››› Software Change Management

Best known methods for deploying software across the enterprise

Abstract:

Deploying operating systems, applications, patches and updates across enterprise networks is a complex process that requires input and approval from both business and technical staff. Standards such as the IT Infrastructure Library (ITIL) are designed to help companies establish processes to maintain service availability and deploy changes with a minimum of disruption.

Automated software distribution solutions such as LANDesk® Management Suite and LANDesk® Patch Manager can substantially reduce the time, network bandwidth and manual reconfiguration required to deploy changes into the computing environment, while reducing costs. This white paper reviews best practices for delivering software changes across complex network environments.

This document contains confidential and proprietary information of LANDesk Software ("LANDesk") and its affiliates and is provided in connection with the identified LANDesk product(s). No part of this document may be disclosed or copied without the prior written consent of LANDesk. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in LANDesk's terms and conditions of sale for such products, LANDesk assumes no liability whatsoever, and LANDesk disclaims any express or implied warranty, relating to sale and/or use of LANDesk products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. LANDesk products are not intended for use in medical, life saving, or life sustaining applications. LANDesk does not warrant that this material is error-free, and LANDesk reserves the right to update, correct, or modify this material, including any specifications and product descriptions, at any time, without notice.

LSI-0317

Table of Contents

Executive Summary

Deploying operating systems, applications, patches and updates across an enterprise network is a complex process that requires careful planning and oversight. Successful processes take into account not only technical issues and requirements, but business needs as well. Developing a consistent, repeatable software deployment process based on industry standards and best practices is critical to ensure maximum service availability while minimizing business disruption and intrusion on the user environment.

A successful software change management strategy combines effective internal processes with clearly defined personnel roles and tools optimized for efficient management. Standards such as the IT Infrastructure Library (ITIL) can help focus your staff's activities, while task-optimized tools such as LANDesk Management Suite can enable your staff to perform their tasks more quickly. Effective tools extend your staff's power to implement and maintain core standards, resulting in improved services levels at the same time that total costs are reduced.

This white paper reviews overall best practices for a successful software change management process, with an emphasis on the use of effective tools such as LANDesk Management Suite to optimize that process.

Overview: Effective change management process

A consistent, repeatable software change management process takes both technical and business issues into account. The goal is to maintain access to software tools and services while minimizing the impact on users, organizational infrastructure and staff.

An effective software change management process consists of four major phases:

- Creating infrastructure and process foundations
- Preplanning/Change proposal
- Building and testing/training support staff
- Deploying the package/Follow-up

The following sections review these industry best practices for deploying a major application across a complex network environment.

Phase 1: Create infrastructure and process foundations

To effectively manage application deployment across your organization, you need to create both a technical infrastructure and a process infrastructure to help ensure that sound decisions are made and that all major stakeholders are involved in the process.

These infrastructure elements should be in place in your organization:

- Configuration standards/Software change process
- Technical advisory board (TAB)
- Change control board (CCB)
- Test lab
- Systems management infrastructure

By creating a consistent, repeatable process for change and release management, you will reduce problems over the long term and enable more consistent service levels throughout your organization.

Configuration standards/Software change process

To ensure efficiency and consistency across the organization, you should start from a standard, documented baseline configuration, then define strict processes for modifying that base configuration. The standards you implement will depend on your organization’s specific needs. General standards such as the IT Infrastructure Library (ITIL) provide an excellent framework for overall change control and service management, while targeted standards such as FiSMA and COBIT provide specific details as they relate to government or industry regulation.

Increasing regulatory compliance makes implementation of a recognized standard critical for successful control over your business and technical processes. The ability to show compliance with specific standards can simplify regulatory audits and focus your overall business processes.

Technical advisory board (TAB)

You should have a technical advisory board with representatives from IT, operations, network infrastructure, security and support organizations. This board brings together the core technical knowledge needed to plan and execute a successful internal software release, and should be individually focused on their own areas of expertise. The TAB will both evaluate software change requests from technical or business units, and will initiate its own change requests in response specific business, regulatory or technical needs.

Change control board (CCB)

The change control board is responsible to oversee changes to the computing environment and includes stakeholders from all major organizations in the company. This is an administration board, not a technical evaluation team; the CCB exists to ensure that the effects of changes are understood and that the interests of each department are taken into account as part of the software release.

