Realtime publishers
"Leading the Conversation"

The Definitive Guide™ To

Vista Migration

sponsored by
altiris®

*Danielle Ruest
and Nelson Ruest*

Copyright Statement

Chapter 9: Putting it all Together

So far, each technically-focused team has been working on their own as they prepare their portion of the technical solution. By now, you should be in the middle of the Organize phase of the QUOTE System with each team progressing towards completion of their engineering activities (see Figure 9.1). It is now time to first, review the status of each solution component, mapping them out to the original logical solution design, correct any defects or deviances if they appear and then, begin integrating each solution component into one cohesive whole that will provide one single flow of operations during the rollout. The objective of this activity is to make sure that each and every engineering aspect of your solution works as expected in every situation and have it accepted by your peers, especially those that will be responsible for these aspects of IT operation once the solution is in place.

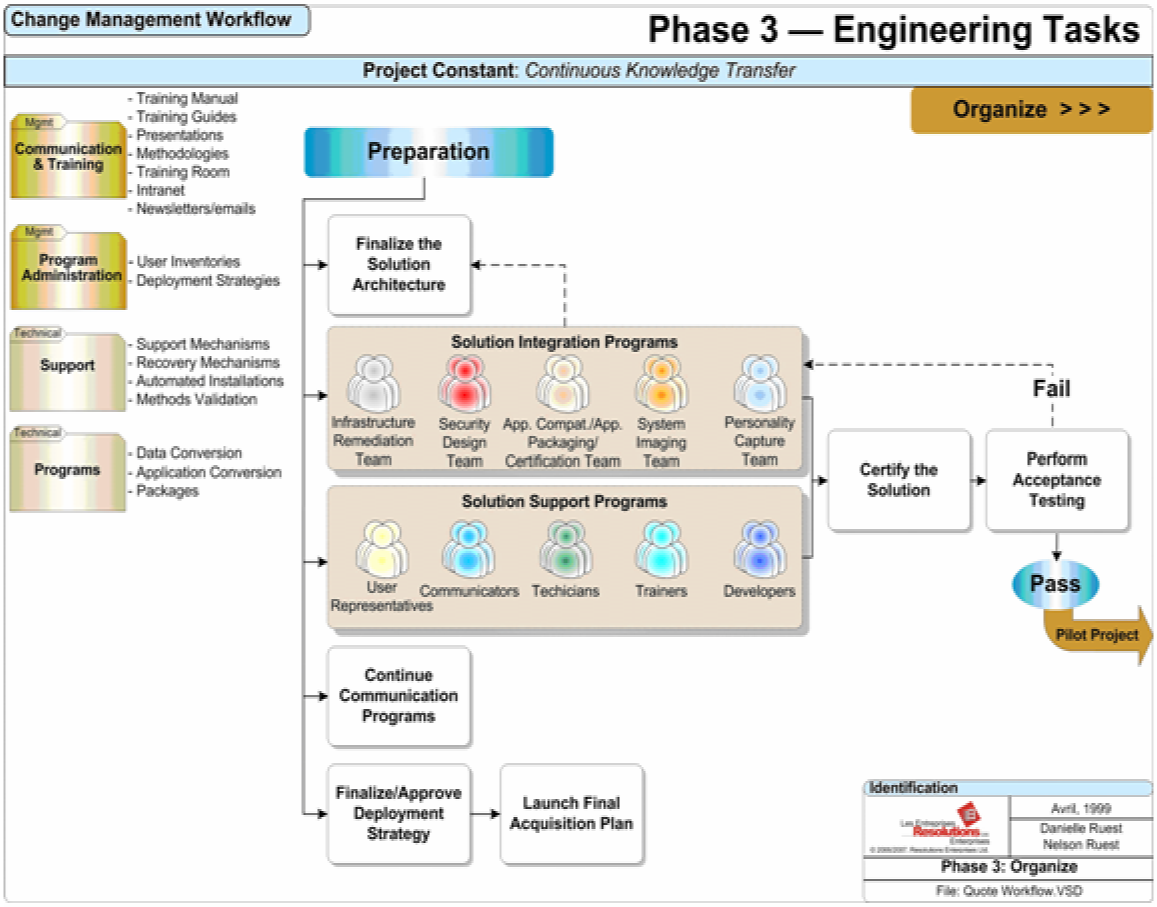


Figure 9.1. The Organize Phase of the QUOTE System

Of course, other personnel have also been working through the activities in the project plan as outlined in Chapter 2—activities that are more administrative in nature such as training and communications preparation—and these activities are soon to be tested and integrated into the migration process, something that will be covered in the next chapter as we run through the Pilot project. But, before you can focus on the overall migration picture, you need to make sure all engineering aspects are up to snuff. To date, you've had five technically-focused teams working out pieces of the solution. They include:

- Infrastructure remediation
- Security design
- Application preparation
- System imaging
- Personality protection

In fact, the technical team has been following the Desktop Deployment Lifecycle (see Figure 9.2). Though they have not been working in a vacuum—if you planned your project well, team members have been shared between the different technical activities so that they can share their expertise with others—they have, however, been focused mostly on their own aspect of the solution.

