



realtimepublishers.com®

The Definitive Guide™ To

Enterprise Network Configuration and Change Management

VOYENCE™

Don Jones

Chapter 6: Network Configuration Management Tools	107
Designing and Staging Projects	108
Contribution to the Configuration Management Process.....	108
Evaluation Criteria	110
Accepting and Tracking Change Requests	111
Contribution to the Configuration Management Process.....	111
Evaluation Criteria	113
Environment Inventory and Review	114
Contribution to the Configuration Management Process.....	114
Evaluation Criteria	114
Change Modeling and Risk Analysis.....	115
Contribution to the Configuration Management Process.....	116
Evaluation Criteria	117
Change Implementation and Deployment	118
Contribution to the Configuration Management Process.....	118
Evaluation Criteria	119
Change Archival and Tracking	120
Contribution to the Configuration Management Process.....	121
Evaluation Criteria	122
Change Rollback and Recovery	123
Contribution to the Configuration Management Process.....	123
Evaluation Criteria	125
Management Reporting.....	126
Contribution to the Configuration Management Process.....	126
Evaluation Criteria	126
Knowledge Bases.....	128
Contribution to the Configuration Management Process.....	128
Evaluation Criteria	128
Problem Tracking.....	129
Contribution to the Configuration Management Process.....	130
Evaluation Criteria	130
Compliance and Enforcement.....	131
Contribution to the Configuration Management Process.....	131

Evaluation Criteria133
Summary134

Copyright Statement

© 2004 Realtimerepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimerepublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

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

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

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

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

If you have any questions about these terms, or if you would like information about licensing materials from Realtimerepublishers.com, please contact us via e-mail at info@realtimerepublishers.com.

Chapter 6: Network Configuration Management Tools

By now, you should have the idea that although proper network configuration management is possible without any tools, it is much easier when you have tools to help. In this chapter, I'll describe the various functions you'll want to look for in network configuration management tools and provide some evaluation criteria for selecting those tools. The functionality you'll want to look for includes:

- Project design and staging
- Change request acceptance and tracking
- Environment inventory and review
- Change modeling and risk analysis
- Change implementation and deployment
- Change archival and tracking
- Change rollback and recovery
- Management reporting
- Knowledge bases
- Problem tracking
- Enforcement and compliance

At first glance, you might be worried that I'm going to recommend that you go out and buy 10 software packages, which isn't the case. This list isn't a list of *tools*; it's a list of major *functionality*. You'll find that most available tools implement several of these functions. For example, a good Help desk ticket-tracking product will also include management reporting, problem tracking, and knowledge base functionality.

For each of these major areas of functionality, I'll give you several pieces of information:

- A description of what the functionality is actually supposed to provide.
- An explanation of how that functionality contributes to network configuration management.
- A set of evaluation criteria, including an evaluation checklist, for evaluating tools' ability to provide the critical features that you'll need.

The evaluation criteria recommendations aren't intended to be all-encompassing or comprehensive; I'm focusing only on features that offer a contribution to network configuration management. For example, a knowledge base product that links to Microsoft's online knowledge base would certainly be a useful tool, but not from the standpoint of network configuration management, which doesn't generally involve a lot of Microsoft products.

Designing and Staging Projects

Simple changes—such as opening a new port in a firewall—are fairly simple. Although they still need to go through a complete peer review and risk analysis, those steps can go by quickly for such a straightforward change, and you don't generally need a dedicated tool to help you design and plan the change.

More complex projects, such as deploying a new router or changing your network's IP addressing scheme, definitely require design and planning. Although these steps can be easily performed without the benefit of tools (indeed, most administrators probably design their changes manually), tools with the right features can offer significant benefits.

Contribution to the Configuration Management Process

One benefit I've discussed at length in previous chapters is the use of templates to develop changes. Another is the ability for a configuration management tool to enforce a workflow for peer review and approval of changes. Both are critical to a valid configuration management process; the steps highlighted in Figure 6.1 (shaded in yellow) show how this contribution might fit into a typical configuration management process.