The CCB is responsible to the business organization, and functions as the bridge between technical staff and business staff. While it may be easy for technical staff to see a CCB as an inconvenience or a roadblock, it's important to remember that the CCB is the most efficient means of gathering critical information together to ensure successful software change management, then disseminating that information back out to key stakeholders across the company. Business oversight is critical for successful change management.

Test lab

You should have access to a test lab populated with typical computers running standard configurations on which to test new software releases and deployment procedures. Ideally this is a dedicated lab running in-house, with separate machines for each approved configuration. Virtual machine technologies from companies like VMware* can enable fast, efficient test labs with a minimum of hardware and maintenance cost.

An effective test lab enables technical staff to identify and mitigate the majority of problems associated with software changes. Extended regression testing can combine with deployment process testing to eliminate most problems before you reach the production environment.

Systems management infrastructure

While not necessary for effective change management, a centralized configuration management and software deployment solution such as LANDesk® Management Suite is strongly recommended. A network-wide systems management solution can vastly simplify testing, planning and target selection as well as automating application deployment, troubleshooting and configuration maintenance.

Active software deployment and configuration management technologies can help IT staff:

- Accurately discover and inventory computers throughout the network
- Quickly select targets for distribution using hardware, software and/or directory service attributes
- Deploy update packages to selected target computers in the background for later execution
- Reduce impact on both total network bandwidth usage and individual computer resource usage using advanced bandwidth reduction technologies
- Devise and execute installer scripts containing multiple installation directives that support silent installs and enable multiple restarts to complete deployment tasks
- Ensure that deployments complete on each target computer through task completion and checkpoint restart technologies
- Gather real-time feedback on the success or failure of deployments to each individual computer
- Automatically redeploy the update to computers on which the initial deployment failed
- Automatically maintain standardized configuration on qualifying computers according to policy rules established by IT staff

By simplifying deployment planning and target selection, reducing network resource requirements, automating distribution and installation, and quickly identifying redeployment targets, an advanced systems management solution can substantially reduce the time and headache required to deploy software changes across the enterprise.

Just as importantly, an effective systems management solution can help you quickly resolve unanticipated problems after deployment is complete. No amount of planning can anticipate all possible problems or conflicts, so tools to extend your helpdesk staff's ability to resolve those problems is critical.

LANDesk Management Suite performs these standard systems management tasks, then extends their efficiency and effectiveness with advanced technologies that further decrease bandwidth usage while increasing the reliability of software deployments—without requiring substantial changes to your network and management infrastructure.

Phase 2: Preplanning/Change proposal

Before proposing a configuration change to your change control board, you need to do preliminary research and develop an initial change proposal that describes the primary technical and business impacts of the update. This is a business proposal that makes the benefits of the update clear and gives stakeholders a heads-up about possible impacts.

The preplanning/change proposal process consists of these major steps:

- Determine the need for configuration change
- Perform initial testing to more clearly identify potential impacts
- Develop a preliminary assessment of impacts and requirements
- Generate a change proposal and submit to the CCB

The following sections provide more detail on each of these steps.

Determine the need

Software changes can range from major OS updates to business productivity applications, and security patches to virus protection updates. There must be a clear business or technical benefit for a change to your core, approved configuration. By limiting casual change to your base configuration, you keep tighter control over the effects of software interaction with your baseline operating system and your critical business productivity applications.

Once you've identified the need for a software change, you need to evaluate and prioritize that change. As you perform additional testing, you may discover that the benefits of the software change fail to justify the costs involved in deploying and supporting it. Similarly, you may find it beneficial to combine several smaller patches or software releases into a single combined release that maximizes your deployment effort.

Perform initial testing

Initial testing is intended to identify major impacts to your overall environment so you can plan additional testing and mitigation efforts. Initial testing should be performed in your test lab on computers running standardized, approved configurations with all major productivity applications represented. This enables you to identify problem resolution needs to be addressed during the Test and Build phase, and helps you identify potential impacts on both network and computing resources.

At this point the goal is to understand and document the software impacts, not solve all potential problems. You will perform more complete application regression testing later in the process once your CCB has approved your initial change request.

Preliminary impact/Scope assessment

Once you've identified the major application impacts, you need to assess impacts on IT and support staff, business processes and service availability, and end-users. Where changes are made to key servers, impacts may be more substantial and require extended planning.

