HEWLETT®
PACKARD

Patching Usage Models

White Paper

Table of Contents

1. Introduction

This paper focuses on different approaches to patching. Depending on the task involved, there are different ways to patch systems. A reactive support situation, for example, should be handled differently than a new system installation. This paper examines the most common patching situations. For each situation, it makes recommendations on how to patch.

A usage model can be though of as a template. It is a framework around which tools and processes can be developed. Each usage model provides guidelines for achieving the intended goal.

The process of patching is further divided into three elements: tools, delivery, and IT processes. The information presented in this white paper can be used to develop a comprehensive set of patch management processes.

Scope

The recommendations given in this document are targeted for the HP-UX operating system. However, the concepts apply to all operating systems. The goal is to present a task-oriented approach to patching and systems maintenance. The techniques described can be extended to other areas of data center operations, as well.

Intended Audience

This paper is primarily intended for systems administrators and IT process planners. It includes general background information on patching along with process flowcharts for different modes of patching. The information should be useful to anyone who must maintain systems in a data center environment or who develops IT processes.

2. What is a Usage Model?

Why Patch at All?

Patches are most often associated with defect fixes. But that is not their only purpose. In addition to fixing problems, patches can be used to:

- deliver new or enhanced functionality
- enable new hardware and software
- provide useful utilities

In terms of defect fixes, patches can repair problems and restore systems to normal operation. Proactively, patches can be used to avoid downtime due to known problems. As a result, most operating environments include a combination of a base operating system and patches.

For the HP-UX operating system, Hewlett-Packard releases a core operating system and provides updates over time via patches. Among these patches are defect fixes, performance enhancements, new hardware and feature enablement. Without patches, the only way to provide changes would be to re-release the operating system. Given the dynamic nature of modern operating systems and the time involved in a product release, this is not a practical approach.

Patches are a vital part of systems support and maintenance. Since every patch is a change to the operating environment, and because change introduces risk, data center managers need to develop processes for managing patches. An understanding of the various ways in which patches are used is an important part of system operations.

Why Develop Usage Models?

There is no one "right way to patch". Rather, the approach needs to be tailored to the situation. For example, in a reactive support situation that requires a patch, modifications should be limited to the smallest change necessary that solves the problem. The engineer involved needs diagnostic tools and a means of retrieving individual patches. By contrast, a person performing a new system installation wants consistency and reliability. For them, a standard bundle of patches may be a better solution.

Patch usage models provide a basis for process standardization. They help people involved with patching -- from help desk agents to systems administrators to IS managers -- understand how patching works from end to end. And they serve to reinforce good system management practices

3. Patch Usage Models

This section presents usage models for different modes of patching. While this list is not intended to be comprehensive, it does cover the most common reasons for applying patches. The usage models presented include:

- New system installation
- Proactive patching
- Reactive patching
- Configuration change
- Operating system version change
- Independent Software/Hardware Vendor (ISV/IHV) qualification

Each of these processes is mapped out and discussed. In some cases, the use model refers to portions of other models. This fact reinforces the idea of common process building blocks. In addition, individual process steps or groups of steps are identified as requiring one or more of the following:

- **Tools**: A utility or application to help perform the step
- **Delivery**: A channel for obtaining needed patches
- **IT Processes**: The formal rules that govern the performance of a task

These common elements work together in meeting an organization's patching needs.

The materials in this section are intended for reference. Each usage model is presented in three parts:

- **Description**: a short overview of the usage model
- **Table**: listing the audience, start and end points, benefits, and elements mentioned above
- **Diagram**: a flowchart showing how the process proceeds from beginning to end

New System Installation

Description

The first usage model covers the installation of a new computer system. The features of this usage model are listed in Table 1. It is depicted graphically in Figure 1. As the diagram indicates, the process begins when a system order is planned, and it ends when the system is ready for its intended role.

As stated earlier, major versions of the base HP-UX operating system (e.g., 10.20, 11.00) are released, and the core O/S then remains unchanged. Modifications are made over time, for things like new hardware enablement and defect fixes, through the use of patches.