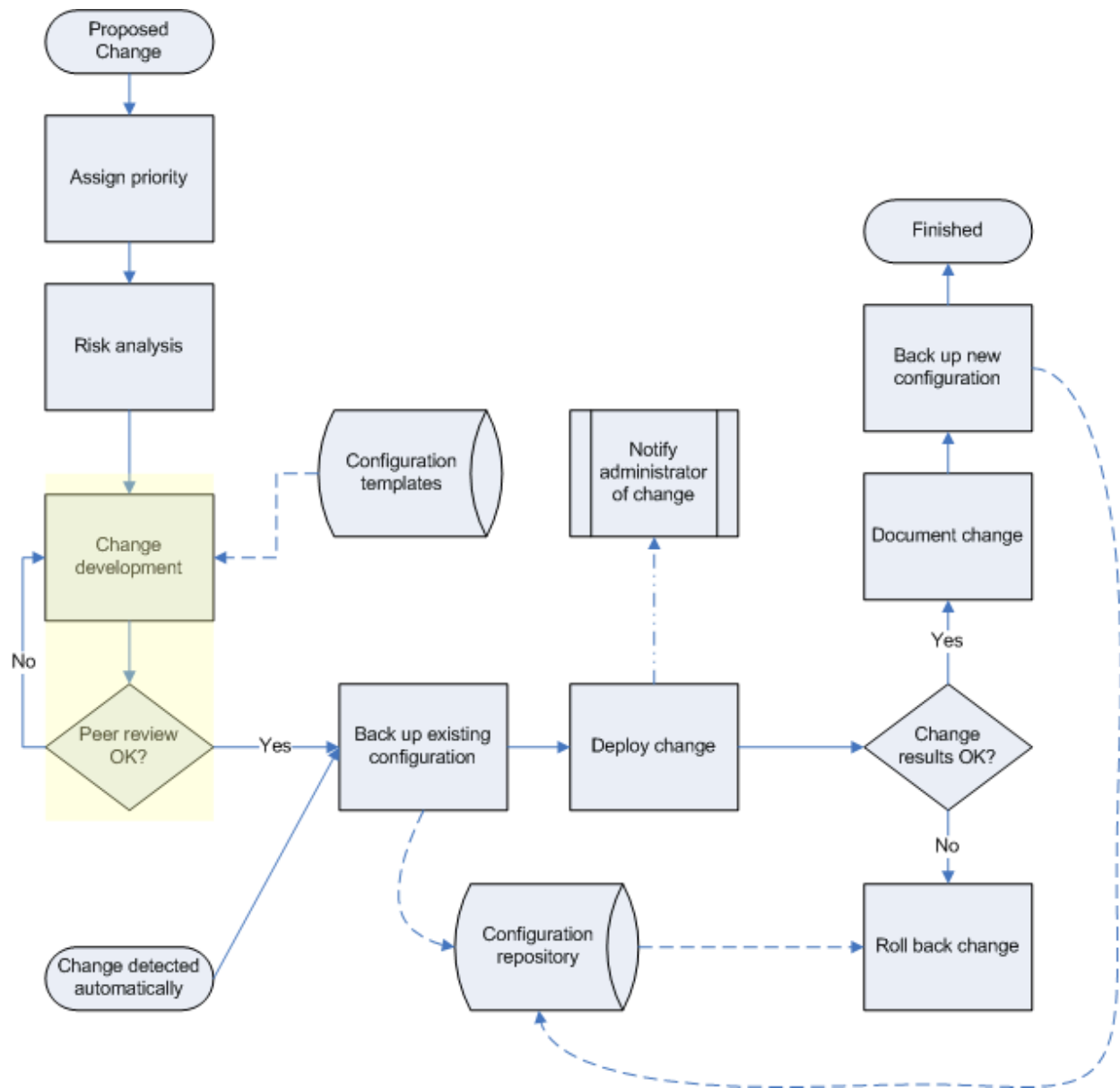



Figure 6.1: Designing and staging phases of a configuration management process.