Impacts can include:

- Target computer selection; assess requirements
  To effectively scope the project, you need to identify how many target computers will receive the software package, where they are both physically and logically located in your network, and what minimum hardware and software requirements each target must have for successful deployment. Ideally, you should perform a new machine discovery and gather fresh inventory information to ensure that your data is current and accurate.

  This computer discovery and inventory effort will inform nearly every other preplanning task, from understanding network impacts to building a deployment schedule.
- Network resource usage
  Depending on the size and complexity of the software change package, deployment can have a substantial impact on network resource usage. For example, the installer packages for Windows XP Service Pack 2 (XPSP2) or Microsoft Office are quite large at around 300 MB each. Delivering those packages to each qualifying computer on a complex network can use a substantial percentage of available bandwidth for an extended period of time. Your technical advisory board should include a network architect who can identify bottlenecks such as WAN links, and help you develop a deployment plan that will minimize overall impact on users and network services.

  For example, delivering XPSP2 to 1,000 computers will require a minimum of 270 GB of network bandwidth using standard unicast distribution—and up to double that with error correction, resends and the need to seed distribution points on remote network segments. Some software distribution systems offer bandwidth reduction technologies such as zoning or multicasting. Understanding the dynamic between your network and your software distribution system can have a substantial impact on how you develop your deployment plan, what infrastructure is required, what network impacts will occur, and what timelines will be required for complete deployment.

- Service infrastructure impact
  If you are making changes to a server, you need to identify the impacts on the computing environment from that change. Impacts can include loss of capability as the server is temporarily taken out of service while software is installed, changed user access requirements to applications delivered by that server, and extended support requirements such as database updates. Your TAB can help you identify and mitigate these impacts.
- Management infrastructure impact
  It's important that your configuration management and software distribution infrastructure be properly set up and configured to maximize efficiencies and reduce both network and user impact. You should ensure that your infrastructure hardware and software are up to the task, and that there are sufficient resources (disk space, RAM, network bandwidth) to complete the task successfully.

  Where processes can be automated, you should take advantage of that automation to reduce management costs and overhead, and to increase overall success of the deployment. If your software distribution system requires staging servers on each network segment, you need to confirm that those servers are in place and accessible to your central management system.
- Staffing requirements
  Take into account not only staffing needs for deployment, but for testing, training and ongoing maintenance. Identify the specific skill sets you need to accomplish the task and the cost of devoting those resources to deploying the software change instead of performing other IT functions. Take into account not only initial deployment support, but the time required to chase down and resolve failed installs. Determine needs both at your primary site, and at any remote offices or satellite locations.
- User and helpdesk training
  Based on your initial testing, you should be able to identify training requirements for both end users and your internal helpdesk support staff. For massive changes to the end-user computing environment, such as an OS replacement or an application suite update, you will need to provide both support training to your helpdesk and end user training on how to access and use key application features. Training may be delivered in a lab setting or be document-based, depending on the complexity of the software change and your users' technical savvy. Identify training needs as part of your initial assessment so appropriate resources can be allocated to the task by the CCB.

  You should also formulate the beginnings of a communication plan that informs both support personnel and end users of the kinds of impacts they will experience and the levels of support they will need to provide for themselves. This communication plan will be refined as the process continues and you gather more information from your testing group and your CCB.
- Preliminary deployment plan/Timelines
  Based on your assessment of other impacts, you can now build a preliminary plan that outlines all the major activities required for the deployment and establishes an overall timeline for all phases of the deployment, including infrastructure development, testing, training, release building, automated deployment and post-deployment follow-up.

  Use this preliminary deployment plan and timeline to clearly identify prerequisites, dependencies and potential roadblocks, then build mitigation plans to deal with those issues. This preliminary deployment plan should clearly articulate all major activities and impacts for each site where the deployment will occur.

  Your deployment methodology will have a substantial effect on your timelines. If you are doing an active deployment where you directly install the software package on each target computer, you will be able to pinpoint milestone dates more easily and control overall timelines. If you are using a policy-based passive deployment, the process will probably take more time to finish and you will have less control over specific dates, but you can also expect more complete coverage with less intervention.

  For maximum efficiency and control over the software change process, you should carefully consider an active deployment to get the software quickly in place, then use a policy-based passive deployment to handle stragglers and maintain overall configuration integrity. A complete systems management solution such as LANDesk Management Suite can help you accomplish this.

