

Process Foundations for Successful Solution Deployment

"How To" Article

NOVELL APPNOTES

Sterling Smith
Consultant
Novell Consulting
swsmith@novell.com

This AppNote discusses general procedures that can be implemented in most organizations to make any technical implementation of any solution more successful. It is intended for technical/project managers and specialists who must manage and implement the software testing process prior to implementation.

Contents:

- Introduction
- Discovering Information Through Testing
- Defining the Test Environment
- Conclusion

Topics	consulting methodologies, implementation guidelines
Products	n/a
Audience	network designers, consultants, integrators
Level	beginning
Prerequisite Skills	none
Operating System	n/a
Tools	n/a
Sample Code	no

Introduction

Suppose management has asked you to research solutions as they relate to cross-platform services that will integrate the network yet ease management. After some research, you conclude that the answer is Novell's Net Services software in the form of NDS eDirectory and Net Management solutions. You sell the company on the idea, budgeting is approved, and now it's time to deploy the solution. That's a big task, involving many risks. A lot must happen between the conception of an idea and its implementation. And it's usually the "little things"—such as project management, testing, and regression methodologies—that kill technology rollouts.

This AppNote discusses general procedures that can be implemented in most organizations to make any technical implementation of any solution more successful. It is intended for technical/project managers and specialists who must manage and implement the software testing process prior to implementation.

The information presented in this AppNote should supplement any existing methodologies that a company requires, as they pertain to the acceptance of solutions into the environment. Although this document may contain a fair amount of business jargon, it is hoped that the reader will gain an overall understanding and appreciation of the internal processes that impact solution deployment in any organization. Other aims are to inspire technologists to become more business-centric in their daily routines, and to suggest areas that can improve the success of technology solution rollouts in the future.

In any complex distributed client-server system, there are both server and client activities that need to be completed before the system is brought into production. Proper planning will help provide the foundation for a seamless turnover and will improve operational service levels during the lifecycle of the solution.

(Remember the six Ps: Proper Prior Planning Prevents Poor Performance.)

The main objective of production acceptance procedures is to ensure that the manageability, operability, and infrastructure requirements of new solutions, systems, or hardware/software rollouts are addressed as part of the planning, testing, development, and deployment phases.

Note: References to "solutions" in this document refer to any network services software or end-user application, client or server. Thus, upgrading a DHCP server and installing a new version of Office 2000 can both be considered solutions for the purposes of this AppNote.

Commitment to Enterprise Architecture

“Enterprise Architecture” is a model that an organization can use to justify both existing and future systems by providing a foundation for the implementation of business solutions using technology. Not only do different components provide features and functionality of their own, these components can be logically combined to create an overall system that is greater than the sum of its parts. This overall effect is the desired future state of an organization’s enterprise architecture, which can be aided by two things: a commitment to technology that solves a business need, and the proper implementation of that technology. The methodologies outlined in this AppNote should help you define an effective enterprise architecture.

The Big Picture: The Process

As with any new venture, an organization should go through the process shown in Figure 1 to ascertain how appropriate a proposed solution is for solving business needs.

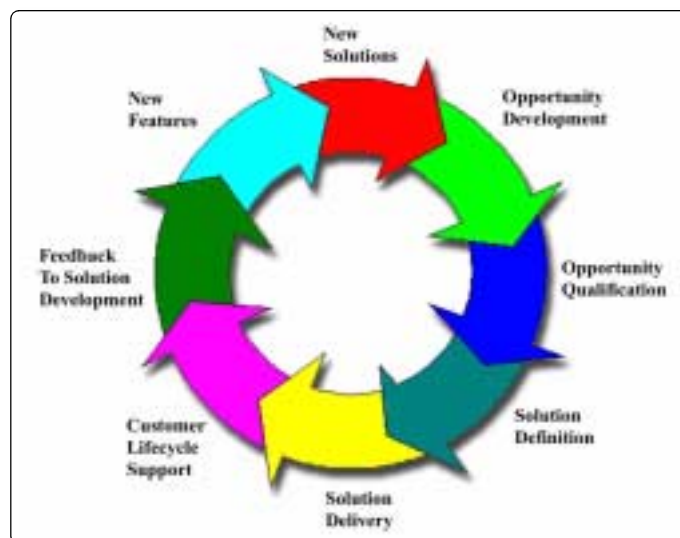


Figure 1: *Business solution process.*

The phases in this process are described below.