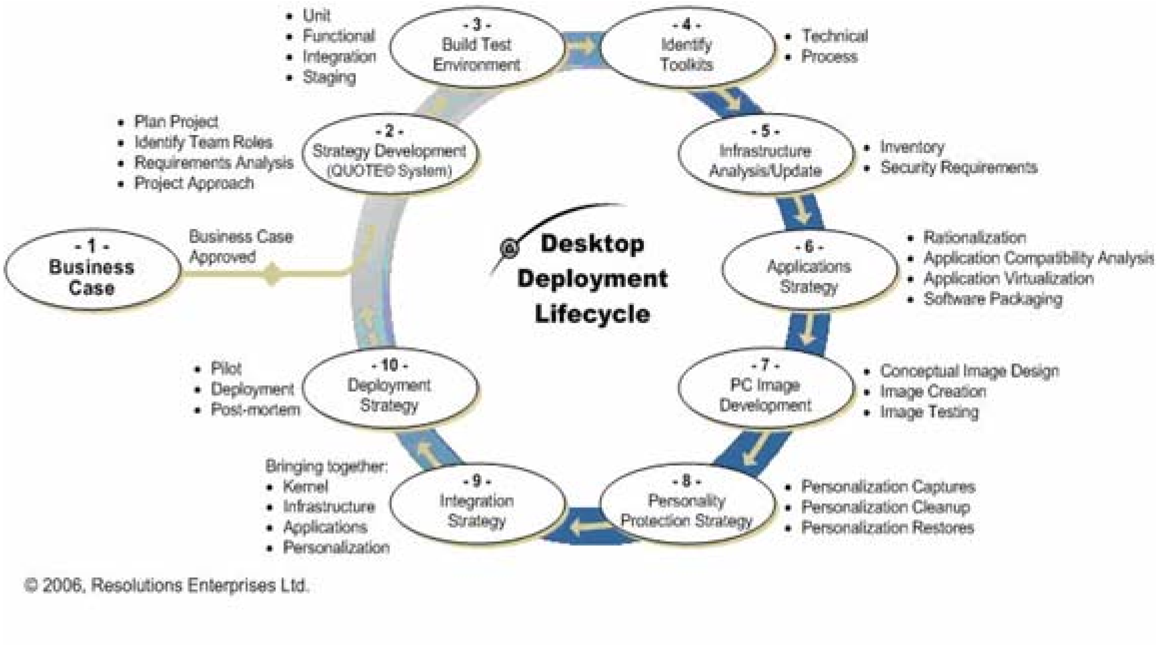
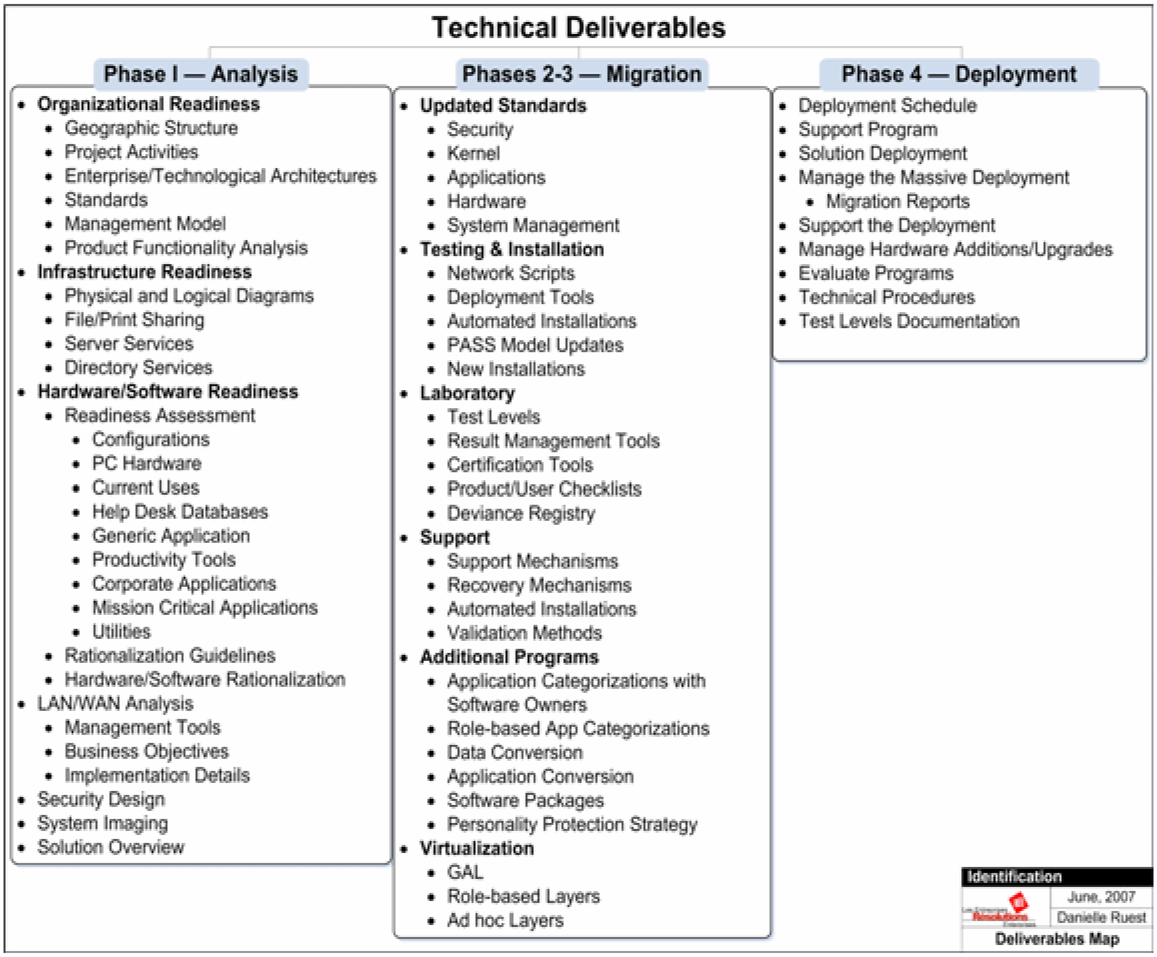

- 3 - Build Test Environment
• Unit
• Functional
• Integration
• Staging
- 4 - Identify Toolkits
• Technical
• Process
- 2 - Strategy Development (QUOTE© System)
• Plan Project
• Identify Team Roles
• Requirements Analysis
• Project Approach
- 5 - Infrastructure Analysis/Update
• Inventory
• Security Requirements
- 1 - Business Case
Business Case Approved
Desktop Deployment Lifecycle
- 6 - Applications Strategy
• Rationalization
• Application Compatibility Analysis
• Application Virtualization
• Software Packaging
- 10 - Deployment Strategy
• Pilot
• Deployment
• Post-mortem
- 7 - PC Image Development
• Conceptual Image Design
• Image Creation
• Image Testing
- 9 - Integration Strategy
Bringing together:
• Kernel
• Infrastructure
• Applications
• Personalization
- 8 - Personality Protection Strategy
• Personalization Captures
• Personalization Cleanup
• Personalization Restores
© 2006, Resolutions Enterprises Ltd.

Figure 9.2. The Desktop Deployment Lifecycle

As each team progresses through its solution design, they also progress through the testing levels. This means that they are now arriving into the Integration and Staging testing levels—levels that require more structure and precision than their predecessors. Remember that one of the features of the Unit and Functional testing levels is that they support undoable virtual disk drives, letting the teams 'go back in time' as they work with the preparation of their portion of the solution. But, Integration and Staging no longer offer this feature. Therefore, the teams must have more precise implementation strategies as they arrive into these environments.

At this point, these teams must also have produced preliminary versions of their technical deliverables. These deliverables should include detailed procedural documentation for the implementation of each part of the solution (see Figure 9.3). These deliverables will provide you with the roadmap for the actual implementation. The Integration testing level provides a first fully-functional network level for this first integration test. Staging provides a much more comprehensive testing level that simulates as many aspects of the production network as possible. For example, if your organization has remote sites, then the Staging level will simulate these remotes sites and allow you to test the solution over simulated or real WAN links. Also, because it is the last level before you get into the actual production network, there should be as few mistakes as possible when you get to Staging. Staging is also the level where you will be performing the last level of acceptance testing. In fact, you must have each portion of the solution accepted by the appropriate authority before you can move out of the Staging testing level and move into the production environment. Part of this acceptance testing will require you to perform a proof of concept, migrating or perhaps re-migrating at this stage, the systems of each member of the project team and perhaps even some key members of the organization, though they should normally wait until the Pilot project.



Technical Deliverables

Phase I — Analysis

- **Organizational Readiness**
  - Geographic Structure
  - Project Activities
  - Enterprise/Technological Architectures
  - Standards
  - Management Model
  - Product Functionality Analysis
- **Infrastructure Readiness**
  - Physical and Logical Diagrams
  - File/Print Sharing
  - Server Services
  - Directory Services
- **Hardware/Software Readiness**
  - Readiness Assessment
    - Configurations
    - PC Hardware
    - Current Uses
    - Help Desk Databases
    - Generic Application
    - Productivity Tools
    - Corporate Applications
    - Mission Critical Applications
    - Utilities
  - Rationalization Guidelines
  - Hardware/Software Rationalization
- LAN/WAN Analysis
  - Management Tools
  - Business Objectives
  - Implementation Details
- Security Design
- System Imaging
- Solution Overview

Phases 2-3 — Migration

- **Updated Standards**
  - Security
  - Kernel
  - Applications
  - Hardware
  - System Management
- **Testing & Installation**
  - Network Scripts
  - Deployment Tools
  - Automated Installations
  - PASS Model Updates
  - New Installations
- **Laboratory**
  - Test Levels
  - Result Management Tools
  - Certification Tools
  - Product/User Checklists
  - Deviance Registry
- **Support**
  - Support Mechanisms
  - Recovery Mechanisms
  - Automated Installations
  - Validation Methods
- **Additional Programs**
  - Application Categorizations with Software Owners
  - Role-based App Categorizations
  - Data Conversion
  - Application Conversion
  - Software Packages
  - Personality Protection Strategy
- **Virtualization**
  - GAL
  - Role-based Layers
  - Ad hoc Layers

Phase 4 — Deployment

- Deployment Schedule
- Support Program
- Solution Deployment
- Manage the Massive Deployment
  - Migration Reports
- Support the Deployment
- Manage Hardware Additions/Upgrades
- Evaluate Programs
- Technical Procedures
- Test Levels Documentation

Identification
June, 2007
Danielle Ruest
Deliverables Map

Figure 9.3. The Technical Deliverables Roadmap

Bring the Teams Together

To bring all teams together and perform the deployment of the solution within both Integration and Staging, you'll need once again to rely on the PC Migration Cycle (see Figure 9.4) since everything needs to be implemented in the proper order to provide the best level of integration.