Once these major impacts are identified, scoped and documented, you have all the materials needed to build a configuration change proposal.

Change proposal to CCB

The change proposal you create for your CCB is designed to communicate information to key stakeholders so they can understand why the software change is important and how it will affect them both during and after deployment, so they can plan business and departmental activities accordingly.

The change proposal should contain all the knowledge gathered during initial test and preplanning, but it will also be refined by the CCB itself as each stakeholder provides additional information or requirements. The proposal itself may need to be revised and resubmitted in light of new data. This process may repeat multiple times to create a final, generally-accepted change plan.

At the end of this process, all stakeholders should clearly understand their individual roles, responsibilities and tasks and be committed to the project. Resources should be clearly identified and budget assigned. Approval of the change proposal by the CCB means approval to dedicate all needed resource to the project with the intention of completing it as defined in the final proposal.

Phase 3: Building and testing/ training support staff

With a preliminary deployment plan and CCB approval in place, you need to build and test the application package. This may consist of simply deploying the default package as supplied by your software vendor, or creating an optimized package with extended logic supplied by your distribution system to customize application settings for different classes of users.

It's crucial that you train your helpdesk support staff on how to deal with predictable software problems. While your support staff will extend and refine their support plans based on experience, providing a strong overview of changes to the user environment will give them the tools they need to plan ongoing support activities and allocate needed resources.

The build/test/train process consists of these major steps:

- Develop a configuration methodology
- Regression test current applications
- Build and test package
- Train support staff
- Finalize deployment strategy/select targets
- Get CCB release approval

The following sections provide more detail on each of these steps.

Develop a configuration methodology

In your initial testing you should have identified core configuration requirements needed to make the software package function correctly, as well as key applications that are adversely affected by installing the new software package. You may need to deploy patches or other prerequisite application to support deployment of your new software package. It's important that you build an overall configuration methodology that can take any required

LANDesk® Targeted Multicast™ technology increases software distribution efficiency, reduces resource requirements

LANDesk Software has developed core systems management technologies designed to substantially reduce both server infrastructure and network bandwidth usage. The result is faster, more efficient software deployments that require less time, money and effort to plan and complete.

In an effort to overcome the high bandwidth usage and performance bottlenecks of traditional one-to-one unicast solutions, multicast technology was created that enables a package to be broadcast once to multiple targets on a target network segment. Unfortunately, while traditional multicast systems have been useful in re-imaging test machines in a lab setting, they have largely proven too expensive, finicky and resource intensive for day-to-day use in enterprise networking environments.

Exclusive LANDesk® Targeted Multicast™ technology makes the use of dedicated multicast and application staging servers functionally obsolete. By taking advantage of an underutilized client computer on the target network segment to function as a temporary application repository and local multicast server, LANDesk Targeted Multicast technology makes it possible to enjoy the benefits of multicast services without the dedicated hardware, router reconfigurations or network flooding typical of traditional multicast solutions.

The result is the same kind of flexibility and power associated with a highly distributed, server-based software deployment infrastructure without the dedicated hardware—or the maintenance, network bandwidth usage, latency or synchronization problems that normally goes along with it. That reduces hardware costs, deployment time, infrastructure complexity and maintenance costs, all while improving deployment times and reducing network bandwidth usage.

The savings can be dramatic, especially for large application suites or OS updates such as Windows XP Service Pack 2 (XPSP2). For example, in a sample network of 1,000 computers divided into 10 network segments, a traditional unicast solution for deploying XPSP2 will require a minimum of 270 GB of network bandwidth. If you're using application staging servers on each network segment, you need an additional 3 GB to seed those distribution points—normally over slower, more expensive WAN links—as well as the time to make that happen. With normal error correction and resends to failed targets, you can easily end up using 300 GB or more of network

continued on next page

change into account. You will identify and refine specific configuration requirements during regression testing.

If you are using an automated software distribution solution such as LANDesk® Management Suite, you can take advantage of job scripting capabilities to build extended logic into your deployment package. For example, you may be able to develop a single software package that can identify and install prerequisite software as part of the deployment process. An ability to dynamically identify and address those requirements can enable your technical staff to deploy fewer packages to support complex environments, while performing critical configuration tasks automatically.