- *Identify new solutions.* The company recognizes the need for a solution (software or otherwise) that can aid in recognizing a return for the business. At this stage, recognition that a new solution is required suffices.
- *Develop opportunities.* Members within a team brainstorm to discuss the requirements for the solution and research possible alternatives. At this stage, data is gathered with regards to cost and requirements for each possible opportunity.

- *Qualify opportunities.* After the research has been completed and the suggestions have been made, it is time to sell the proposed solution to the decision makers. If a decision is made to continue with the proposal, the process continues.
- *Define the solution based on gathered data.* The details of the specific solution are not necessarily divulged at the opportunity qualification stage, because (1) decision makers are most interested in the business benefits and the return on investment to the shareholders; and (2) decision makers do not necessarily care about the technical detail—that's the job of the technical professionals.
- *Deliver the solution.* This phase comprises all of the major tasks required to deliver the solution. At this stage, a project is formed, members are formed, work breakdown structures are determined, and accountabilities are set.

Project management is a world unto itself, and is a critical factor in the success of any project. For more information on project management and certification, visit the Project Management Institute's Web site at <http://pmi.org>.

The solution delivery phase comprises the following steps:

- *Design and build the solution.* At this stage, specialized members of the project team develop specifications and requirements for the solution in a more formal fashion. They then build the solution for testing purposes.
- *Test and document the solution.* All solutions must undergo some degree of testing for appropriateness of task and to ensure business continuity. The remaining sections of this AppNote discuss this in more detail.
- *Train the trainers.* Prior to deployment, the ongoing needs of the users of the solution must be understood and those who will train and/or support the solution must be trained.
- *Deploy the solution.* At this stage, the solution is implemented according to the requirements and as documented during the testing process.
- *Customer support.* After the solution is deployed, keep updated notes on any outstanding issues and supplement this material as part of an ongoing process to ensure customer satisfaction with the solution.
- *Feedback.* After a solution has been implemented, always provide a way to request feedback to help you understand areas of excellence and areas that need additional work. User suggestions could become new features that can be implemented into future solutions.
- *New features.* Hopefully, with enough feedback, the solution can be improved upon and expanded to greater returns and business benefits.

The Importance of Testing

It is vital to understand why solutions should be tested; otherwise, production systems may lose performance, reliability, manageability, and up-time. Here are some of the reasons why testing of any software component is very important:

- It is imperative that technical personnel understand the underlying design and implementation of a given product.
- Better understanding of a product allows more options in configuring, optimizing, supporting, and utilizing the product. This translates into more immediate and cost-effective business benefits.
- Testing can uncover possible issues in the proposed production environment, allowing the opportunity to address the issues before they become mission-critical. Additionally, finding architectural limitations in the tested product may save time and money that would otherwise be wasted in implementing an inappropriate solution.
- Testing creates an inquisitive and explorative environment in which creativity may allow additional solutions to be uncovered. In addition, it is easier to run “what-if” scenarios and adjust as necessary before the project is implemented.
- Testing makes good business sense. The company depends on professionals to make good business decisions backed by unemotional, quantitative data. Testing allows advocates to put to rest any technical issues while being able to justify the adoption of the product to key decision makers.

Discovering Information Through Testing

By asking questions and including participants who will use the solution when it is put into production, solution testers will have the understanding necessary to test the solution effectively. Those testing solutions must:

- Understand the environmental factors in which the solution is being executed
- Understand the design considerations and potential impact of the solution being tested
- Understand the expectations for the solution from the perspective of users and technical staff

Understanding the Environment

Successful testing of a solution involves understanding the environment in which the solution is to be executed. Environmental factors usually include the political and systematic processes that are established within a company’s structure.

