortem. You should collect comments from every pilot participant: project managers, coordinators, administrators, technicians, users, trainers, support personnel—centralized and local, communicators, and any other personnel that participated in the pilot. Comments and recommendations should be used to modify deployment approaches in order to improve the quality of deployment services. You don't want to identify issues with the approach when you're ready to perform massive deployment operations, so get it right during the pilot.

If you find that some re-engineering is required, pilot project users should be first on the deployment list for the next phase. Because they are the first to receive the new Vista operating system, pilot users accept to bear any inconveniences the project may bring if its approaches and solutions are not completely fine-tuned. That's why it is important to "reward" them with the first reception of the refined solution.



**Figure 2.7:** The Transfer phase concentrates on repetitive deployment tasks.

Final modifications should be made to the deployment strategy before you start. Make sure you get management to buy off on this strategy before you can proceed. So, now you're ready for deployment at last. All of your processes and solutions have been tested and approved and you're ready to proceed with the deployment itself.

Actual deployments tend to be quite complex to orchestrate—users aren't available for training sessions, parts don't come in on time, technical staff fall sick—sometimes it seems as if everything is going wrong, especially with massive. Two factors will help: strong deployment coordination and deployment risk management. Since the deployment is in fact the coordination of a sequence of activities, the personnel you select to administer it require very strong coordination skills. You can also rely on the communications program which should warn users one month, three weeks, two weeks, one week and then one day before they are migrated.

As for risk management, it means identifying potential project weaknesses and preparing for them. Before launching the deployment, make sure that you've covered all the bases. Your pilot project should have helped identify the majority of these risks. Remember: always have a backup plan. For example, you should have a select list of replacement users ready and on hand in case some of the targeted users are not available. This way, you can continue to meet your daily deployment objectives and return to missed users later.

When you proceed with the deployment, you'll quickly see that after a while it becomes quite repetitive. All the better, practice makes perfect! You'll soon be repeating all of the activities in the pilot project on an ongoing basis. Don't get lulled by the monotony of the process! Always perform ongoing project status evaluations and be ready to modify approaches if something goes wrong. Also provide status reports to management so they can see you are continuing to meet objectives and staying right on target.

### ***Evaluation Phase — Moving on to Production***

#### **The Evaluate Phase**

This phase includes:

- Project Post-mortem Evaluation
- Transfer of Project Processes into Recurring Administration including:
  - Final Transfer of Knowledge to Operations Staff
  - Deactivation of Project Support Mechanisms
  - Final Documentation Delivery
- Beginning of the Solution Evolution

The final phase includes three major activities. The first is to perform a project post-mortem to evaluate the project's performance against its objectives. Did it complete within budget? Did it meet every objective? Did it fit within defined timelines? Where there any overruns? Answers to these questions help form the rules and guidelines for future projects. After all, you know there is another OS deployment in your future.

The second is the transfer of the deployment solution to recurring operations. If you didn't have a managed desktop before the deployment, then now is the time to make sure you maintain all of the deployment processes in your ongoing operations, otherwise, you'll find yourself in the same situation next time you need to run a project of this type. To do this, you need to finalize all project documentation and have project personnel perform a final transfer of knowledge to administrative and operations staff. All project support mechanisms will be deactivated.

 In a modern architecture, it is important to deactivate and not dismantle project support structures since you know they are required on a recurring basis.

And third is the evolution of the solution. Once operational staff has accepted delivery of the new technology and signed off on it, secondary evolutionary projects can begin. For example, after deploying and implementing Windows Vista, you can now begin to learn how to master it and leverage its capabilities to increase quality of service within your IT infrastructure. This is the right starting point for this evolution because many of Windows Vista's features are only available to a network that has completely migrated to this operating system. These modifications can often be performed in the background with no further disruption of organizational or operational activities. This way, the Evaluate Phase becomes the starting point of a new Questioning phase and process as your organization moves on and evolves with the new capabilities you just implemented (see Figure 2.8).