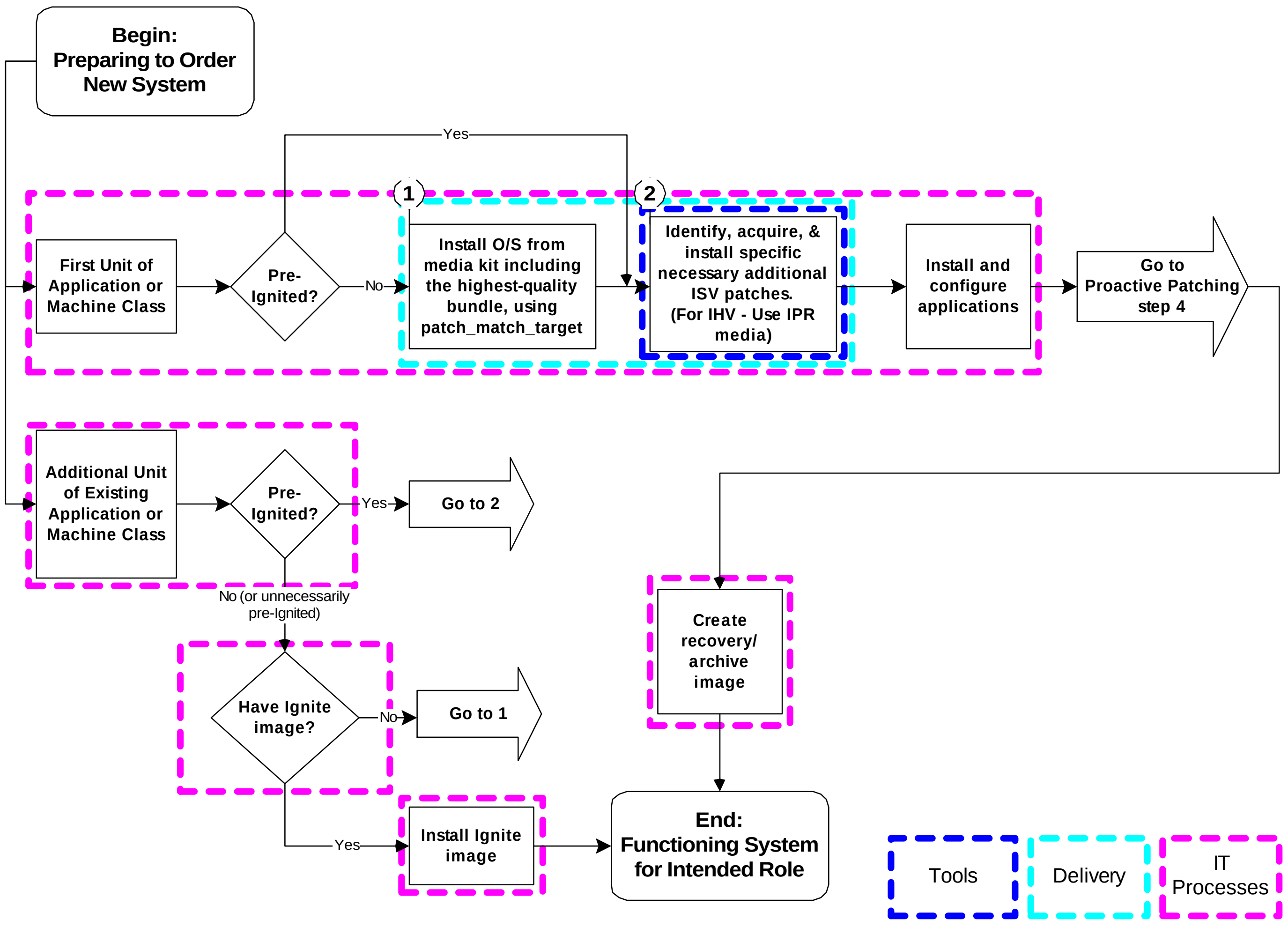
Walking through the flowchart, there are two main branches for a new system installation. The first is for systems that are the first of their kind. In this situation, there will probably not be a template upon which to base the configuration. In the second branch, the system is an incremental addition to an existing application class. Ideally in the latter case, a system "golden image" has been created using Ignite-UX, and this image can simply be applied to the new system.

Looking at the Tools, Delivery, and IT Processes blocks, it can be seen that a tool will be required to identify necessary patches. In addition, a means of patch delivery is required both for installing the base O/S and for installing addition patches. Finally, all aspects of the process need to be governed by established IT processes. Without these, system set-up is an ad hoc process that invites problems.

Table 1: New System Installation

Audience	Systems administrators and integrators	
Starting Point	Preparing to order a new system	
Completion Criteria	System is functional for its intended role	
Benefits	♦ Stable, complete operating system and utilities ♦ Timeliness: quick, repeatable ♦ Highest quality	
Element	**Need**	**Possible Solution**
Tools	A way to identify specific, necessary, additional patches for 3rd party hardware and software.	Consult with product vendor
	A tool to create a master system image	Ignite-UX (IUX)
	A tool for installing patches and patch bundles	Software Distributor (SD-UX)
Delivery	Standard bundles of high-quality patches	Support Plus media
	A way to retrieve individual patches	Electronically: HP's IT Resource Center web site (http://ITResourceCenter.hp.com) Other: Request a patch tape from the HP Response Center
IT Processes	1. Process for setting up new computer systems 2. Process to install core O/S and patches from media or archive image 3. Process for installing and configuring applications 4. Process to create a system recovery/archive image 5. Release-to-production process	

Figure 1: New System Installation



Begin: Preparing to Order New System
Yes
1
2
First Unit of Application or Machine Class
Pre-Ignited?
No
Install O/S from media kit including the highest-quality bundle, using patch_match_target
Identify, acquire, & install specific necessary additional ISV patches. (For IHV - Use IPR media)
Install and configure applications
Go to Proactive Patching step 4
Additional Unit of Existing Application or Machine Class
Pre-Ignited?
Yes
Go to 2
No (or unnecessarily pre-Ignited)
Create recovery/ archive image
Have Ignite image?
No
Go to 1
Yes
Install Ignite image
End: Functioning System for Intended Role
Tools
Delivery
IT Processes

Proactive Patching

The next usage model covers proactive patching. Unlike reactive patching, which aims to solve a known problem, proactive patching seeks to prevent problems and downtime due to known problems for which a solution exists. The usage model for proactive patching is listed in Table 2 on the next page and shown in Figure 2 on the following page.

The starting point for proactive patching is a system that is functioning normally. This raises the question "why patch proactively?" The simple answer is that latent problems may exist on a system that appears to be working well. There are many types of potential problems that can be avoided or fixed through proactive patching, including:

- security vulnerabilities
- memory leaks
- silent data corruption
- performance degradation

A good example of the need for proactive patching was Y2K. Systems that functioned normally prior to January 1, 2000, would likely have experienced problems if they had not been proactively updated.

Looking at the process flowchart, there are two main approaches to proactive patching. The first is normal, scheduled maintenance. This is often an opportunity for proactive support. Depending on the needs and the level of system support, the needs may be met using a system bundle or through a custom proactive patch analysis.