 I discussed the importance of change planning and review in Chapters 2 and 3.

Evaluation Criteria

For designing and staging projects, you need the following specific features:

- **Template-based change design**—This capability provides additional security by enforcing configuration standards and reducing human error. Your templates can be designed to meet mandated security requirements—helping to ensure that future changes based upon those templates also meet security requirements—without requiring as rigorous a security review for each and every template-based change.
- **Template enforcement**—Rather than simply providing templates as something you *can* use, some tools can *require* you to use them. This feature can help improve consistency and reduce errors and the introduction of security flaws.
- **Workflow management**—A tool should accept proposed changes and revisions to those changes but not allow the changes to be deployed until the changes have been reviewed and approved. Ideally, the tool should allow some level of customization to the workflow process so that you can define a process that meets your environment’s specific needs.
- **Security scanning and analysis**—The solution should ideally provide some basic analysis of proposed configurations to point out known security vulnerabilities. Often, this feature is integrated in the templating process or in a pre-deployment review; what is important is that some automatic scan of the configuration highlight known problems and provide an opportunity to correct them.

Figure 6.2 provides a handy evaluation checklist for this functionality that you can use during your tool evaluations.



For all the checklists included in this chapter, as a methodology for evaluating tools, assign a suitability score—such as 1 to 3—for each feature. Products with higher scores should implement features in a way that makes more sense for your environment; products with lower scores provide features, but perhaps don’t do so in quite the fashion you would prefer.

	Product	Product	Product	Product	Product
Template-based					
Template enforcement					
Workflow					
Security checks					

Figure 6.2: A tool evaluation checklist for the designing and staging functionality.

Accepting and Tracking Change Requests

Keeping track of change requests is an essential part of the change management process. Whether these requests come from a user or are generated internally by your network administration team, the change request should be the trigger for your entire change management process. Documenting the request will provide historical reporting and an anchor point for your process. The request allows ownership of the change to be easily assigned to the person designing it, then the person reviewing the design, the implementer, and so forth—this designation helps to provide a more orderly flow within your configuration management process.

Contribution to the Configuration Management Process

The ability to accept and track change requests should be considered crucial, although you might need to manage this aspect of configuration management with a tool other than your primary configuration management solution (such as a Help desk ticket-tracking system). As Figure 6.3 shows, change tracking plays two important roles in a configuration management process (as highlighted in yellow). First, it provides the starting point for all changes. Second, it provides a final checkpoint for evaluating completed changes to determine whether the completed changes do, in fact, fulfill the request.