Your initial configuration methodology should take into account all approved configurations. You can further leverage this knowledge to document the changes made to your approved core configurations in your software configuration management database as you establish new baselines.

Regression test current applications

Full regression testing will focus on two distinct areas—ensuring that software configuration settings are correctly applied at install, and ensuring that your other applications continue to function correctly after installing the new package. Regression testing is a complex discipline that exceeds the scope of this document; you should already have a regression testing methodology and completion standard in place.

The goal of regression testing is to ensure successful deployment throughout your environment. Some hardware or software may need to be updated to support the new package. Identify required updates, then develop a plan to address those issues. It may not be cost-effective to update all target computers to meet the requirements of your new software package; you will need to either replace those computers with more capable machines, reallocate those computers to other functions, or develop extended mitigation plans to work around their limitations. These decisions need to be documented as part of your overall deployment plan.

continued from previous page:

bandwidth before the job is done. To minimize impact on network services you either need to work at off-peak hours or spread the job over a very long period of time.

Using LANDesk® Management Suite with LANDesk Targeted Multicast technology, that same transfer requires as little as 5.4 GB of network bandwidth, and requires no dedicated server hardware on the subnet. Management Suite performs a standard HTTP unicast to each temporary multicast representative on each of the 10 network segments (a total of 2.7 GB of network bandwidth used—the same amount required just to seed the application staging servers in a standard unicast distribution). Targeted Multicast technology running on each of the 10 temporary multicast hosts then performs a single multicast distribution on each network segment (another 2.7 GB of network bandwidth). Assuming a high, 20% failure rate that requires unicast redistribution to each affected computer, that scenario could consume as much as 54 GB of additional network bandwidth on the fast local networks (a process that can be automated using exclusive LANDesk® Peer Download™ technology).

Standard unicast uses a minimum of 300 GB of network bandwidth and 10 dedicated staging servers.

Targeted Multicast uses between 6 GB and 60 GB of network bandwidth and requires no dedicated servers.

The savings are clear. LANDesk Management Suite with LANDesk Targeted Multicast technology can enable faster, more efficient software changes that require less time, money and effort to plan and complete.

The information that you gather during regression testing will inform all other elements of your final deployment plan and training program. It's critical that regression testing is extensive enough to identify the majority of configuration and support issues in advance. The cost of problem resolution increases substantially as problems are discovered late in the process, so it's important to perform vigorous testing as early as possible.

Build and test package

Once you've refined your configuration requirements mitigation efforts, you need to build and test the software deployment package in your test lab on computers running your standard configurations. At this point the goal is to ensure that package installs and runs correctly, and that target computers function as expected once deployment is complete.

You should also test your deployment methodology as much as possible. For example, if you are using LANDesk® Management Suite to automate package distribution, you should test both your overall deployment process and the specific deployment scripts you build. Ensure that all target computers are reached, that individual script elements such as prerequisite evaluation and install function as expected, and that feedback mechanism report accurate status data. Refine and retest as needed.

Because this is an iterative process, it is to your advantage to use an automated method for quickly restoring test machines to those standard base configurations for retest, such as re-imaging each computer or using virtual machine technology like that provided by VMware* or others. Repeat the build and test process until you can successfully install the package to all standard configurations. Identify your final build candidate.

When testing is complete, document the specific configuration changes made as part of the software deployment. You will use this information to update your configuration management database as part of your post-deployment follow-up.

Train support staff

As a result of your release team's work in defining, testing and debugging the release you will gain a substantial knowledge base about the software change package and how to support it that you need to transfer to both your support staff and your extended IT administration staff.

If you have a training organization, take advantage of their expertise to develop a training program that covers overall architecture and configuration changes, specific features and application support requirements. It's critical that this training be developed and delivered prior to deployment to ensure minimal interruption of user services and to enable rapid mitigation of any application issues that weren't discovered and addressed during testing.

If your software package changes the user interface or adds new programs, you should develop end-user training as well to ensure successful software usage. At the very least you need to provide an information sheet that directs users to sources of information; for complex OS or application updates you may need to schedule face-to-face training.