The other type of proactive patching is in response to a notification or event. For example, organizations that subscribe to patch notifications through HP's IT Resource Center may receive a notice about a potential security vulnerability. At that point, the IT staff responsible for systems maintenance needs to review the notice for applicability and potential risk. Based on their analysis, they may decide to do one of the following:

- apply the patch outside of a regularly-scheduled maintenance window
- defer installation until the next maintenance cycle
- not apply the patch

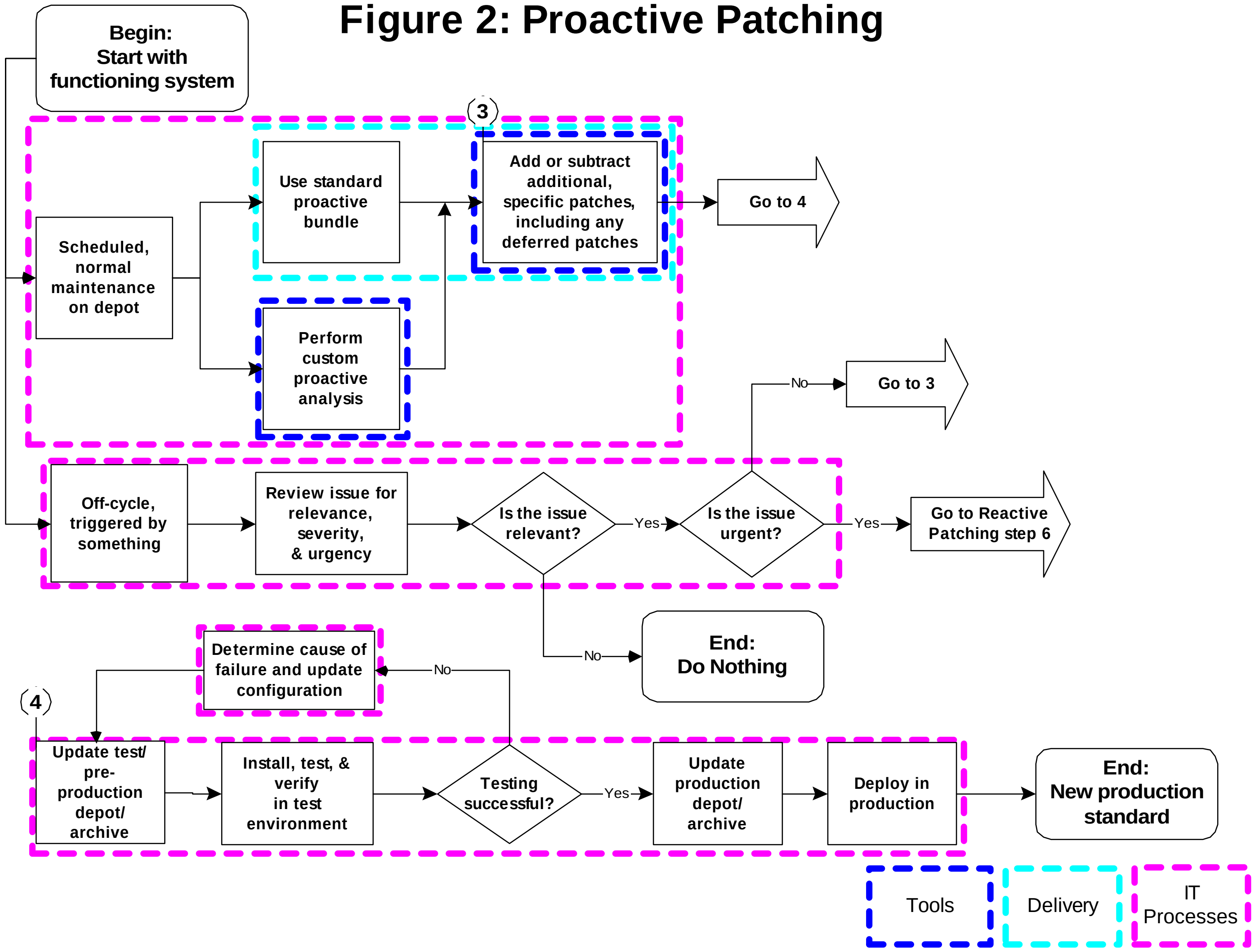
As with a new system installation, some of the steps in proactive patching require tools, and some require a method for patch delivery. All steps should be part of a comprehensive plan for software change management.

Unlike reactive patching, which has clear goals and results, the success of proactive patching is harder to quantify. In fact, the desired result is that no one notices that it has happened. But even so, proactive patching is a vital part of systems management.

Table 2: Proactive Patching

Audience	IT planners, systems administrators, and proactive support engineers	
Starting Point	Functioning system	
Completion Criteria	Updated production standard	
Benefits	♦ Problem avoidance ♦ Standardization ♦ Reduced downtime costs ♦ Reduced risk ♦ Enhanced functionality and tools	
Element	**Need**	**Possible Solution**
Tools	Tool for proactive patch notification	IT Resource Center web site (http://ITResourceCenter.hp.com)
	Tool for custom proactive patch analysis	Custom Patch Manager (CPM) available through the IT Resource Center web site
	A tool for adding or subtracting patches from a pre-determined list.	Software Distributor (SD-UX)
Delivery	Standard bundles of high-quality patches	Support Plus media
	A way to retrieve individual patches	Electronically: HP's IT Resource Center web site Other: Request a patch tape from the HP Response Center
IT Processes	1. Process for scheduled, normal maintenance 2. Process for evaluating off-cycle events and notifications 3. Process for testing changes in a non-production environment, including: ♦ Software depot creation ♦ Installation on test system ♦ Operating system verification testing ♦ Application verification testing ♦ Move-to-production process ♦ Production verification	

Figure 2: Proactive Patching



Begin: Start with functioning system
3
Use standard proactive bundle
Add or subtract additional, specific patches, including any deferred patches
Go to 4
Scheduled, normal maintenance on depot
Perform custom proactive analysis
No
Go to 3
Off-cycle, triggered by something
Review issue for relevance, severity, & urgency
Is the issue relevant?
Yes
Is the issue urgent?
Yes
Go to Reactive Patching step 6
No
End: Do Nothing
Determine cause of failure and update configuration
No
4
Update test/ pre-production depot/ archive
Install, test, & verify in test environment
Testing successful?
Yes
Update production depot/ archive
Deploy in production
End: New production standard
Tools
Delivery
IT Processes

Reactive Patching

The kind of patch usage that people are most familiar with is reactive patching. This is done in response to a problem that is currently visible and impacting the system. The usage model for reactive patching is presented in Table 3 and Figure 3 on the next two pages.

The starting point for reactive patching is a system with a problem. When this happens, the critical first step is to determine the root cause of the problem. This requires diagnostic tools and processes for troubleshooting.