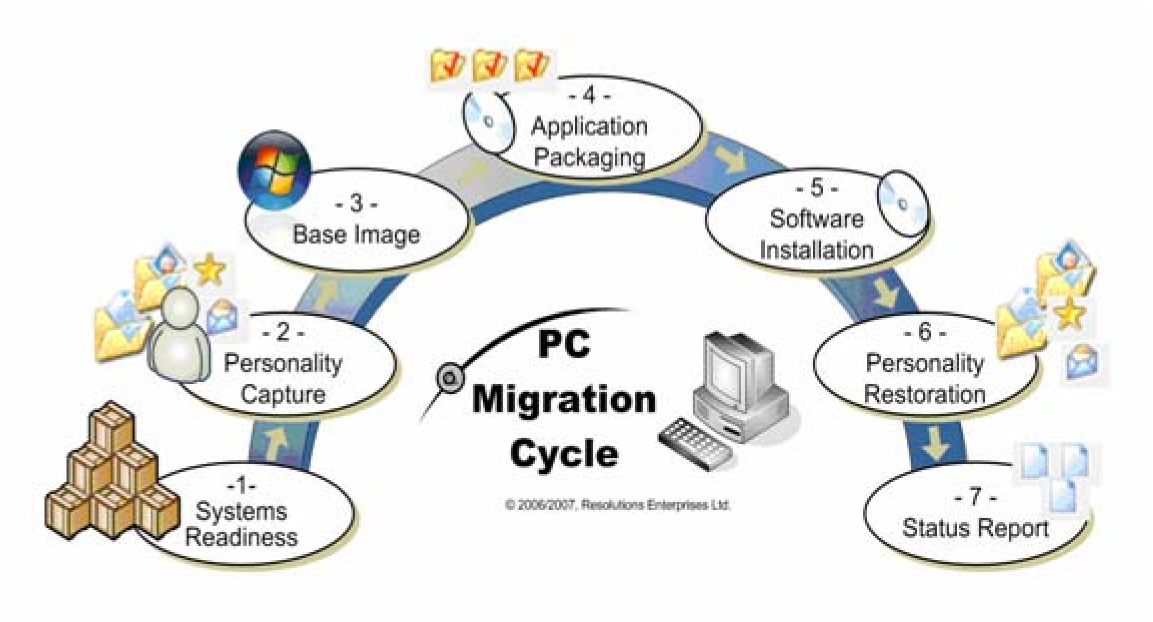


Figure 9.4. The PC Migration Cycle

As illustrated in the PC Migration Cycle, everything starts with Systems Readiness; this means that you will need to look to your server team to begin the integration process. Let's see what it entails.

✎ No matter what the size of your organization, it is always best to rely on a systems management tool for your migration. It's true, as pointed out in Chapter 4, Microsoft offers several free tools for desktop migration, most notably the new tools available for Vista through the Windows Automated Installation Kit (http://www.microsoft.com/downloads/details.aspx?familyid=C7D4BC6D-15F3-4284-9123-679830D629F2&displaylang=en)—which you will need in any case—and through the Business Desktop Deployment Workbench (http://technet.microsoft.com/en-us/library/53fa01fb-6aae-4230-ac26-4ed036a50908.aspx). But even though these tools are a vast improvement over their predecessors, they still rely on several command-line interfaces which make the process more complex by default.

If at all possible, we highly recommend that you obtain and rely on a real commercial systems management product. Once again, Chapter 4 listed several of these—Altiris, LANDesk, Microsoft, Special Operations Software, Symantec and more—and we do hope that by now, you've seen the true value these products offer and you have integrated the appropriate one into your project. After all, this migration is not a one-time event. Once you're finished moving to Vista, you'll need the right tool to manage ongoing system changes. Batch files and command-line systems just don't cut it in the long run.

Move into the Integration Testing Level

As detailed in Chapter 3, the Integration environment already offers several different system features (see Figure 9.5):

- A directory server running services such as Active Directory
- A database server running a version of SQL Server (ideally, version 2005)
- A file server running the file services role
- A deployment server
- An inventory server

Depending on your situation, your deployment and inventory servers may or may not be configured. If you already had a systems management product and you were satisfied with it, then these servers should be running the same version of these services that you have in your production network. If you did not have a systems management tool and you made the (right!) decision to obtain one, then these servers are just member servers awaiting the installation of the roles on them.

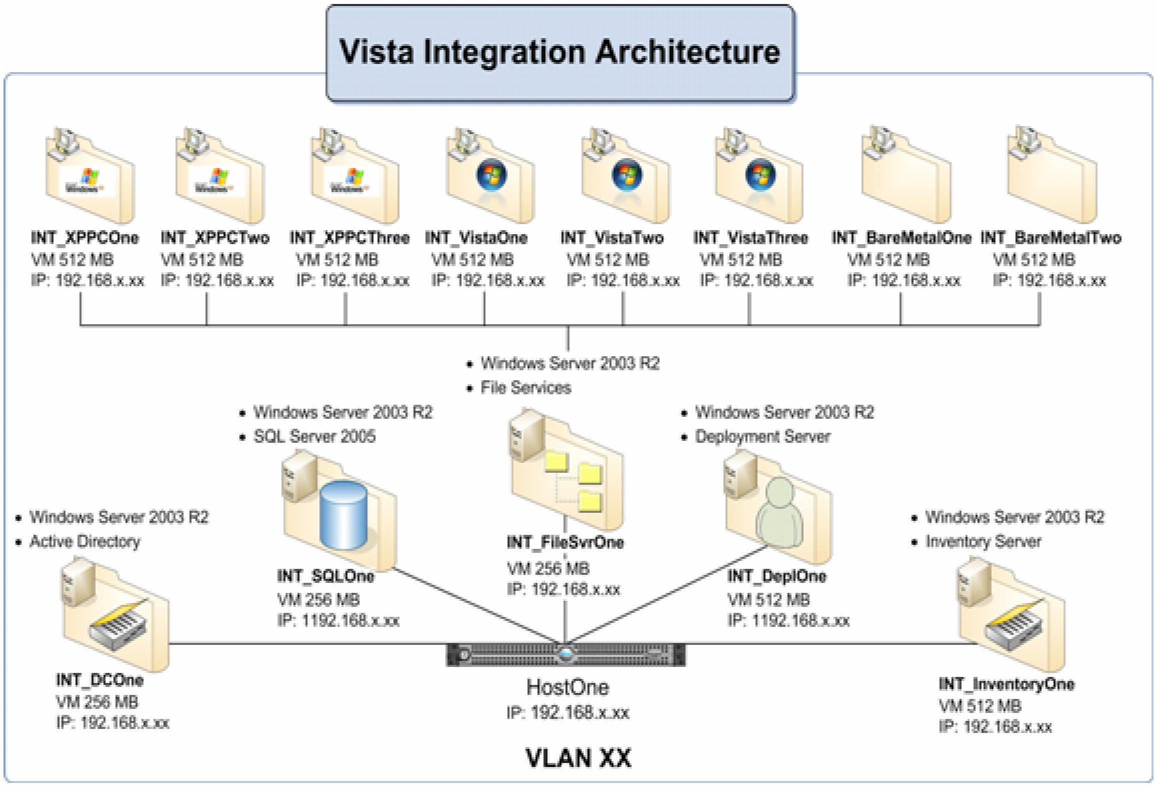


Figure 9.5. The Content of the Integration Testing Level

✐ Note that this testing level includes separate servers for database, deployment and inventory. Depending on the size of your organization and the type of systems management tool you purchased, you may only need a single server for each of these roles. Many organizations that deploy Microsoft's System Center Configuration Manager (formerly Systems Management Server) often do it on a single server even if they have larger organizations because it is the easiest way to deploy and manage this complex product.

What is most important here is that this environment must have divided roles as much as possible. Each of the previous environments—Unit and Functional—have all of the server roles integrated into a single server. Many, many people make the mistake of developing a solution that runs fine when a single server runs all the roles, including the domain controller (DC), because they test everything with 'god' rights. It is admittedly hard to avoid when you are on a DC. This is why it is so important to start separating roles as much as possible in the Integration level. When roles are separated, you have to use proper delegation of authority to run your tests and if they don't work here, then you **know** they won't work in production. This is the main goal of this testing level.