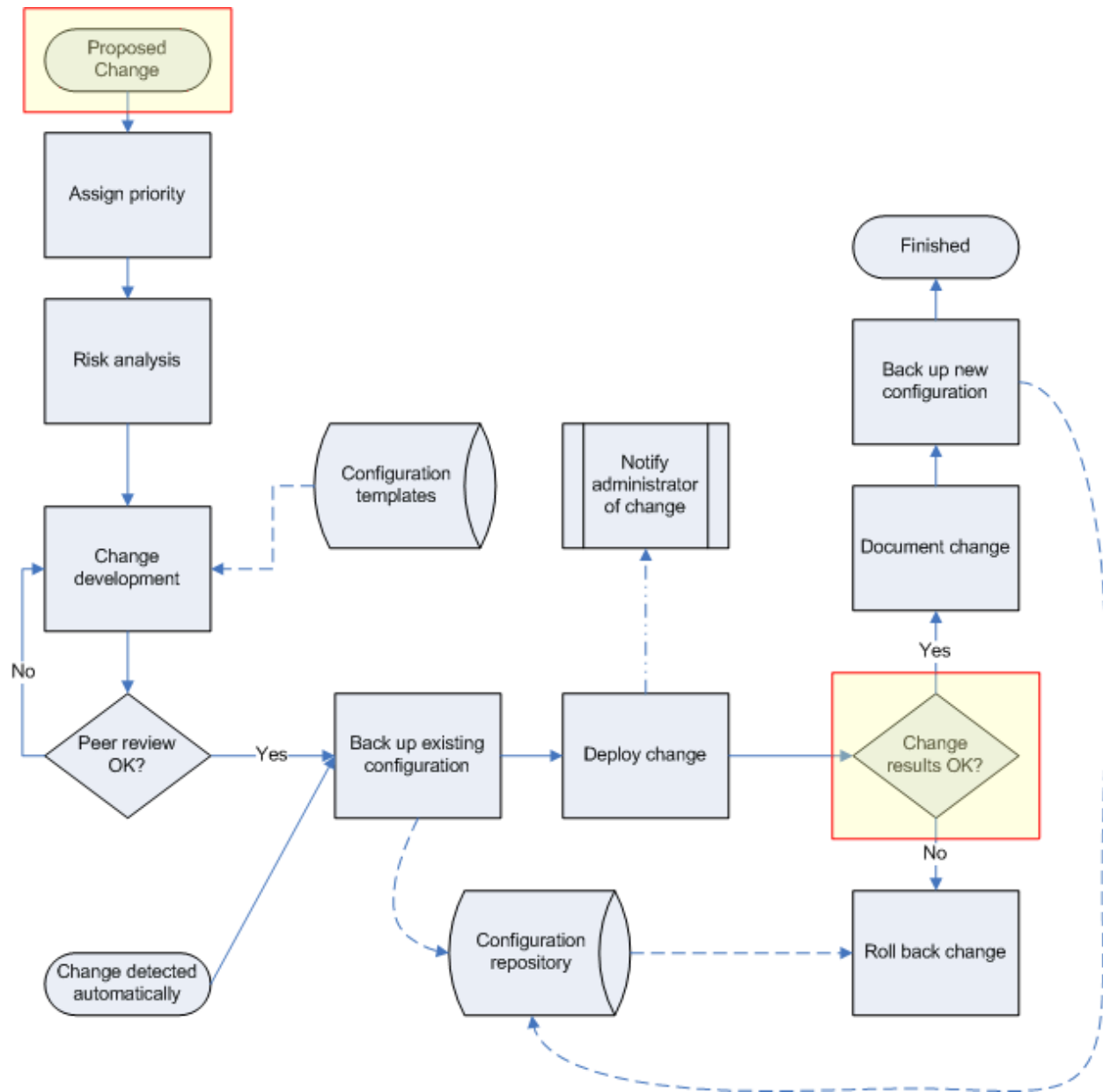


Figure 6.3: Accepting and evaluating change requests.

Most change request-tracking functionality can be provided by Help desk ticket-tracking systems. Change requests can then be reviewed and assigned a priority, assigned to a particular network administrator, and used to control “ownership” of the change throughout the process.

Evaluation Criteria

The ability to accept and track change requests is a crucial part of configuration management. The basic functionality you'll need includes:

- The ability to record requests and assign a severity to them. Ideally, you should be able to define your own levels of severity.
- A way to track historical information about the request. For example, your process might require a weekly review of all recent requests, and the results of that review might be documented in the change request itself. Comments such as “Can't currently complete, re-evaluate after new version of router OS deployed” can provide valuable information both for your administrative team and to the original requestor.
- An effective change-tracking system should include a way to embed or link to external documents such as network diagrams. These *exhibits*, as they're often called, can provide a more accurate description of the desired change than several paragraphs of text can provide.
- The ability to leave a change request open until the change has not only been completed and deployed but also evaluated for effectiveness. This last step in the process helps to ensure that the change, as deployed, meets the needs specified in the request.

As I previously mentioned, this functionality might not be readily available in a single tool. For example, you might use a Help desk ticket-tracking system to accept the change request and a standalone network configuration management solution to actually manage the workflow of developing the change, having it reviewed, and so forth. This process can be effective and might better fit the way your company is organized—with a Help desk accepting change requests and network engineers then actually working on the changes. Figure 6.4 provides a checklist for this functionality.

	Product	Product	Product	Product	Product
Assign severity					
Historical tracking					
Exhibits					
Final evaluation					

Figure 6.4: A tool evaluation checklist for the accepting and tracking change requests functionality.