A common but inappropriate approach to reactive patching is to apply many changes at once, hoping that one of them will fix the problem. This might consist of applying a patch bundle, or it might simply be the application of several individual patches. In the past HP support engineers would sometimes tell customers to "bring the system up to the latest patch level" before they would attempt to resolve a problem. This approach is no longer recommended by HP Response Center engineers.

There are many reasons why it is wrong to make several changes at once in a reactive situation. First, even if the problem is solved, the cause of the problem will remain unclear. Second, any change introduces some amount of risk. The more changes that are made, the greater the risk. Especially in reactive support situations, changes need to be minimized. Finally, if the application of several patches does not resolve the problem, the process of troubleshooting must start again at the beginning. But since the system no longer matches the production standard, even if a solution is eventually identified, it may not work for similar systems. In short, time spent in diagnosing a problem is well worth the investment.

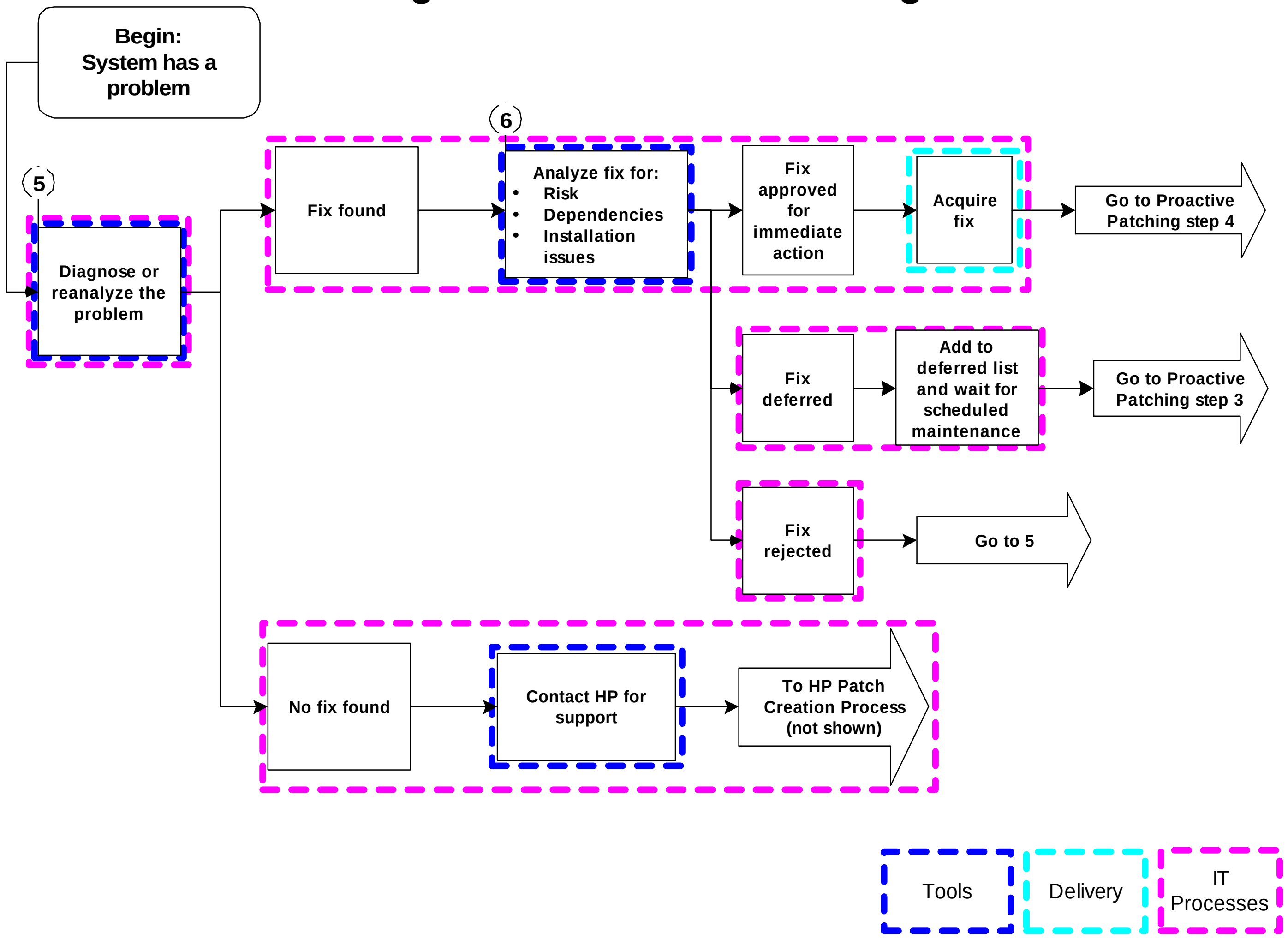
Returning to Figure 3, there are two possible results from problem diagnosis. Either a fix can be identified, or none can be found. In the latter case, it is time to contact HP support. This leads to HP's patch creation process. If, on the other hand, a patch can be found, there are still decisions to make. If the problem is severe and it is clearly addressed by the patch, it may need to be installed immediately. If the problem can wait, the fix may be deferred until the next regularly-scheduled maintenance window. Finally, it may be that the problem is not severe enough, or the fix is not complete or reliable enough, to warrant its use. In this case, the initial fix is rejected and the process starts over.

For reactive patching, tools play a key role in the process. Tools help the system administrator to diagnose the problem. They also help in testing and implementing a solution. It is also critical to have established procedures in place for reactive support.