The first place to start the integration process is with the servers. Chapter 5 outlined all of the different processes that were under the aegis of the server team. This is the team that will begin the integration process. To do so, they will need to perform the following activities:

- Install/update the inventory tool
- Install/update the profile sizing inventory tool
- Install/update the OS deployment tool
- Install/update the application deployment tool or install an application streaming tool
- Prepare the application packaging or application streaming environment
- Prepare the central repository for Vista Group Policy templates (ADMX files)
- Prepare Vista Group Policy objects (GPO)
- Prepare required shared folders and assign appropriate security rights to them
- Prepare Distributed File System (DFS) namespaces and replication systems
- Prepare the central Key Management Server to manage Vista licenses

Because they have had the opportunity to work on this in two previous testing environments, the server team should have documented processes and perhaps even automated processes for these implementations. As you have seen in previous chapters, some of these operations are relatively straightforward while others are more complex.

For example, if you already have a systems management tool in place, you'll need to simply update it. This means that it should be installed and configured according to your existing standards and strategy and then, it should be updated to include Vista support. This update procedure is what needs to be fully documented so that it works without a hitch. If you don't have an existing systems management tool, then you'll need to have a full-fledged architecture designed for the implementation of the new tool. This architecture and the procedures that support it is what you'll want to test at this stage.

When the server components have been updated, then the other team members can begin their own integration. As mentioned earlier, the Integration environment will already include several components that will allow these PC-focused team members to work with their own solution components. These activities will include the following:

- Implementation of the personality protection system
- Implementation of the OS imaging system
- Packaging or sequencing of the applications that were retained after the rationalization process
- Implementation of the application deployment or streaming mechanism
- Implementation of the Vista security policy as was designed by the security-focused team

Once each of these systems is in place, you move to an end-to-end test of the PC Migration Cycle. Here all team members learn to 'play nice' with each other as each part of the process is tested. You'll need to test each of the migration scenarios you've elected to support: upgrade, refresh and/or replace. The Integration environment should include the appropriate machine types—Windows XP and bare metal systems—to let you test whichever scenario you selected. The purpose of these tests is to make sure everything works as expected and to refine any procedural documentation you need to rely on.

Maintain the Deviance Registry

It is also be very important to manage the deviance registry at this level. This is where you begin to map the logical solution design to the actual physical design as it is implemented. For example, you will want to compare the logical description of the PASS System Kernel against the actual contents of an installed Vista system. You can also compare the personality protection policy against the actual results of a personality capture and restoration. You can compare the structure of the Generalized Application Layer (GAL) against the actual results installed on a system once it is deployed.

Each time you find a change or a result that is not as expected—a deviance, in fact—you need to run it through the paces. How important is it? Is it a show-stopper or is it nice to have? Who defines these categories and do they have the proper authority to do so? In the end, you'll end up compromising—every project does—as project resources, costs, scope and quality begin to affect the delivery timeline.

Managing the deviance registry is one of the toughest jobs in the project. That's because it is this registry that will help define the final solution you will be putting in place. This is why it is so important to try to test out as many potential situations as possible while you're in the Integration environment. Try to simulate every possible situation you will be facing in the actual deployment. Record the results, provide corrections, and then, when you've signed off on all of these tests, you can move on to the Staging test level.

Move to the Staging Testing Level

As mentioned earlier, the purpose of the Staging level is to simulate production as much as possible. This is why it contains more than the Integration level (see Figure 9.6). In fact, it includes everything you can find in Integration plus the following items:

- Multiple domain controllers
- Multiple file servers
- Multiple management servers
- A simulated remote site to simulate WAN deployments

In addition, this level includes the possibility of a test that cannot be done in other levels: the proof of concept (POC). This proof of concept is a live test that is performed against the systems in use by the project team. Their systems are upgraded through the use of the new deployment mechanisms integrated into the Staging testing level. This lets you perform an end-to-end test that will ensure the technical solution actually works as expected in a live setting.

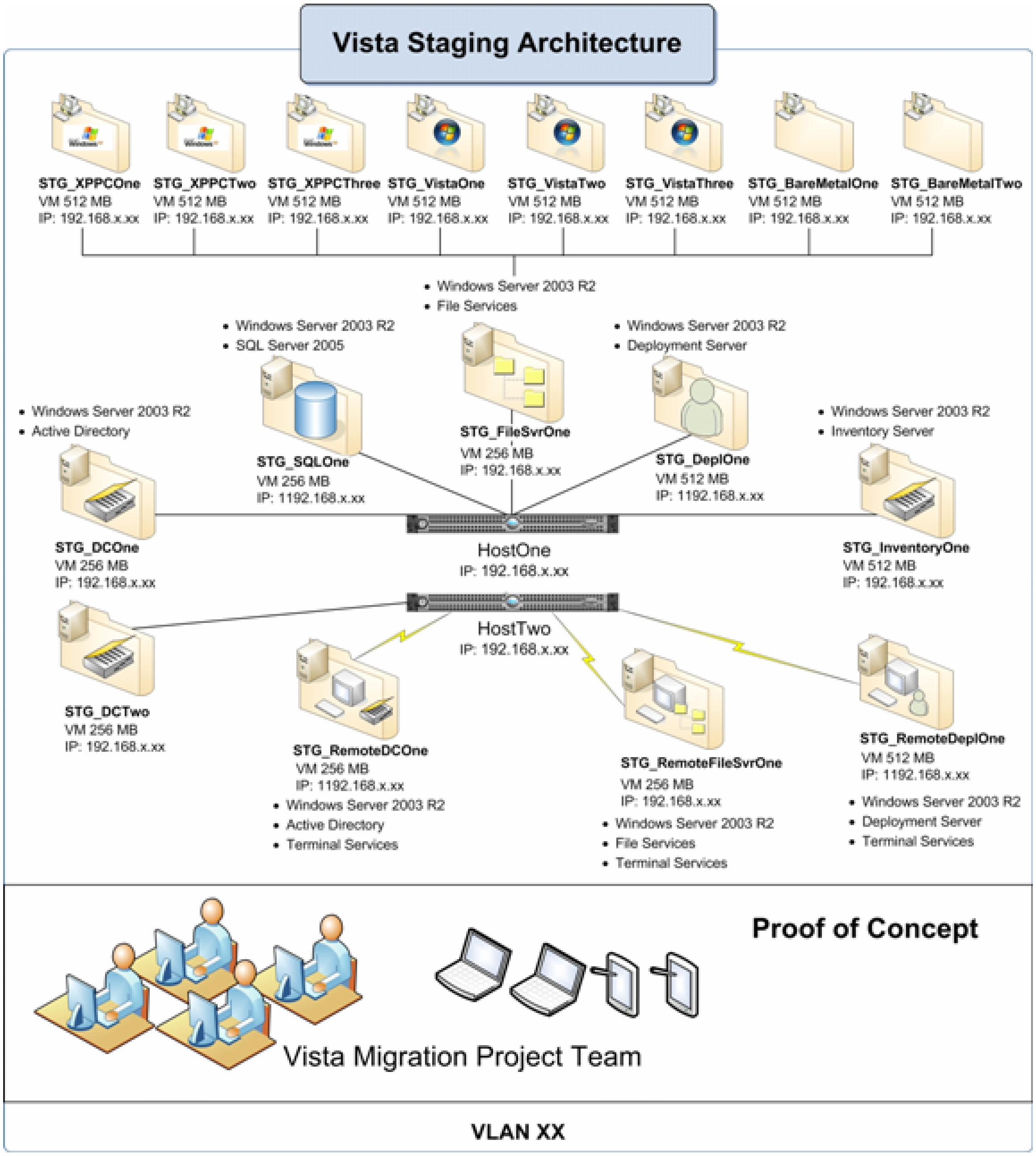


Figure 9.6. The Content of the Staging Testing Level

But before you perform the POC test, you need to test anything that you could not test in the Integration level. Focus on those features that are included in the Staging level, but are not in the Integration level. This involves Group Policy replication between multiple DCs, migration repository replication between sites, capture of personality data in remote sites and then replication to the main site for backup purposes, and simply testing WAN load for the migration. This should help flesh out your solution to meet real-world needs.