Finalize deployment strategy/select targets

As part of your preplanning you developed an initial deployment plan that took into account your network architecture, regional requirements, configuration management infrastructure and overall timelines. With your release package built, you need to refine that plan to take into account any additional knowledge gained during testing. For example, if you discovered application patch needs or configuration requirements, you need to address those needs as part of your overall deployment plan. Your finalized deployment plan should include not only timelines and methodologies, but should clearly articulate completion milestones that define when the task is complete.

You should also select your targets in advance to ensure complete coverage and to help you assign deployment support resources. Access to accurate and up-to-date machine inventory data is critical to ensure that both hardware and software requirements are met prior to deployment, and to enable accurate target selection.

If you are using an automated software distribution solution such as LANDesk® Management Suite, you can easily create queries that leverage both

LANDesk® Peer Download™ technology increases package availability and enables policy-based management

One of the banes of a system administrator's life is dealing with the computer that was turned off during a major deployment, or the laptop that was on the road. Traditionally, IT has been forced to deal with these stragglers on an individual basis, and chasing them down often requires more time and effort than the initial distribution.

This problem can become even more pronounced as administrators attempt to maintain application policies that control which users have access to which applications, and which patches are installed on which computers. Dozens of configurations in hundreds of permutations can make keep IT staff constantly running to maintain consistent, standardized configurations.

Exclusive LANDesk® Peer Download™ technology works in conjunction with LANDesk® Management Suite's application policy management features and LANDesk® Targeted Multicast™ technology to make software packages directly available to clients without requiring dedicated package depots or staging servers on the local network segment.

When a package is deployed to a target computer using LANDesk Management Suite, the original package is temporarily stored in a distribution cache on that computer before it's executed; administrators can choose how long a package file is cached before it's automatically deleted—typically 45-90 days. LANDesk Peer Download technology takes advantage of that cache to enable managed computers on the same network segment to quickly download the package from a peer computer's cache rather than triggering a new deployment event at the central package repository. The only impact on the source computer is a momentary file copy; traffic stays on the fast local network to minimize overall impact.

For example, if a computer was turned off or a laptop was on the road during package deployment, when the computer next starts up on the network it can automatically search for another computer on its own network segment that has already received the package, and pull the package down from that computer's cache. The software is automatically installed without further administrator intervention, and there's no need to pull the package over the WAN from the central package repository.

This provides the same kinds of functionality normally associated with a package depot—including policy-based management and nearest server access—but without requiring the dedicated server or the latency and package synchronization problems that come with it. By distributing load across computers on the local network segment you not only enable massive package availability, but you eliminate the dedicated hardware/software infrastructure normally required to support it. This enables painless scalability and complex policy-based management with minimal cost, time, maintenance and effort.

inventory and directory services data and enable you to group target computers not only by OS requirements, but by location, department business unit, approving manager or any other supported organizational unit. If you are performing a staged deployment, you will want to develop separate target queries for each distribution phase. LANDesk Management Suite can further leverage this target list to enable “seed and activate” deployment staging. See the section “Increase Speed and Efficiency with LANDesk® Management Suite” later in this paper for more information.

With a passive, policy-based deployment you still need to clearly identify target computers so you can develop a methodology that clearly identifies when the deployment is complete. An effective inventory management system is critical to completion analysis and reporting.

CCB release approval

You already received preliminary approval from your CCB before you began the test/build phase. Once you have built the release package and refined your deployment strategy, you need to obtain approval from the CCB prior to deployment. This approval meeting helps ensure that all prerequisite actions and training have been performed, and ensures that stakeholders are informed and ready to provide needed support. It also provides a final check that technical, service and business issues have been successfully addressed in the release itself.

The result of the CCB release approval is a go/no-go decision. The CCB can choose to postpone or cancel the product release based on your final readiness report, so it’s important to make sure that you have addressed the key impacts identified during initial planning, created mitigation plans and clearly described how you will maintain service levels throughout the deployment. Approval by the CCB means that you can begin deploying your software change package immediately.

Phase 4: Deploy and follow-up

Once you have approval and you’ve communicated your deployment schedule, use your systems management solution or software distribution process to deploy the software package to clients. Even the most effective distribution system cannot guarantee successful deployment to all targets, so you will need to follow up on failed deployments and develop strategies for dealing with new clients in your environment. Your new standard configuration should be documented and implemented in your configuration management database to ensure consistent performance and service levels as you go forward.