Table 3: Reactive Patching

Audience	Systems administrators and reactive support engineers	
Starting Point	System is experiencing a problem	
Completion Criteria	The problem has been resolved or is judged to be minor	
Benefits	♦ Problem resolution ♦ Timely delivery of fix ♦ Controlled changes	
Element	**Need**	**Possible Solution**
Tools	Diagnostic tool to determine root cause of problem	Diagnostic & support media available on Support Plus
	Tool to analyze patches for: ♦ Applicability ♦ Risk ♦ Dependencies	IT Resource Center web site (http://ITResourceCenter.hp.com)
	A tool for installing patches	Software Distributor (SD-UX)
	A means to submit bugs to HP for which no patch exists	HP Response Center
Delivery	A way to retrieve individual patches	Electronically: HP's IT Resource Center web site Other: Request a patch tape from the HP Response Center
IT Processes	1. Problem diagnosis procedures 2. Process to evaluate candidate solutions 3. Process to submit a new defect report to vendor 4. Process for testing and validating fixes in non-production environment 5. Process for distributing fix to software depot servers 6. Release-to-production process for fix	

Figure 3: Reactive Patching



Begin: System has a problem
5
Diagnose or reanalyze the problem
6
Fix found
Analyze fix for:
• Risk
• Dependencies
• Installation issues
Fix approved for immediate action
Acquire fix
Go to Proactive Patching step 4
Fix deferred
Add to deferred list and wait for scheduled maintenance
Go to Proactive Patching step 3
Fix rejected
Go to 5
No fix found
Contact HP for support
To HP Patch Creation Process (not shown)
Tools
Delivery
IT Processes

Configuration Change

System changes are made for many reasons. Some examples include:

- Addition of new hardware
- Replacement/upgrade of existing hardware
- Upgrade of existing applications
- Installation of new software
- Migration to a new hardware platform

Whenever a change is made, it may be necessary to add or update the patches on a system. Table 4 and Figure 4 on the next two pages describe this process. There are two distinct paths depending on whether the change is related to hardware or software. In the case of adding new hardware, the patches required are likely to be a mix of bug fixes and hardware enablement drivers. Once the necessary patches are identified, the process follows the standard path of retrieval, validation, and deployment.

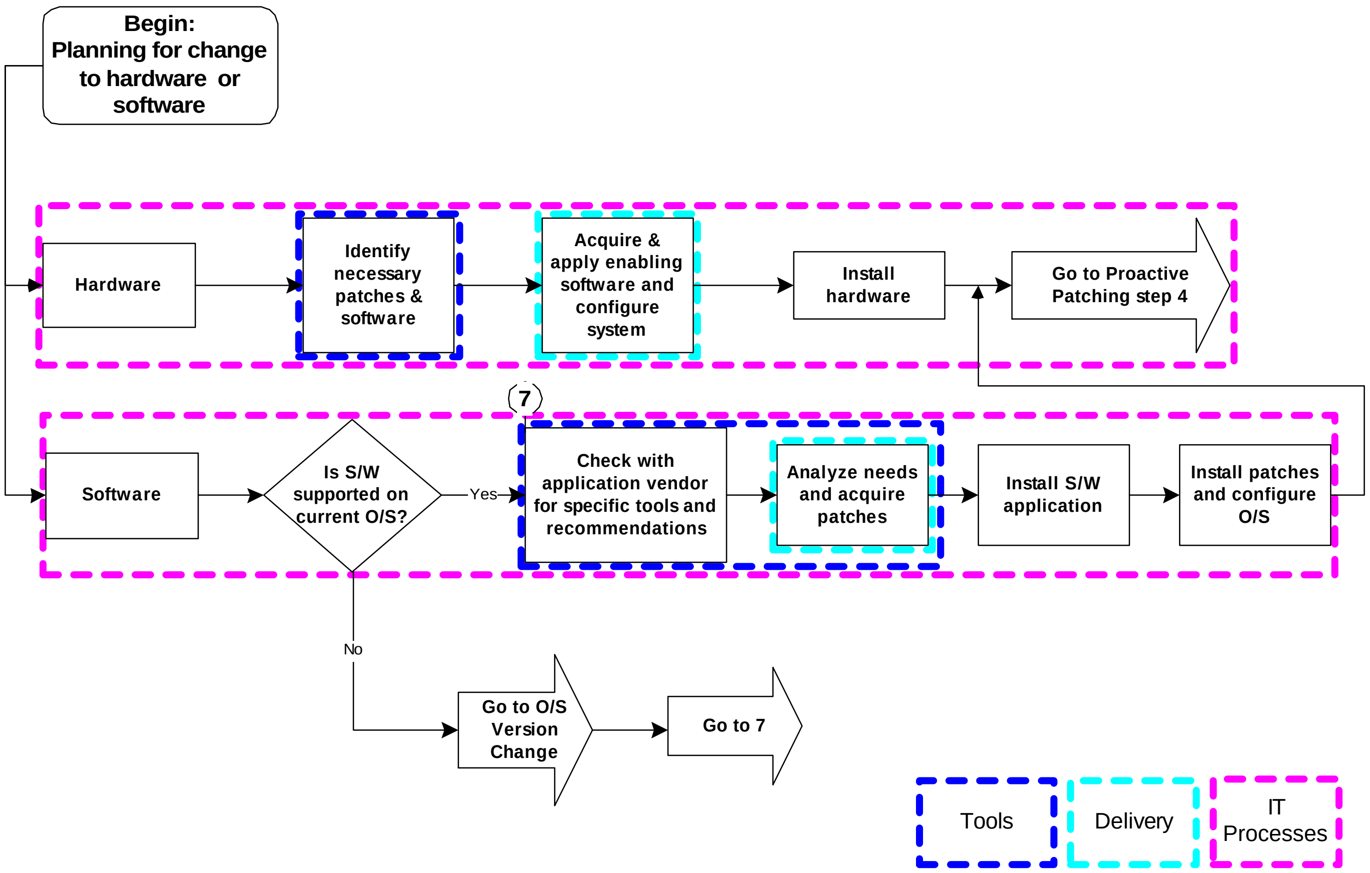
Because each application is unique, it may not be possible to develop specific IT processes for upgrades. However, a generic process map will still help to outline the steps required. For many changes, the first thing that must be determined is whether the new application or version is supported on the current operating system. If the hardware or software involved requires a new version of the operating system, this change must be made first.

Unlike the core operating system, HP does not always have a set of recommended patches for third-party applications. Rather, each application provider must specify what combination of operating system and patches their software is certified to work with. (See the usage model for ISV/IHV qualification.) For this reason, it is often necessary to contact application vendors to get their patch recommendations. As with hardware, once the necessary patches have been identified, they must be retrieved. The process then moves on to testing and deployment.