> ✐ Make sure all servers are running Windows Server 2003 R2 so you can take advantage of its new and updated DFS features. For some of you, this may mean that you need to perform server upgrades as well in your production infrastructure.

Review the Migration Workflow

When you feel you're ready, you can move on to the POC. To perform this test, you'll need to run through the migration workflow (see Figure 9.7). This workflow is the process you use to perform any single migration. Of course, it relies very heavily on the PC Migration Cycle, but because it involves migrating a real computer that is in use, it also involves other activities. Each is covered in the workflow.

As you can see, this workflow begins well before the actual migration. It includes the following activities:

1. Everything begins with a communication to the user to inform them of the upcoming change. The details of this communication will be discussed in Chapter 10 as we discuss the administrative aspects of the deployment.
2. The technical aspects of the migration begin with a detailed inventory of the PC to be migrated. This includes an inventory of existing profiles.
3. Next, you apply rationalization principles to what is found on the original system. Rationalize applications, hardware and profiles. This may require a visit to the user to validate the rationalization results.
4. Produce the final system datasheet for this PC.
5. Hand off this datasheet to the release manager who will assign the task to the deployment team.
6. The deployment team will determine if any hardware modifications are required—upgrades or even a PC replacement.
7. Map out the applications required for this user post-OS install.
8. Perform the personality capture. Validate the data. Optionally perform a full PC backup.
9. Perform any required hardware modifications.
10. Deploy the OS to the system. This can involve a disk wipe, possible repartitioning to take advantage of features such as BitLocker and then an OS installation of the thin kernel.
11. Stream the Generalized Application Layer to the system.
12. Stream any role-based or ad hoc applications.
13. Restore the personality to the system.
14. Validate the deployment and produce the PC migration report.
15. Inform the migration support team that this PC is now migrated.

As you can see, there are several steps to a single PC migration.

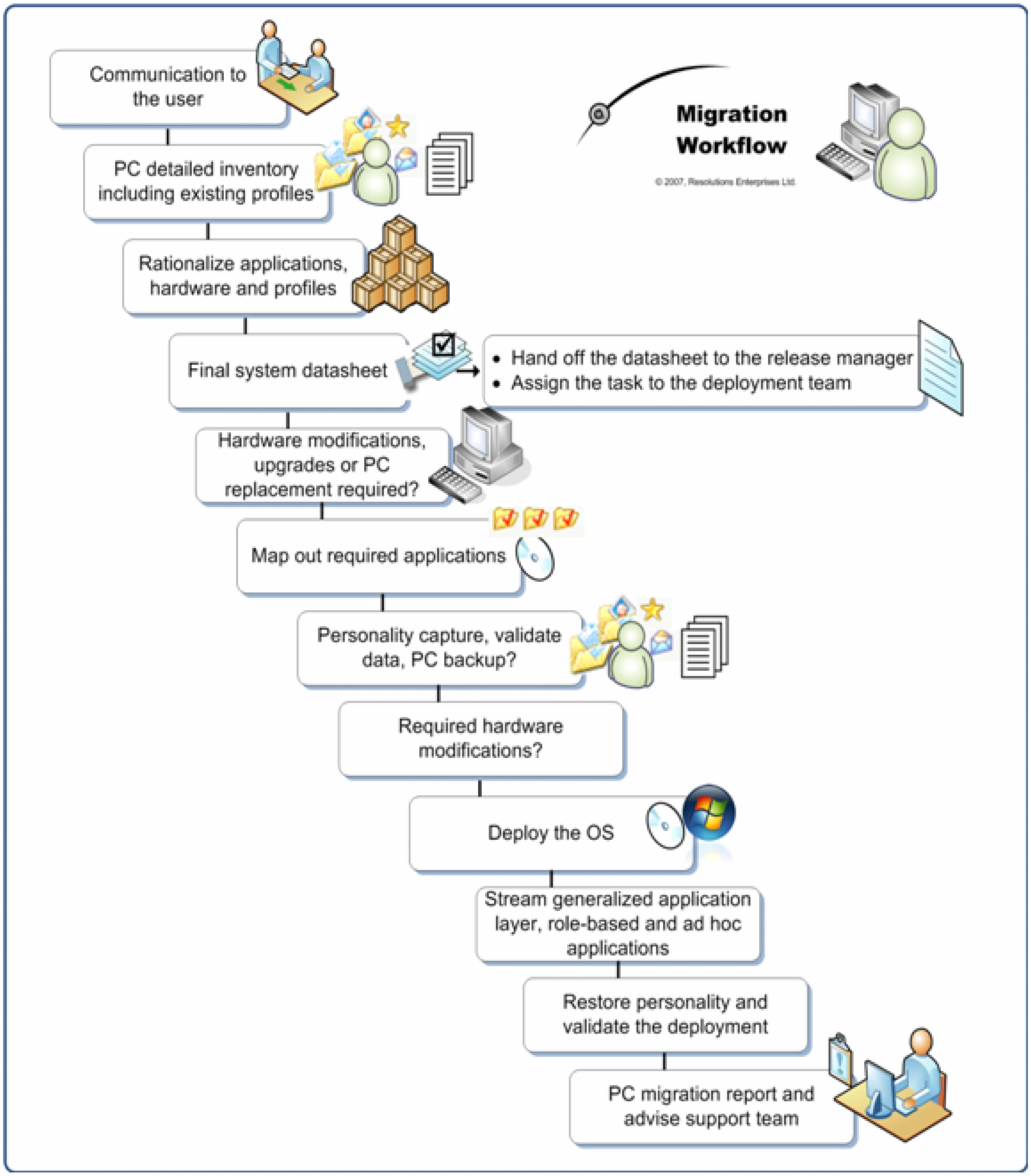

Migration Workflow
© 2007, Resolutions Enterprises Ltd.
Communication to the user
PC detailed inventory including existing profiles
Rationalize applications, hardware and profiles
Final system datasheet
• Hand off the datasheet to the release manager
• Assign the task to the deployment team
Hardware modifications, upgrades or PC replacement required?
Map out required applications
Personality capture, validate data, PC backup?
Required hardware modifications?
Deploy the OS
Stream generalized application layer, role-based and ad hoc applications
Restore personality and validate the deployment
PC migration report and advise support team

Figure 9.7. The Migration Workflow

Perform the Proof of Concept Project Team Migration

You should now be ready to perform your POC with the project team. This requires a few prerequisites:

- Since you will be using the Staging environment as your source for the migration, the PCs used by your project team should be joined to the Staging domain to fully support the migration. Be sure to be extremely careful with their profiles when you do this. You might consider using your profile protection system to make sure users continue to have access to all their data.
- While the Staging environment does simulate production as much as possible it may be missing some services. For example, you may not have an Exchange server in Staging, therefore, you will need to have users connect to the production Exchange server. This can easily be done through the use of RPC over HTTP. You might also not have all of the file shares users are used to working with. Show them how to remap these shares and make them persistent with their production domain accounts.
- If users are working with Windows SharePoint Services (WSS) between the Staging and the production domains, show them how they can avoid logon prompts by placing these Web sites into the Intranet Zone of Internet Explorer.

The goal is to make the experience as normal as possible for all project team members while performing a full end-to-end technical test. This means you might omit some applications on purpose for some systems so you can test your migration support strategy. This support strategy should include the ability to fast track application deployments to users when they are found missing among other items.

> ✎ Don’t do this test until you have fully tested each other technical aspect in the Staging environment. Your solution should be as ready to go as possible before you get to this stage otherwise you will have to repeat it several times. You can imagine how annoying this would be for your team members.

You also need to run this POC for some time. A minimum of two weeks is required and if you can make it longer, then you will find out much more. The project team will have advice about all sorts of things from the position of one icon to the structure of the file system on the desktop. Make sure you capture each comment—once again the Project Team Site in Windows SharePoint Services is probably the best location for this type of feedback. Use a triage system just like the one used for the deviance registry to identify what is important and what is not.

Once again the goal is to perform a very tight quality assurance (QA) on each and every technical aspect of the project. Also, make sure the in-process reporting works as expected through the migration. And, of course, make sure the end result is as expected. Simulate as many problematic situations as possible. For example, begin the deployment and cut of the network connection in the middle of the operation to see how you can recover from this situation. If you’re ready for any situation, your deployment project will provide much better results.