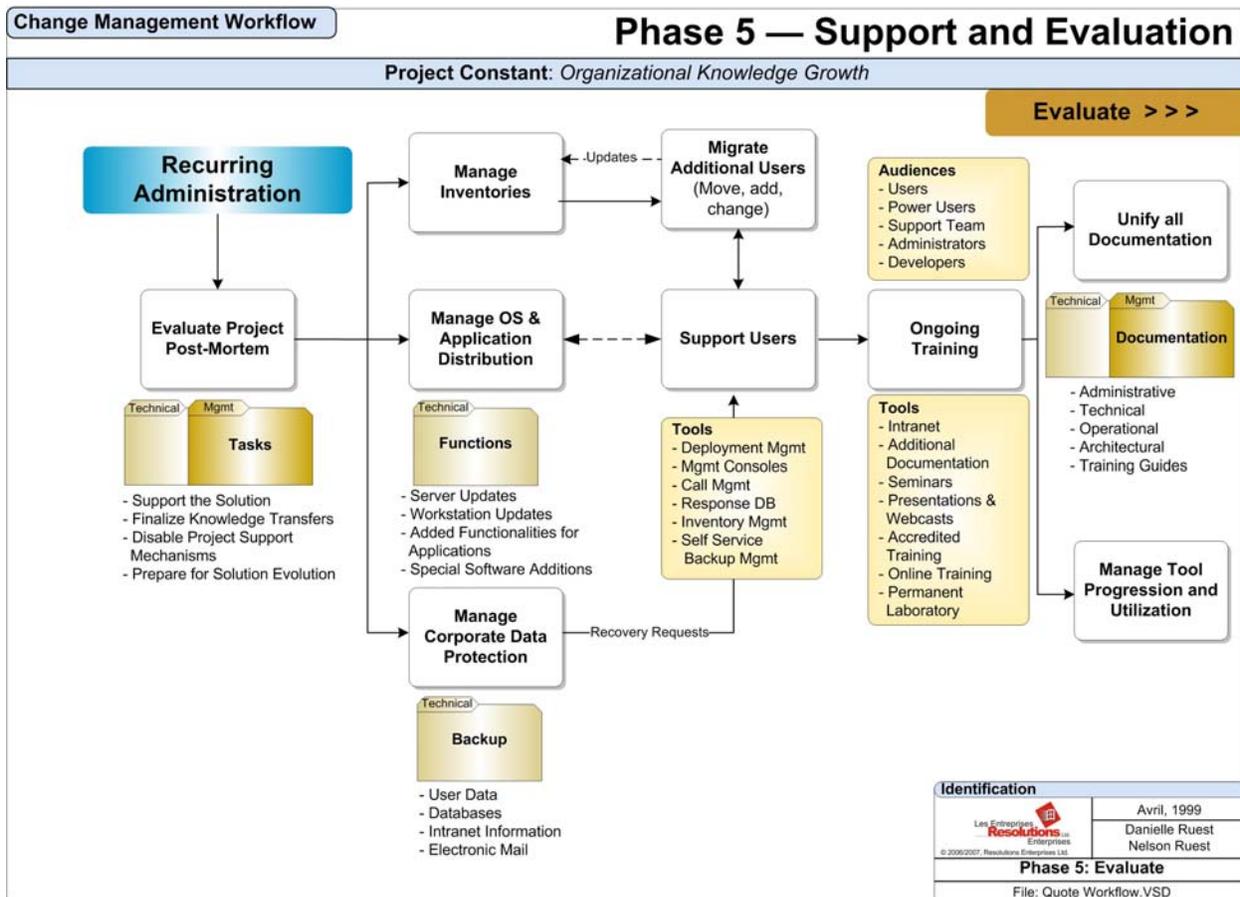


Figure 2.8: The Evaluate phase begins the productive use and evolution of the solution.