Factors to consider include:

- *Who are the destined users of the solution?* Identifying the destined users of the solution is important. When the solution- testing matrix is developed later in the process, feedback from the users who will utilize the solution in the production environment will determine whether the user requirements for the solution have been met.
- *Who should be considered potential test users?* Since a solution could potentially have different audiences and different uses, one should select a valid statistical sample of the user population for purposes of receiving feedback when testing a proposed solution.
- *What network resources will be utilized if the solution is deployed?* An understanding of the infrastructure resources that will be required to successfully utilize the solution in production is critical for successful production deployment. It is also useful for designing the associated test processes. If these are designed incorrectly, technical issues may surface during deployment and user expectations may not be met.
- *Who should be included in the decision-making process?* Necessary team members should be selected at the beginning of an important project, especially if the scope of the project is clearly visible to the organization. Clear and open communication will help keep the project going smoothly and foster an open and cooperative working environment.
- *Have all company procedures and requirements been met?* Although solution testing is just one process, it is important because it allows the next logical step to occur: the production turnover process. All company requirements and procedures should be met before this important process occurs. Usually this includes quantitative data including justification, topology, architecture, installation and customization, and support and operations documentation.

Understanding Design Considerations

Successful testing includes understanding the nature of the solution and its potential impact on company network resources and user productivity. The information gathered about the design considerations of the solution directly impact the testing design and procedures.

Important design considerations and potential impacts include:

- *Software and hardware requirements.* The solution's software and hardware requirements must be understood. One of the biggest mistakes in testing and deploying a solution is under- estimating these requirements. This usually stems from not understanding how the solution will be used in the production environment. For example, to install an Oracle database on a NetWare file server, all of the following variables must be captured to calculate server

memory requirements: record size, number of records, number of concurrent users, and performance expectations. Failure to fully understand the solution's hardware and software requirements will give skewed results.

- *Solution interdependencies and third-party requirements.* A challenging yet important consideration is understanding the solution's interdependencies with other products. A solution regression matrix is a listing of all products that depend on this product for proper execution. A complete solution regression matrix allows more thorough testing and helps incompatibilities to be understood long before the solution is put into production.
- *Solution performance and optimization.* Important considerations that can have serious impacts on both network performance and user expectations include understanding the ramifications of a solution's architecture and design on performance. Optimization techniques need to be understood and documented during the testing phase.

In such optimization efforts, consider the following:

- *Location of the software solution.* Installing the solution locally increase network performance, but increases management and support costs (unless a product such as Novell's ZENworks is installed). Installing solutions to be shared on a network volume will decrease network performance overall, but allow for a more managed installation. Having standards for locations of management and application software, for example, will help eliminate confusion in resolving future issues.
- *Unique aspects of the solution.* Incomplete assessments of software usually produce unexpected results. For instance, Office 2000 now supports automatic download and repair from a shared source through the Windows Installer. Variables such as these must be calculated when performance measures are taken.
- *Documentation of installation and customization procedures.* Ill-fated documentation on installation and customization can meet with haphazard performance metrics. Proper documentation of installation and customization ensure that testing and production configurations can be reproduced for further examination or enhancements.
- *WAN link requirements.* The impact of wide area network links must be understood as a potential impact on performance. Furthermore, test processes must be designed to account for performance over WAN links.
- *Risk assessment.* Risk assessment is defined as a continuous process to identify and measure specific security-related risks, and to apply and monitor responsive solutions. Risk assessments identify company assets, their value, and the vulnerabilities of these assets. From these findings, you can establish

appropriate levels of security for each asset or group of assets. Solutions must be analyzed not only as assets that may need security, but also as vulnerabilities for other assets so that they may be properly protected.

Understanding User and Technical Staff Expectations

Determining the expectations of the users of the solution and establishing technical expectations are part of the overall solution testing process. Both technical staff and users have a goal in mind for the solution, although their goals and expectations can be expressed in different ways.

Users will express their goal for the solution as, “I want this thing to work so I can do my job.” This statement needs to be refined into specific tasks and processes that the solution is expected to perform. Once the specific tasks and processes are defined, the technical requirements should be appended and subsequently tested.