Obtain Acceptance from Peers

The last thing to do before you can be deemed ready to exit from the Staging environment and move on to the final stages of your project is to obtain final acceptance from your peers. This involves the following groups:

- **Architectural team**: the internal architectural team needs to approve all changes to both server and workstation architecture.
- **Support team**: the support team needs to approve and accept the new support methods proposed by the project for the new OS. Of course, the project will be responsible for immediate migration support for the duration of the project warranty period (usually 20 working days), but after that, the systems pass on to normal support operations.
- **Subject matter experts (SME) and software owners**: each application package, custom or commercial must be accepted by the designated SMEs in your organization. Remember that the project team cannot take on this responsibility since it cannot possibly include experts on every product in use in your organization.
- **Operations team**: the operations team needs to approve the change in operations and the new operational strategies designed by the project before the project can move full steam ahead with the migration. This is probably the most important technical approval level required by the project because it is the operations team who will live with these changes once the project is complete. Because of this, more on this subject is covered at the end of this chapter.
- **Internal developers**: this team needs to approve the changes to their applications and their programming strategies. Make sure they are involved early on in the project.
- **End user developers**: these developers work with applications such as Word macros, automated Excel spreadsheets, and Access databases. If you did the smart thing, you will have made sure they use proper client server programming practices to ensure you do not need to deploy a full version of Access to every end user, saving the organization both time and money.
- **Management**: upper management must approve the results of the project since they are funding all of the efforts.
- **Project management sign off**: the project management team must also sign off on the final results of your technical solution. Since they will have used the system for some time through the POC, they will be in the ideal situation to determine if the solution is mature enough for sign off.

If you don't get each of these sign offs, then you must return to the drawing board and revisit your strategies until you get them right.

Validate your Operational Strategies

One of the most important technical aspects of this project is the operations hand off—the hand off of the operational strategies that will be in use once the migration is complete. The operations team will be responsible for the ongoing management of all of the tools you put in place and use during the migration project. Your goal is to make sure the duties they will take on are easier and simpler to perform than what they are doing now.

Depending on the type of migration project you decide to run—for example, forklift versus attrition migrations—your project will either run for a very short period of time and then move into ongoing operations or, it will last a very long time because it is waiting upon your normal PC replacement cycle to complete. We would argue that both types of projects should run for the same duration.

In the traditional approach, forklift migrations perform a massive replacement of the operating system on all of the PCs in the network in one fell swoop. While the project takes about nine months to complete overall, the actual massive migration should only take about three months (see Figure 9.8). Then, once all of the systems are modified, you hand off to operations and the organization goes about its business as usual.

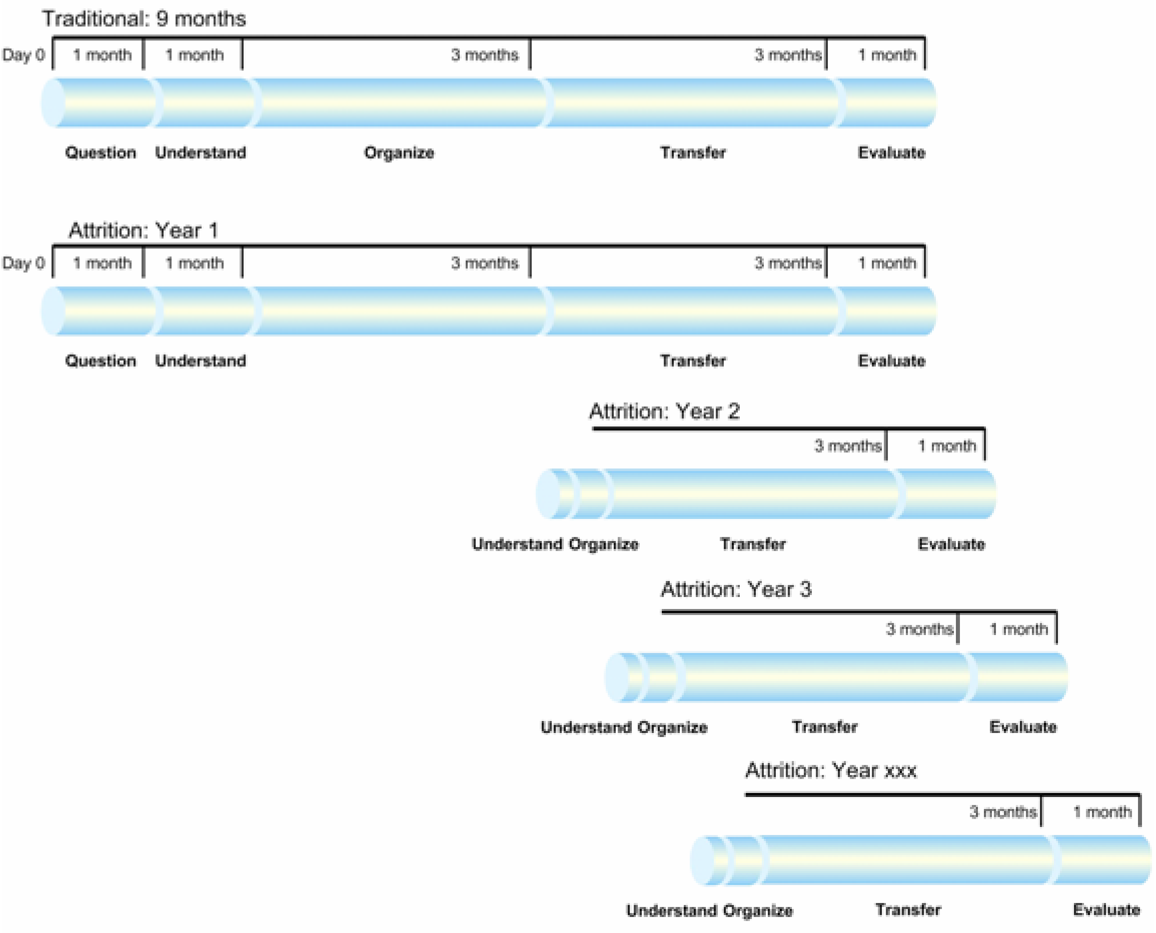


Figure 9.8. The Traditional versus an Attrition Migration Approach

In an approach based on attrition, the migration project will last as long as it takes to replace every PC in the network. For some, this may take three years while others may take four or even five years to complete the process. Each time a new batch of PCs comes in, the project picks it up and migrates them. But, there is a lot of opportunity for change during this process. For example, there will be a considerable number of updates and service packs to Windows Vista during a three to five year period. But most importantly, no single organization can run a project—any project—for three to five years and actually have it succeed. Well, let's say that very few organizations can hope to succeed at this level.

We would suggest that instead of running the project for this period of time, you should use the same timeframe as the forklift project and migrate only the first batch of PCs through the project. What needs to change is the nature of the project deliverables. This is where the operations hand off comes in. This hand off is important for both approaches, but it is much more important for the attrition approach because of its very nature.

The operations hand off for either project must include the following deliverables:

- Kernel Image Management Strategy
- Application Management Strategy
- Personality Protection Strategy
- Vista Security Management Strategy

In addition, both types of projects should include the following deliverables, but these are most important for the attrition project:

- New PC Integration Strategy
- PC Migration Strategy

To make the most of your current migration project, you should aim to transfer its overall operation into production at its end. Since it is unreasonable to expect that a migration project could possibly run for a number of years, you should make sure that you transition all migration processes into production. This way, the idea of an OS migration becomes part and parcel of normal, ongoing operations. When a new batch of PCs comes in the year following the closure of this project, the operations team is ready and knows how to transition these PCs with the new OS into production. Because you have provided a structured process for PC migration, the operations team does not need to do anything out ordinary and can proceed with business as usual. This is the best and easiest way to ensure success for such a long term project.

Let's look at some of these strategies.