## Creating a Migration Project Plan

Now that you understand the QUOTE<sup>®</sup> System, it becomes important to see how it maps out to the Desktop Deployment Lifecycle (DDL) presented in Chapter 1 (see Figure 2.9). As you can see, it maps out as follows:

- Question Phase: step 1, generate the business case and step 2, strategy development
- Understand Phase: step 3, build the test environment and step 4, identify toolkits
- Organize Phase: step 5, infrastructure analysis & update, step 6, applications strategy, step 7, pc image creation, step 8, personality protection strategy and the first part of step 9, integration strategy
- Transfer Phase: the second part of step 9, integration strategy and the first part of step 10, deployment strategy
- Evaluate Phase: the second part of step 10, deployment strategy

As you can see, the bulk of the technical work is within the Organize phase.

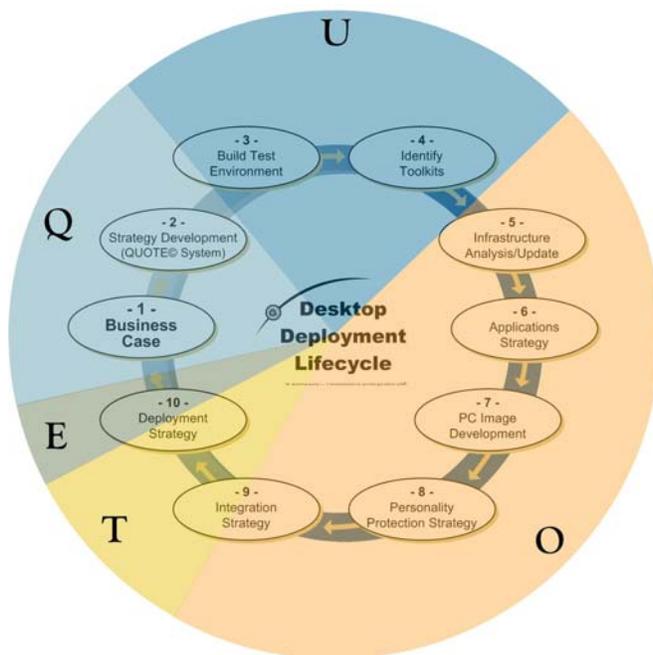


Figure 2.9: Mapping the QUOTE<sup>®</sup> System to the desktop deployment lifecycle.

## Project Planning Considerations

Each IT project—whether a deployment project or not—relies on the five phases of the QUOTE<sup>®</sup> System. You should structure your project to align with these phases. But even with such an alignment, you'll need to consider the actual objectives you aim to achieve with this project. In Chapter 1, we discussed the ability to rely on this project to finally gain complete control over your PC environment. We reiterate it again: if you don't have control now, then get it; if you have control now, then refine it.

### Making Common Mistakes

Don't make the most common mistakes with your project. For example, make sure the technical lead isn't also the project manager. Both roles are vital and need to be assigned to different people. Deployment projects are intricate, not to say complicated. Make sure you do not underestimate the level of work required to make this project a success.

Also, define your scope and stick to it. As time goes by, others will want to add this or that to your project scope. Don't let scope creep be the downfall of your project.

If you choose to lock down the desktops and you should, don't let anyone become an exception. In projects we have run, less than one percent of users have full administrative access and then, only through the Run As command. We'll give you tools to make it work in future chapters.

Make sure you track project progress and report on it to the Steering Committee. You might even want to put it into the communications plan—sort of a series of coming soon announcements—to build anticipation for the release and to ensure management knows how well you're doing.

Worst of all; don't let politics dictate how the project will run. Public sector organizations are the worst for this, unfortunately for them, but private organizations are not immune to this either. This is one more reason why communications are so important. Offset politics by managing expectations.

Finally, test, test and retest. This is the best advice we can give. Don't make the mistake of delivering poor quality products. If you are in a service industry, then quality of service is how success is measured. Go for the gold!

There are four pillars to any project: Cost, Scope, Quality and Time (see Figure 2.10). Each is directly tied to the other. If you change the value for one, the value for the others will automatically change. As the project proceeds, you will be under pressure to do exactly that. When deadlines loom, management will want you to cut on quality to save money, but short-term quality cuts will turn into long-term costs. The same thing with scope; management and others will pressure you to get rid of technical 'wish lists'. Don't do it. Each time you get rid of a project component to try to cut corners, you'll have to pay for it in the end. Build some leeway into your project plan so that you can have some flexibility, but aim for the highest quality. Your organization will benefit from it in the end.