The deployment and follow-up process consists of these major steps:

- Deploy the package
- Identify failed deployment/create chase list
- Document new OS configuration policy
- Post-deployment review

The following sections provide more detail on each of these steps.

Deploy the package

Using the methodologies and timelines identified in your release plan, deploy the package to target users. You should have already defined your release schedule, personnel requirements and supporting tools. If you are using a systems management solution such as LANDesk® Management Suite, deployment should be highly automated and complete with little intervention from technical staff.

You should also document any problems or challenges you experience during deployment to help you refine your process for future releases. This enables you to create your own list of best known practices that are specifically optimized to your unique computing environment. You will communicate this information as part of your post-deployment review.

Create chase list and maintenance policies

Depending on the tools and methodologies you use to deploy your software change package, you can expect a failure rate of between 5% and 20%. If you are using a software distribution solution such as LANDesk® Management Suite, the system can be configured to use technologies such as Wake-On-LAN and task completion to automatically redeploy to failed targets.

If your systems management solution supports policy-based configuration management, you should create and implement application policies that automatically deploy your software change to qualifying targets. These policies can function both as your overall deployment method and as a supplement to a direct-deployment strategy. Automated policy management increases your chances of overall success, and is a key element of your ongoing maintenance plan.

As part of your deployment plan you should have identified a deployment window that takes these automated measures into account and identifies when direct intervention will be required to ensure successful deployment. A certain number of deployments will fail regardless of automation or other systemic measures and you will need to create a "chase list" of computers to follow up on.

The reasons for deployment failure can include hardware or network failure, computers disconnected from the network or out of the office, and unanticipated software crashes. If you are using an automated software distribution system such as LANDesk Management Suite, you should be able to quickly gather deployment status information. Management Suite features real-time status data to help you quickly build your chase list, as well as remote problem resolution tools to help you identify and resolve the issues that caused the initial deployment to fail. In some cases you may need to manually deploy the software package directly onto the computer from CD.

Document new OS configuration policy

In addition to technology-enabled automatic configuration management policies, you should maintain a configuration management database that clearly defines and documents your standardized, approved OS and application configurations so you can maintain configuration integrity and ensure ongoing service levels.

You should already have documented planned configuration changes as part of your deployment plan. During deployment you may have identified additional support files or configuration changes needed to ensure successful completion; update your configuration management database to reflect these changes.

In addition to documenting the configuration changes and methodologies associated with the software change, you should maintain a library of the installers, software files and configuration files needed to complete the update. These files will form the basis of disk images and provide a comprehensive archive that will enable you to maintain consistent configurations throughout the enterprise, as well an enabling you to gracefully return computers to a pre-install state if needed.

Using LANDesk® Patch Manager to automate patch deployment

LANDesk® Patch Manager is an automated vulnerability assessment and patch management solution that integrates seamlessly with LANDesk Management Suite. Using a central vulnerability database, Patch Manager can quickly identify application or OS patch needs and automatically distribute the correct patch. You can even define your own custom vulnerability definitions and create patches to further customize target computers based on the applications or OS currently installed.

This ability to automatically assess and update every computer in your environment can make it easier to bring computers up to standard prior to a major software deployment, simplifying your deployment plan and enabling IT staff to take control of baseline configurations across the enterprise.

Post-deployment review

As part of your deployment plan you should have defined completion criteria. Once those goals have been met your technical advisory board should meet to report new knowledge gained during deployment and use that data to refine your ongoing deployment processes. This information should be collected and presented to the CCB along with recommendations for future enhancement to the process.

This post-deployment report to the CCB enables technical staff to define a specific ending point for the software deployment and to clearly transition to support activities. By keeping key stakeholders in the information loop, you ensure that expectations are set correctly and that progress is both recognized and effectively documented.

Increase speed and efficiency with LANDesk® Management Suite

This paper describes an overall process and set of best practices that have been tested and proven at leading enterprises around the world. These processes can be implemented by any organization using any set of technological support tools, though you will realize significant efficiency boosts if you supplement these practices with some kind of automated software distribution and configuration management solution.

LANDesk® Management Suite enables superior planning, deployment and reporting efficiency at a low cost and with minimal impact on network, infrastructure and personnel resources. An easy-to-use interface combines with exclusive technological innovations to streamline information-gathering, planning, deployment and maintenance activities. The result is greater capability combined with faster results—at a substantially reduced cost over other solutions.