Kernel Image Management Strategy

One of the outcomes of your project is the use of a system stack such as the PASS System Kernel. As you know, this kernel includes several different layers that are designed to provide service levels to end users. Because they include several different components and because of the nature of operating systems, Windows and others, you know that you will need to put a kernel update management system in place once the deployment is complete. Therefore, one of the technical deliverables you should expect is a Kernel Image Management Strategy.

This strategy must include how and when you update the kernel. As time goes by, you will be relying on the kernel to install and update systems on an ongoing basis. Of course, you don't want to find yourself in a situation where you need to deploy the kernel, then spend inordinate amounts of time waiting for it to update once installed. Conversely, you don't want to have to find yourself in a position where you are constantly returning to the reference computer, opening it up, updating it and then running it through Sysprep to depersonalize it again.

This means you need to find a happy medium between updating the kernel remotely and updating the actual kernel image. The best way to do this is to implement a formal update strategy.

This strategy should be based on a formal change request process. This change request process would rely on the rationalization guidelines that emerge from this project to evaluate the validity of the request. If approved, the change would then be delivered in the next kernel update (see Figure 9.9).

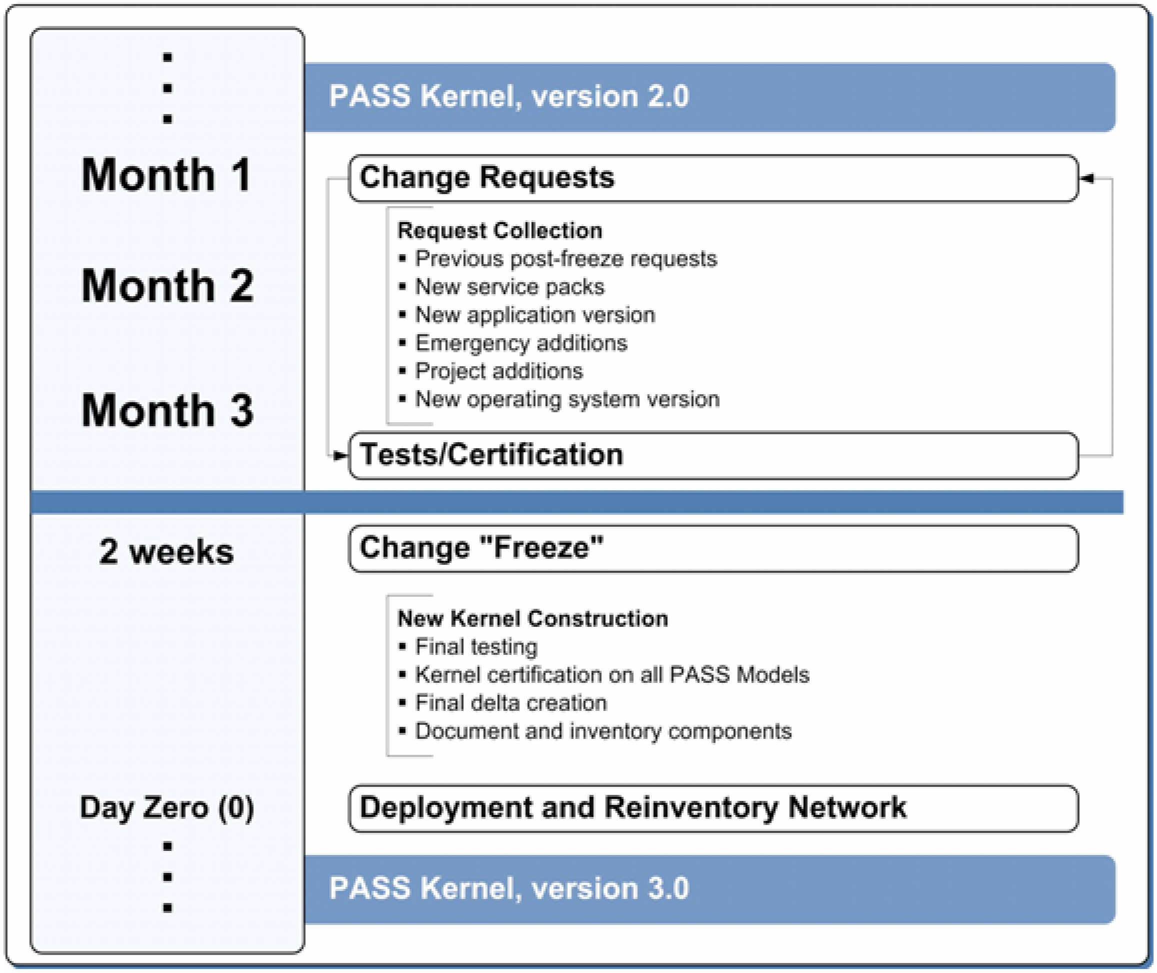


Figure 9.9. The Kernel Update Process

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 three-month basis at the very most. Waiting longer for update collection creates packages that are too unruly. Updating the kernel more frequently can become too expensive. In the meantime, the deployed copies of the kernel are updated through a centralized update management tool. Each time the kernel is updated, its version number changes. Four updates per year have proven to be the optimum rate of updates for the kernel in every organization using this process to date. Two weeks before release, a freeze is applied to change requests in order to allow for the preparation of the update.

Because Vista supports a single worldwide image, this process should be relatively simple to perform. In fact, you'll have at least two images if you include one for the x64 platform. It should therefore be much simpler than what you do today. In addition, if you maintain the reference computer inside a virtual machine, you will be able to greatly reduce the effort required to maintain the kernel since you won't have to rebuild the reference computer each time you need it.

Application Management Strategy

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

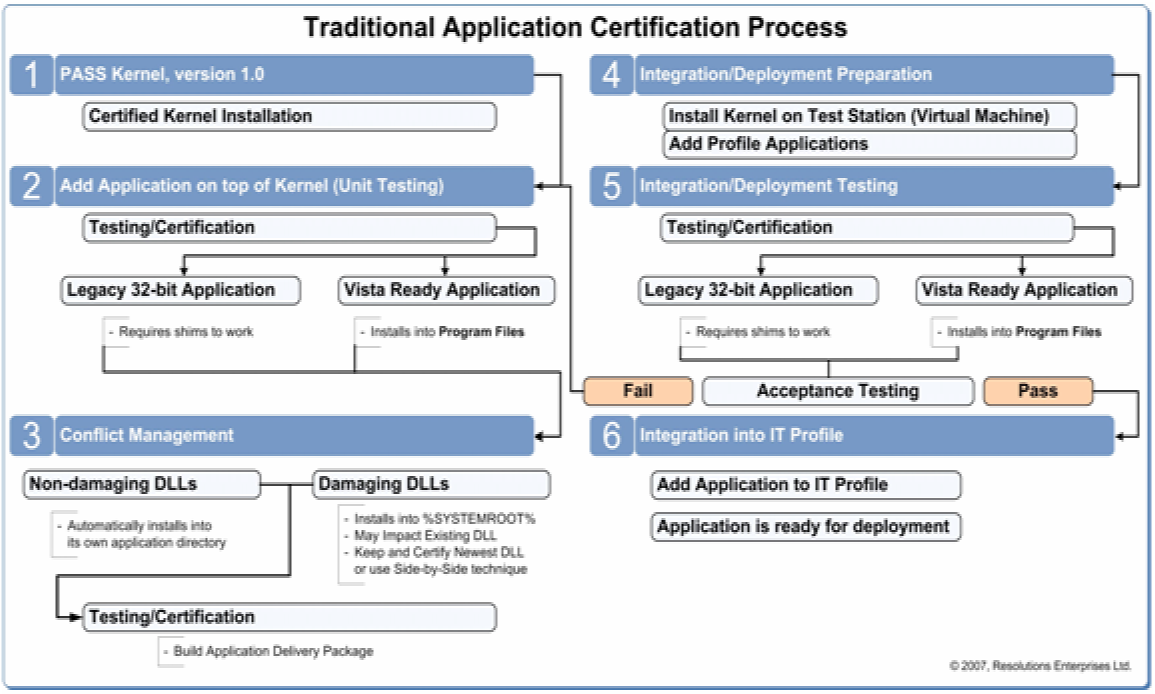

Traditional Application Certification Process
1
PASS Kernel, version 1.0
Certified Kernel Installation
2
Add Application on top of Kernel (Unit Testing)
Testing/Certification
Legacy 32-bit Application
Vista Ready Application
- Requires shims to work
- Installs into Program Files
3
Conflict Management
Non-damaging DLLs
Damaging DLLs
- Automatically installs into its own application directory
- Installs into %SYSTEMROOT%
- May Impact Existing DLL
- Keep and Certify Newest DLL or use Side-by-Side technique
Testing/Certification
- Build Application Delivery Package
4
Integration/Deployment Preparation
Install Kernel on Test Station (Virtual Machine)
Add Profile Applications
5
Integration/Deployment Testing
Testing/Certification
Legacy 32-bit Application
Vista Ready Application
- Requires shims to work
- Installs into Program Files
Fail
Acceptance Testing
Pass
6
Integration into IT Profile
Add Application to IT Profile
Application is ready for deployment
© 2007, Resolutions Enterprises Ltd.