Figure 2.10: The four pillars of the project.

Go to the DGVM's [companion Web site](#) to obtain a copy of a **Sample Project Plan** relying on the QUOTE<sup>®</sup> System to help build out your own project. Remember that you can supplement its information with the information contained in the sample project plan Microsoft provides in the BDD.

## Case Study: Supporting a Migration

Organization type ..... Public Sector  
 Number of users ..... 2,500  
 Number of computers ..... 3,000  
 Project focus ..... Migration to Windows XP  
 Project duration ..... 8 months  
 Specific project focus ..... Migration from multiple operating systems to a single OS  
 Administrative type ..... Centralized

An organization that moves to a new operating system must use complete support mechanisms during the migration. These systems must support not only the users during their familiarization with new tools, but it must also support the migration process itself and the teams that operate within it. This is often more than an internal Help Desk can manage because their own personnel are already occupied. This organization chose to supplement its support program with special help for the duration of the migration. Thus they had to increase staffing levels according to the following table:

Group	Increase
Technical support personnel	40%
User support personnel	20%
Project information support personnel	10%
<b>Average temporary staff increase</b>	<b>24%</b>

This increase was based on the following key elements:

1. The architectural solution was very sound.
2. The deployment was very well structured.
3. The project used a very complete communications program.
4. Appropriate levels of training were offered.
5. The support team was given intensive training.
6. The project included a data conversion program.
7. Technicians were very well prepared before the beginning of the deployment.

These key elements helped the organization control all support costs during the migration. Most of the temporary personnel were used to replace ongoing operations and free internal technicians to work on the project, keeping all knowledge in house after the project.

### **Required Teams and Roles**

Also, now is the time to start building up your teams. Remember that we strongly recommend that you get two types of external help if you need it, but you probably will: expert help to supplement technical leadership for the project, and technical help to replace your staff in ongoing operations and free them to work with your project team and gain immediate and direct knowledge on the new technology. After all, one of your main goals is to make sure this knowledge stays in house after the project. If you acquire external technical help and put them on the project, the knowledge will leave with them when the project is over. Don't make this mistake!

Rely on the DDL to determine which teams are required. Build on the team you gathered for the preparation of the Business Case. In the end, you'll require the following team roles:

- Logical Architect or Technical Lead
- Technical Architect
- Project Manager
- Project Administrator/Coordinator
- Lab Coordinator
- Lab Assistant
- Project Steering Committee
  - Project Sponsor
  - Business Unit Manager(s)
  - Chief Technical or Information Officer
  - Upper Management
- IT Team Members
  - Inventory Analysis
  - Infrastructure Assessment
  - Security Considerations
  - Application Compatibility Analysis
  - Application Rationalization
  - Application Packaging
  - System Image Preparation
  - Personality Capture/Restore
  - Personality Rationalization
  - Deployment Mechanisms
  - System Integration

- Support or Help Desk member(s)
- User Representative(s)
  - Software Product Owners
  - Standard Users
  - Power Users
  - Business Unit Experts
  - Local Application Developer(s)
- Financial Analyst or Asset Management Expert
- Communications
  - Communications Lead
  - Technical Writers
  - Relay Agents
- Training
  - Training Lead
  - Technical Writers
  - Technical Trainers
  - End User Trainers

Remember that these roles are important even if you don't have a massive number of PCs to deploy. You may even have the same person playing several roles, and therefore, wearing many hats during the project. Just make sure each role is covered. Of course, in larger projects, you'll want actual teams and team leaders for each crucial activity in the project.

You're off and running. Your project is in its building stages and you now have a very good idea on how to proceed. Next, you need to begin preparing for all of the technical activities of this project. That's exactly where future chapters will take you. Meanwhile, you have a lot of work to do. Good luck!

 Remember, if you want to give us feedback on this project and the content of the eBook, feel free to do so. All you need to do is write us at [vistaguide@reso-net.com](mailto:vistaguide@reso-net.com).

## Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.