Table 4: Configuration Change

Audience	IT planners and systems administrators	
Starting Point	Planning for a change to hardware or software	
Completion Criteria	New hardware or software is installed and functioning properly	
Benefits	♦ Minimized downtime ♦ Upgraded/enabled new hardware or software ♦ Stable operations	
Element	**Need**	**Possible Solution**
Tools	A tool to identify necessary patches	Vendor recommendations
	Tool to analyze patches for: ♦ Risk ♦ Dependencies ♦ Applicability	IT Resource Center web site (http://ITResourceCenter.hp.com)
	A tool for installing patches	Software Distributor (SD-UX)
Delivery	A way to retrieve individual patches	Electronically: HP's IT Resource Center web site Other: Request a patch tape from the HP Response Center
IT Processes	1. Process to evaluate possible changes to configuration 2. A hardware installation process 3. A new software installation process 4. A software upgrade process 5. A process for implementing, testing, and evaluating changes in a non-production environment 6. Process for distributing software changes to software depot servers 7. Release-to-production process	

Figure 4: Configuration Change


Begin: Planning for change to hardware or software
Hardware
Identify necessary patches & software
Acquire & apply enabling software and configure system
Install hardware
Go to Proactive Patching step 4
7
Software
Is S/W supported on current O/S?
Yes
No
Check with application vendor for specific tools and recommendations
Analyze needs and acquire patches
Install S/W application
Install patches and configure O/S
Go to O/S Version Change
Go to 7
Tools
Delivery
IT Processes

Operating System Version Change

One of the most challenging tasks facing a systems administrator is performing an operating system version change. Maintaining data integrity while minimizing system downtime can be very difficult. For this reason, the first step in the change process should be a determination of whether the change is really necessary. While a new version of the operating system may offer advanced features -- 64-bit computing, for example -- these must be balanced against requirements for stability and availability.

Often, a version change is required when migrating to a new hardware platform. For example, the latest server platform may require the latest operating system version. In this case, the challenge is compounded by the fact that both the software and the hardware are being changed. Since the new hardware will not run the current operating system, the only option is to perform a new installation and then migrate the existing data. Before this is done, it is imperative to verify that this operation is supported by the applications involved. A data migration plan is a key component in the overall process. The format should be similar to a disaster recovery plan, and like a disaster recovery plan, it should be thoroughly tested in advance.

If, after reviewing all available options, an in-place version change is decided upon, the next step is to review the existing system configuration. In terms of patches, different operating system versions require different patch versions. In many cases, patches from preceding O/S versions have been incorporated in the more recent O/S. However, if a patch is released for a particular subsystem or application after both versions of the O/S are in production, there will be equivalent patches for each operating system. The concept of patch equivalency is best illustrated with an example. Assume that a new tape drive is released that requires an enablement patch to work with HP-UX. There is likely to be one patch that works with HP-UX 10.x, and an equivalent patch that works with HP-UX 11.x. In planning an O/S change, it is important to establish a patch equivalency list to ensure that everything will function properly once the change is complete.

Testing and validation play a vital role in performing an operating system version change. Because every configuration is different, systems used to evaluate changes should be as similar as possible to the systems used in production. In addition to the operating system, evaluation systems should be loaded with the complete operating system, application, and data stack. If possible, simulated load testing should be performed. Rigorous testing in the evaluation phase will help to ensure success in production.

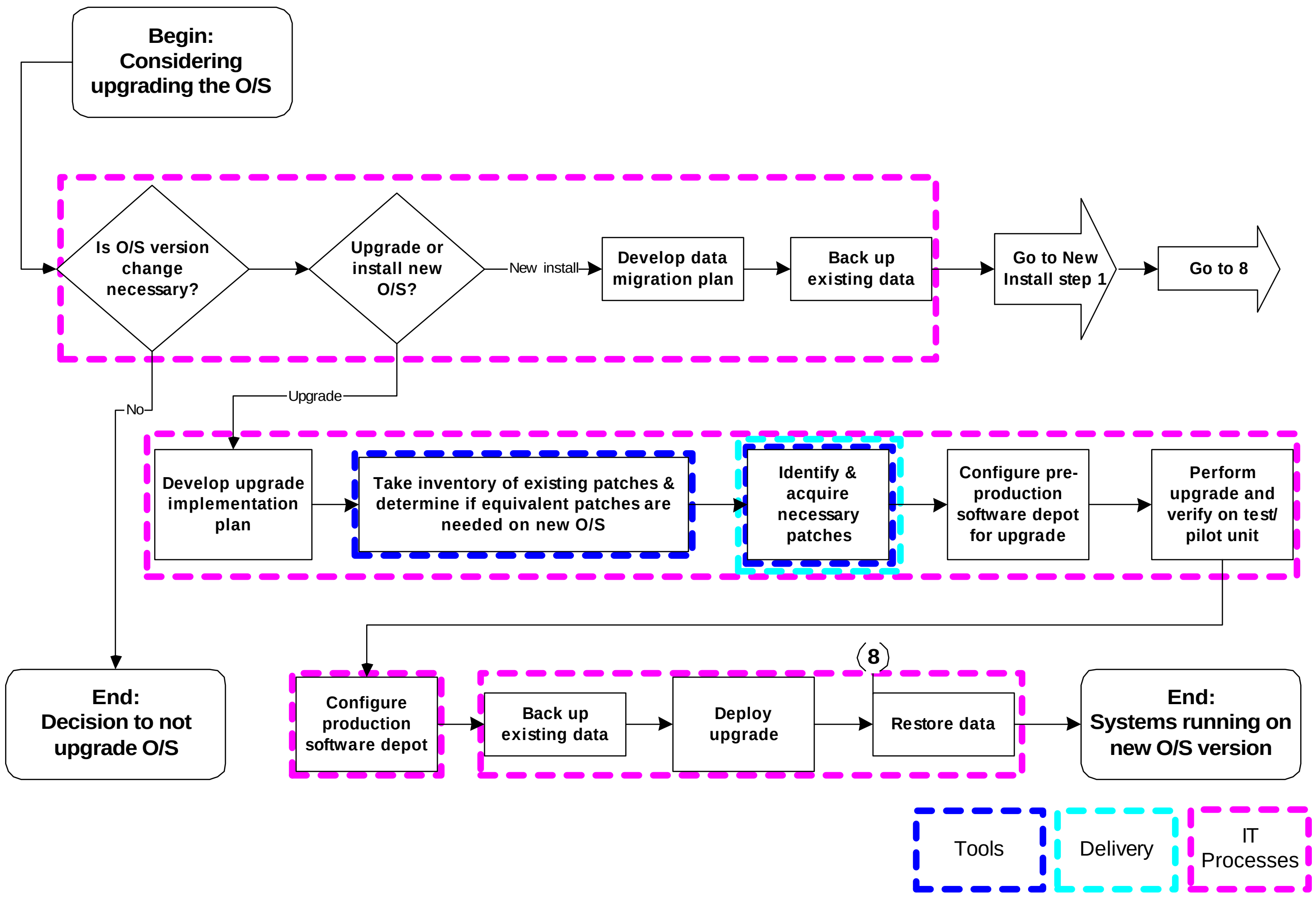
As outlined above, a data migration plan should be developed for an operating system version change. The ability to upgrade the operating system is no guarantee that existing applications and data will continue to work. (See configuration change usage model.)