Figure 9.10. The Traditional Application Preparation Process

If you are using traditional application installations, then you will need to perform conflict management—managing the potential conflicts caused by dynamic link libraries (DLL) included in the software product that may overwrite or otherwise damage existing system or application DLLs. To do so, you need to run each product through a testing phase similar to that of the kernel (see Figure 9.10) and you need to prepare and maintain an inventory of every single component that can exist on a PC. This is a very heavy management process; one more reason why having software owners is important because they at least take on the burden of software vigilance by watching for potential upgrades and new versions.

But, if you move to software virtualization, you'll still have to package and prepare applications and have them approved by the SMEs, but at least you won't have to worry about two aspects that are part of the traditional process:

- DLL management because the DLLs are all managed by the virtualization agent.
- Application deployment because when applications are streamed, they are only cached locally; change the central source of the stream and it will automatically be restreamed to all clients.

These are compelling reasons for this move. Another reason which we feel we don't tout often enough is the fact that *all of your system kernels are going to be pristine at all times* (see Figure 9.11) because the local application cache holds all of the deltas beyond the thin kernel. And, because it is a cache, it can be flushed at any time, returning your system to its pristine state, This has to be worth its weight in gold to you and your operations staff.

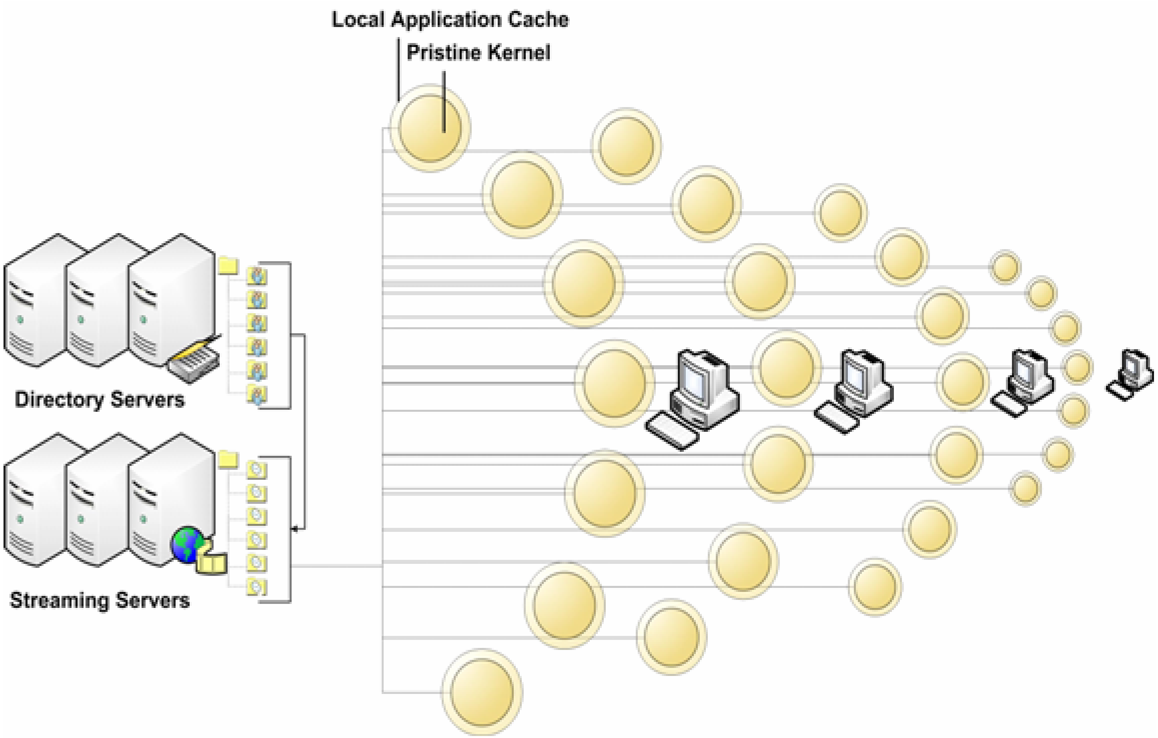


Figure 9.11. Working with Application Virtualization, Streaming and Pristine Kernels

Technical Outcomes of the Project

A migration project, especially a PC migration project, is a transformation project that lets you clean house and close the door to unwanted content. That's what the rationalization process is all about: get the dust out of the corners and make sure it doesn't come back. If this is not what you will get at the end of this project, then you wasted an excellent opportunity to take complete control of your PC environment. We believe you should, at the very least, expect the following technical results from this project:

- **Managed systems**: you should be deploying a complete infrastructure and strategy for long-term management of your PCs.
- **Thin system kernels**: the only software that actually installs on your PCs should be the operating system and anything that does not support application virtualization such as antivirus and low-level utilities. These systems will be pristine at all times and should be very easy to reset back to the standard should any issues arise.
- **Application virtualization and streaming**: you should be virtualizing everything from the GAL to ad hoc applications, anything you can in fact. This will avoid all potential conflicts, ensure any application will work at the same time as any other, facilitate software updates, facilitate software license management and prepare you for the long-term future of Windows PC evolution.
- **Locked-down systems**: your PCs should now be completely locked down. After all, if you put in the effort it requires to move to an application virtualization and thin kernel strategy, and then let users have administrative rights on their PCs, you will definitely not be able to take advantage of a pristine kernel at all times.
- **PC migration strategy**: you should have a defined PC migration strategy that not only supports actual migrations but also any PC replacement that occurs during normal operations.
- **Personality protection**: you should have a complete personality protection system that saves end user data and lets you back it up at any time. After all, with all of the new compliance regulations affecting almost any type of organization, it is a really good idea to have a central copy of all user data at all times. This way, you have access to any data whenever you need it.
- **Constant inventories**: you should have learned your lesson this time and made sure you have a constant and complete inventory system in place. Isn't it time you covered all your assets?
- **Secure infrastructure**: with Windows Vista's new security features, you should have a much more secure PC infrastructure once all systems are migrated.
- **Greener PCs**: using the new power-savvy Group Policy settings in Vista, you should be able to save considerably on power consumption for each PC in your network.
- **Reduced support costs**: because your systems are locked down and are now managed, support calls for PC issues should be greatly reduced. One of our customers who moved to locked-down PCs in 1998 reduced their PC-related help desk calls by more than 800 percent! You might not have this dramatic a drop, but you should definitely have to deal with less than ever before.

- **Lower overall Total Cost of Ownership (TCO)**: putting a new managed PC infrastructure in place based on Vista and application virtualization should provide considerable savings in terms overall costs for the operation of PCs, savings like you have never seen before.
- **A smart PC management strategy**: PCs have been a thorn in the side of IT professionals for years. Now, with the strategy outlined here, you can finally gain control of this aspect of your network once and for all. Be smart, get it right this time!

These aren't the only benefits you will derive from this project. One will be a complete change management strategy if you continue to use the QUOTE System for your other IT projects. Others will be derived from the other, administrative processes that are tied to the deployment. These will be covered in the last chapter of this guide.

Whatever you do, at this stage you need to make sure your technical expectations—those that were outlined so long ago in your original business case—are fully met and, if possible, exceeded. Then and only then will you feel you've made the proper investment and moved to the very best of platforms.

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.