Improved information-gathering and reporting

LANDesk® Management Suite gives you on-time access to accurate, useful information so you can make better decisions for your unique environment.

Extensive device auto-discovery helps you understand what computing assets you have and where they're located. Detailed inventory scans tell you about the hardware and software configuration of each managed computer. A unified database provides centralized access to computer information, and a powerful query builder gives you the ability to quickly sift that data to find the right information for the current task—whether the data comes from computer inventory or your directory service infrastructure.

Real-time status information keeps you informed of deployment status so you can speed tasks to completion, while detailed business intelligence reports enable accountability and speed both planning and status reporting. The result is faster, easier access to the data you need to get the job done.

Increased process automation

Leveraging accurate information to automate management processes can increase your staff's ability to meet service levels and business needs while tightly controlling costs.

A task scheduler combines with easy database queries to automatically deploy software packages to the right targets. Advanced task scripting enables fully automated software installs while minimizing manual intervention. Task completion and policy-based management tools help make sure the job gets done, while application healing automatically repairs damaged installs.

Simplified management infrastructure

LANDesk Software has developed core technologies that not only improve overall IT control, but do so with a minimum of impact on your hardware infrastructure.

Traditional systems management solutions start with a centralized cluster of servers that are connected to support servers located on each network segment. LANDesk® Management Suite implements exclusive core technologies to reduce or eliminate the need for dedicated management servers on remote networks. LANDesk® Targeted Multicast™ and PXE Proxy technologies can automatically select a managed computer on the remote network to function as a temporary service center that performs management tasks—without requiring either dedicated hardware or manual device configuration.

The result is more efficient software distribution and OS deployment, without a dedicated hardware infrastructure. As your needs scale, this automated infrastructure scales with it—automatically. That simplifies infrastructure planning and maintenance, and reduces overall hardware and personnel costs.

Example: Seed-and-activate deployment

LANDesk® Management Suite can substantially reduce the time required to deploy a software change, providing better service levels to your end users while minimizing impact on IT staff, network resource usage and hardware infrastructure.

Automated target selection combined with exclusive LANDesk® Targeted Multicast™ technology makes it possible to distribute a software package to the deployment cache on LANDesk-managed target computers without actually executing it. This ability to seed target computers means you can deploy the package as soon as it passes your internal test requirements. Because Targeted Multicast uses as little as 80% of the bandwidth typical of standard unicast distributions, you can use it in conjunction with bandwidth throttling to seed targets with the software installer package in the background while completing your training and CCB processes.

When you receive CCB approval, you can deploy a simple activation script to run the installer package on your targets. You can begin installing the software change package within minutes of CCB approval because you've already done the heavy-lifting of the distribution in the background. This ability to seed-and-activate enables you to perform multiple management tasks simultaneously, reducing the total time required and resulting in faster overall deployment with less impact on both users and network resources. For critical security patches, this reduced time requirement can make the difference between deflecting a malicious attack, and becoming its next victim.

Conclusion

Deploying software across complex enterprise networks requires extensive planning and discipline to ensure that both technical and business requirements are successfully met. A consistent, repeatable software change management process is critical to ensure compliance with configuration standards, to minimize impact on business processes and to maintain service availability to your users.

The following deployment model can be applied to any software package, from a simple patch to a major OS or application rollout.

- Create infrastructure and process foundations
  - Configuration standards and change process
  - Technical advisory board
  - Change control board
  - Test lab
  - Systems management infrastructure
- Preplanning/Change proposal
  - Determine the need for configuration change
  - Perform initial testing to more clearly identify potential impacts
  - Develop a preliminary assessment of impacts and requirements
  - Generate a change proposal and submit to the CCB
- Build and test/Train support staff
  - Develop configuration methodology
  - Regression test current applications
  - Build and test package
  - Train support staff
  - Finalize deployment strategy/select targets
  - Get CCB release approval
- Deploy and follow-up
  - Deploy the package
  - Identify failed deployment/create chase list
  - Document new OS configuration policy
  - Post-deployment review

As with any general model, this process should be refined and adapted to the specific needs of your business and network needs. Service management standards such as the IT Infrastructure Library (ITIL) can provide more detail on individual steps.

To further maximize your resources, reduce costs and increase overall capability you should supplement these processes with an automated systems management solution such as LANDesk Management Suite.