Finally, a formal release-to-production process should be established and followed when moving a change into production. This should include details about master software location and procedures for installation and validation. It should also include contingency planning in case of problems, including provisions for backing out changes and returning to the original configuration.

Table 5: Operating System Version Change

Audience	IT planners and systems administrators	
Starting Point	Considering a different O/S version	
Completion Criteria	System and applications functioning on new O/S version, if required	
Benefits	♦ New hardware or software capability/enablement ♦ Safety of data ♦ Timeliness ♦ Enhanced functionality	
Element	**Need**	**Possible Solution**
Tools	A tool to identify necessary patches	IT Resource Center web site (http://ITResourceCenter.hp.com)
	A tool to identify equivalent patches on different versions of the operating system	IT Resource Center web site -- Patch Equivalency Tables
	A tool for installing patches	Software Distributor (SD-UX)
Delivery	A way to retrieve necessary patches	Electronically: HP's IT Resource Center web site Other: Request a patch tape from the HP Response Center
IT Processes	1. Process to evaluate possible changes to an O/S version 2. A software upgrade process 3. A process for evaluating changes in a non-production environment 4. Process for distributing software changes to software depot servers 5. A data backup process 6. A data migration process 7. A data restore process 8. Release-to-production process	

Figure 5: Operating System Version Change



Begin: Considering upgrading the O/S
Is O/S version change necessary?
Upgrade or install new O/S?
New install
Develop data migration plan
Back up existing data
Go to New Install step 1
Go to 8
No
Upgrade
Develop upgrade implementation plan
Take inventory of existing patches & determine if equivalent patches are needed on new O/S
Identify & acquire necessary patches
Configure pre-production software depot for upgrade
Perform upgrade and verify on test/ pilot unit
End: Decision to not upgrade O/S
8
Configure production software depot
Back up existing data
Deploy upgrade
Restore data
End: Systems running on new O/S version
Tools
Delivery
IT Processes

ISV/IHV Qualification

The final usage model involves qualification testing by independent software vendors (ISV's) and independent hardware vendors (IHV's). Since the requirements for this type of patching vary depending on the vendor's goals, there are no firm rules for this usage model. But there are some general guidelines that can help in establishing a target. These are shown in Table 6 and Figure 6.

Qualification testing often involves conflicting goals. On the one hand, there may be the need to test with the latest features. On the other is the need for a stable O/S. Often the best approach to balancing these requirements is to use standard patch bundles to establish a stable base O/S and then to add any necessary individual patches.

As indicated in Figure 6, there are two basic paths for this usage model. The first branch is followed if the product is new, or if this is the first time it is being qualified on a particular operating system. The other path is for products that have already been qualified on one version of the O/S, and are being re-qualified on a different O/S version or configuration.

In the case of a new qualification, the first step is to choose a target O/S version or versions. This decision is clearly up to the vendor based on their internal research and goals. Once the operating system version is chosen, the next step is to establish the degree of compatibility desired. This will help to drive the choice of patches. For example, if a hardware vendor is developing a new interface card that will only work on the newest platform, they will probably want the newest O/S version, features, and patches. If, instead, they want maximum compatibility, they may need to choose multiple operating system versions. In this case, the vendor will probably want to avoid newer features in establishing a standard.