Test users should be positioned to test the solution’s functionality against expectations from a user perspective. The technical team may need to take those expectations and translate them into technical requirements. Expectations should also be added to the list of technical requirements, and subsequently documented.

The technical staff will express its requirement as, “I want the solution to perform optimally.” Again, what “optimally” is must be defined. To reduce help-desk calls, the technical staff expects the solution to have a certain response time and easily-understood distress signals. These expectations may be achieved by properly designing the solution based on interoperable standards and by setting reasonable expectations for both technical staff and users.

Defining the Test Environment

When the information discovery phase of testing the solution is complete, designing the test processes for the solution can begin. Designing the test process includes the following tasks:

- Building a test lab environment
- Building the test components
- Building the test design
- Testing the solution
- Drawing conclusions
- Maintaining documentation

Building a Test Lab Environment

A lab should be set up to install, configure, and test the solution for the company's network environment. The hardware and software should be representative of the existing network environment. Try to use similar network boards, LAN topology, and workstation operating systems. In addition, make sure the hardware acquired fits the requirements of the solution being tested. Failure to do so may result in inaccurate results and incomplete testing analysis.

The lab environment should not affect the operation of the existing network. However, you should maintain a connection to the current network backbone for migration and backward compatibility testing.

This lab should be divided into three logical networks:

- *Development/Test (Tier 1)*. This environment should be able to be freely created and destroyed using any new software component which needs to be fundamentally tested for obvious effects. General beta and untested products are tested here, as well as production code that is being designed for the first time. A change control is not desired at this level. (A *change control* is a formal process of communicating to levels of the company about the technical work that is being performed and its potential impact on business operations.)
- *Model Office (Tier 2)*. This environment should closely resemble the production environment. The Development/Test environment should be used to understand the issues and document for a Model Office installation, where additional research and development can take place. A change control here is up to the discretion of the company.
- *Production (Tier 3)*. By the time products go through this tier, all testing and documentation should have taken place. Issues should be minimized due to the rigorous testing from the previous tiers. Here, a change control needs to be issued before changes can be made.

At this phase, you should document procedures regarding hardware and software utilized, along with automated tools and scripts. The data collected should represent the goals of the testing process. It can also be used to generate a report showing the test methodologies and results of exercising the new system in a manner that will be very close to how the system will be used. Environmental factors found in the present production environment should be simulated so that the IT organization can predict how the system will perform.

Building the Test Components

Once the hardware has been gathered, it must be tailored to fit the testing process.

Network Clients. Using the software requirements from the discovery phase, build as many clients as necessary to complete the testing process. If there are client standards, a disk -imaging program could aid in the process. (Novell's ZENworks for Desktops 3 has this capability. See <http://www.novell.com> for more information.)

Clients will require:

- Operating systems such as Windows 95/98/NT/2000, UNIX, or DOS
- Network connectivity software such as the Novell Client for Windows 95/98, the Novell Client for Windows NT, the Microsoft Client for Novell Networks, or others
- Third-party support components such as those identified during the information discovery phase

Network Servers. Using the requirements for the solution, build the servers necessary to perform the solution testing. Any procedures required for the solution to perform optimally should be documented and are subject to Certification Testing Verification.

Server operating systems may include:

- Novell NetWare
- Windows NT/2000
- UNIX

Routers and Connectivity Devices. If the testing scenario warrants, install and configure any routers and connectivity devices. For maximum results from the testing process, the configuration should mimic production to the fullest extent possible. If extensive connectivity devices are used for testing, they should be clearly documented in a topology diagram.

Connectivity devices include:

- Routers
- Hubs, bridges, repeaters, and switches
- CSU/DSUs
- Simulated WAN links through dedicated cables

Building the Test Design

Information that must be included in the test design includes the following:

- Test matrix
- Solution regression matrix

- Automation tools for testing

Test Matrix. The test matrix should list the goals and expectations of the solution, as indicated in the discovery phase. For each goal, list the supported test tasks that will be used as a measure of success against the desired outcomes. The example below shows a test matrix for the imaging component of ZENworks for Desktops 3.