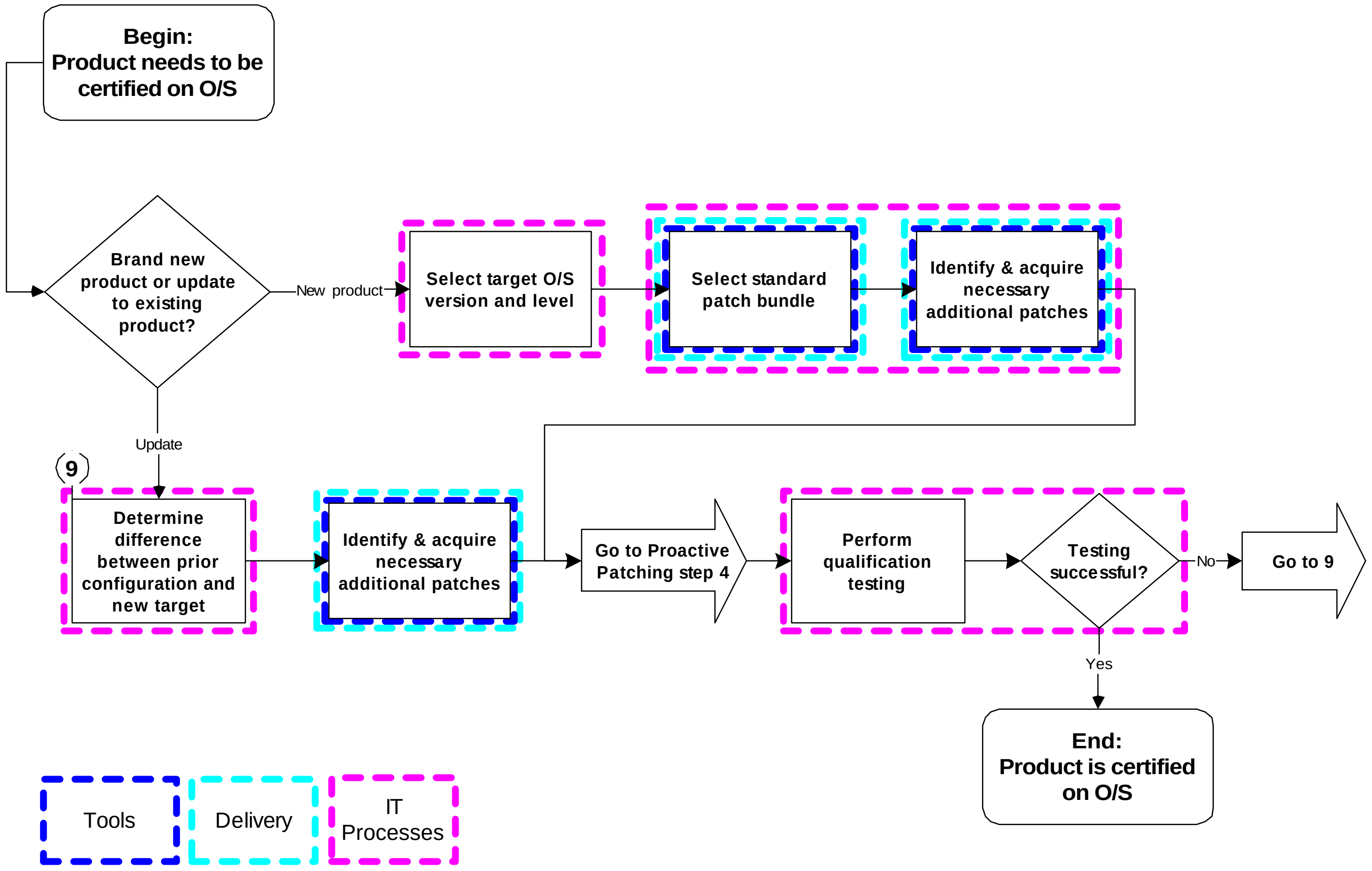
The process is somewhat easier for a product that has already been certified on a different version of the operating system. For this situation, the first step is to analyze the differences between the certified O/S version and the target. For minor version changes, it may be possible to simply add patches and re-test. For changes in major O/S release, the process is similar to a brand new certification.

Once the O/S and patches have been configured, qualification testing can begin. Again, the type of testing involved will vary. If the testing is successful, the process ends. If not, the reasons for certification failure must be determined. If they are related to the O/S configuration, it may be possible to change some system software and patches and to re-test. Otherwise, hardware or software re-work may be required. The process is repeated until the product passes the certification tests.

Table 6: ISV/IHV Qualification

Audience	Independent software and hardware vendors	
Starting Point	Product needs to be certified on O/S	
Completion Criteria	Product is certified on a particular O/S version	
Benefits	♦ Confidence in result ♦ Established standard for product	
Element	**Need**	**Possible Solution**
Tools	A way to select from standard patch bundles	Support Plus bundle usage matrix
	A tool to identify necessary additional patches	IT Resource Center web site (http://ITResourceCenter.hp.com)
	A tool for installing patches	Software Distributor (SD-UX)
Delivery	Standard patch bundles	Support Plus media (software) Independent Product Release (IPR) media (hardware)
	A way to retrieve necessary patches	Electronically: HP's IT Resource Center web site Other: Request a patch tape from the HP Response Center
IT Processes	1. Process to select target O/S version and configuration 2. A software installation process 3. A configuration update process 4. Certification testing process	

Figure 6: ISV/IHV Qualification



Begin: Product needs to be certified on O/S
Brand new product or update to existing product?
New product
Select target O/S version and level
Select standard patch bundle
Identify & acquire necessary additional patches
Update
9
Determine difference between prior configuration and new target
Identify & acquire necessary additional patches
Go to Proactive Patching step 4
Perform qualification testing
Testing successful?
No
Go to 9
Yes
End: Product is certified on O/S
Tools
Delivery
IT Processes

4. Summary: Putting Theory into Action

The six patching usage models presented in this paper cover some common goals of patching. The are summarized in Table 7. There are undoubtedly other ways in which patches are used that were not covered. A usage model is a template that describes the goals of a process and the steps involved. The models go a step further by identifying the key tools, delivery methods, and processes necessary to support the overall goal.

Table 7: Summary of Patching Usage Models

Usage Model	Description
New System Installation	Used when installing new systems. Systems can be either the first of kind/class or incremental units in an existing environment. The focus is on establishing or maintaining production standards.
Proactive Patching	Covers proactive systems maintenance. Proactive patching avoids failures due to known problems for which solutions already exist.
Reactive Patching	Used when a system or systems are experiencing problems. The focus for this usage model is applying the minimum change necessary to restore function.
Configuration Change	This model is used when planning a change in hardware or software to an existing system. The process minimizes system downtime during the change.
Operating System Version Change	Used when an operating system version change is being considered. Emphasis is on maintaining data integrity and software stack.
Independent Software/Hardware Vendor Qualification Testing	For developers of third-party hardware and software. Emphasis is on the development of an operating system standard for qualification testing.

The task of patching falls within the larger framework of software change management. While usage models are a useful tool, they need to be integrated with other operational processes. Each operation is different. Processes must be formulated that meet specific needs. As needs change and evolve over time, processes must reflect these changes. Otherwise, they can quickly become outdated and fall into disuse. Creating, maintaining, and adhering to formal IT processes is a cornerstone of high availability computing.