Component: Imaging for ZENworks for Desktops 3
Goal: Perform client imaging

Task	Date/Successful?
Create a Linux Boot Diskette	
Load the Image Proxy Service	
Place the image file on the proxy	
Create the Linux partition on the client	
Restore the Linux partition	
Utilize Server Imaging Policy to deploy image	
Utilize Workstation Imaging Policy to deploy image	

Solution Regression Matrix. The solution regression matrix, which was espoused in the previous section, should now be formally created. This matrix documents which applications are dependent on other applications and components, and which should be tested. Although many people indicate that time does not usually allow for such rigorous testing, this important step will document which other components should be targeted during troubleshooting sessions and for minor updates to the solution after deployment. For instance, if documenting additional applications that need regression testing with ZENworks for Desktops Imaging, one should include TCP/IP and DHCP.

Automation Tools for Testing. If required, any tools that need to be acquired and configured for use in the test scenario should be documented on the test matrix. These tools may include:

- *Network traffic generators.* Under most circumstances, the solution should be tested under heavy stress and load conditions. This will ensure that the product not only scales well, but will also most likely represent the production environment.

- *Scripts that perform repetitious tasks.* Scripts that automate tasks can be used to generate traffic and workload on a specific solution component or process. By using scripts to focus on these areas, possible architectural limits or design issues will be uncovered.

Testing the Solution

At this phase of the testing process, data requirements set forth in the design phase should be gathered during each test's iteration. The data collected should be used to generate a report showing the test methodologies and the results of exercising the new system in a manner that will be very close to how the solution will be used.

Environmental factors found in the present production environment should be simulated so that the IT organization can predict how the system will perform.

Test users who represent the users' expectations for the solution should be utilized during this phase. If required, feedback should be solicited.

Drawing Conclusions

At this phase, you should give recommendations regarding the solution testing, backed with quantitative data. Questions to consider when evaluating the findings to determine a conclusion include:

- Did the testing reveal any technical weaknesses?
- Did the testing meet technical expectations?
- Did the testing meet user expectations?
- Is the solution ready for deployment?

If it is concluded that the solution is not ready for deployment, provide information as to why the solution cannot be used in production. If testing is not successful—which in itself would prohibit the use of the solution, the information should detail what is necessary for a successful test to occur, or whether successful testing is even possible.

If the conclusion is that the testing was successful and deployment is possible, compile the information gathered to serve as a complete “certification testing” document to support future tests of the solution.

Maintaining Documentation

You must document and maintain architectural and engineering design guidelines during the lifecycle of the system. In addition to general guidelines, installation and management documentation play a key role to ongoing success of deploying a solution. If the tested solution is promoted to production, it is important to adhere to company standards in supplying both timely and accurate information.

Expend the required effort to maintain up-to-date documentation so that problems within the production environment can be more easily identified and resolved. Be sure to adhere to change control policies, as these create an ongoing record of the solution in the environment.

In addition to general guidelines, the following documents play a pivotal role in providing productive working environments when deploying technology and creating standards and guidelines:

- *How to Install, Customize, Initialize, and Shut Down the Solution.* This document provides detailed information that indicates what is required to install and customize the solution for deployment within the company. In addition, procedures to launch the solution should be given, with any dependencies noted.
- *How to Manage the Solution.* In this document, specific information regarding the tools that are used to manage the solution should be indicated. If necessary, management tasks that are considered important to the company should be detailed.
- *How to Troubleshoot the Solution.* Now that the solution is working, how should common issues relating to the product be handled? This document should answer questions about trouble-shooting, common error conditions, and available vendor support.

Conclusion

Hopefully, this AppNote has spurred some thinking about the wider scope and implications of deploying solutions into an enterprise architecture. Taking the time beforehand to fine tune internal processes will result in a more proactive environment that will yield higher reliability, satisfied users, and happy shareholders who have realized a return on their investment.

Copyright © 2001 by Novell, Inc. All rights reserved.
No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Novell.

All product names mentioned are trademarks of their respective companies or distributors.