Environment Inventory and Review


Many organizations don't have a solid grasp on what is already in their environments; the ability to automatically inventory, archive, and review your existing environment is a critical feature for any configuration management process. This inventory process should include some sort of auto-discovery mechanism so that devices you might otherwise forget about—like that seldom-used router connected to your backup ISDN WAN links—are discovered and incorporated into the inventory.

Contribution to the Configuration Management Process

By providing an initial baseline of what you have, inventory is often one way to begin implementing configuration management in your environment. Inventory is also the best way to stay on top of periodic changes and to provide an ongoing history of configurations within your environment. Only an ongoing inventory can provide an accurate picture of your network; simply relying on an archive of changes created by a tool will miss any changes made outside that tool.

Another function of inventory often includes automatic detection of changes to devices, in the event that a device is modified outside the primary configuration management process. This capability often includes integration with Syslog, RADIUS, or TACACS+ logs, and can usually take automated recovery actions (which you define) or, at the very least, generate notifications.

 I'll discuss this integration in more detail later in this chapter.

 In previous chapters, I've discussed how you can use tools such as cron and command-line scripts to create your own rudimentary inventory system. Although this type of inventory system will work, it won't provide integrated reporting, notification of unmanaged changes, and other key elements of a good configuration management process. Thus, more sophisticated, automated tools become so important in managing networking configuration.

The inventory process can also serve as an ongoing backup system, backing up device configurations so that no matter what else happens, you have a valid, recent backup to use for recovery purposes (such as a device failure).

Evaluation Criteria

Inventory and review capabilities should include the following:

- Absolutely vendor-neutral technologies—I've never run across a completely homogenous network; there is always something—a wireless access point, a firewall, or a switch—that is not your “usual brand.” Having a solution that can work across different vendors' products is absolutely critical.
- The ability to maintain an ongoing inventory on a scheduled basis—For example, if a new security-related patch is released for your routers, your network configuration management solution should be able to provide a report of devices in need of the update, based upon their model number and current OS revision level. This inventory should be automatically updated on a regular basis to maintain its accuracy.

- Integration with logging systems such as Syslog, SNMP, TACACS+, or RADIUS—This integration provides automated re-inventorying of devices whose configuration has been changed outside the scope of your configuration management tool.
- The inventory should include whatever hardware information your devices are capable of providing. For example, one useful effect of a useful inventorying function is automatic notification when a router's installed memory changes, which might be a sign that someone has pulled memory out of the router without authorization.
- Inventory should be closely tied to reporting functions, providing both historical and auditing reports as well as easy-to-access support information, such as a list of devices' model numbers, serial numbers, OS revision, and so forth.


 I'll discuss these features in more detail later in this chapter.

Figure 6.5 provides an evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Vendor-neutral					
Scheduled inventory					
Log integration					
Hardware information					
Reporting integration					

Figure 6.5: A tool evaluation checklist for the inventory and review functionality.

Change Modeling and Risk Analysis

Change modeling and risk analysis is an important step once a change has been designed. The idea is to generate a proposed change—a new configuration file for a device or even an entirely new network diagram. These proposed changes allow for both a peer review and a risk analysis in which you determine the possible negative impact the change could have on your environment and make sure you have plans in place to handle those eventualities. Tools that can help you analyze the risk, spot potential problems, and document both your risk analysis and your mitigation plans, can be valuable in minimizing downtime should the worst occur once your change is implemented.

Contribution to the Configuration Management Process

Modeling and risk analysis is the first technical step in the configuration management process. Ideally, a solution should provide a way for an administrator to make a “first pass” at what a configuration change would look like. A risk analysis can then be more effective because whoever is performing that analysis will have some concrete idea of what the change actually does: “Oh, we have to update 30 routing tables for this change? OK, I understand the risk associated with that.” As shown in the highlighted area of Figure 6.6, risk analysis—and the associated change modeling—is one of the first steps that should be done when a change is requested.

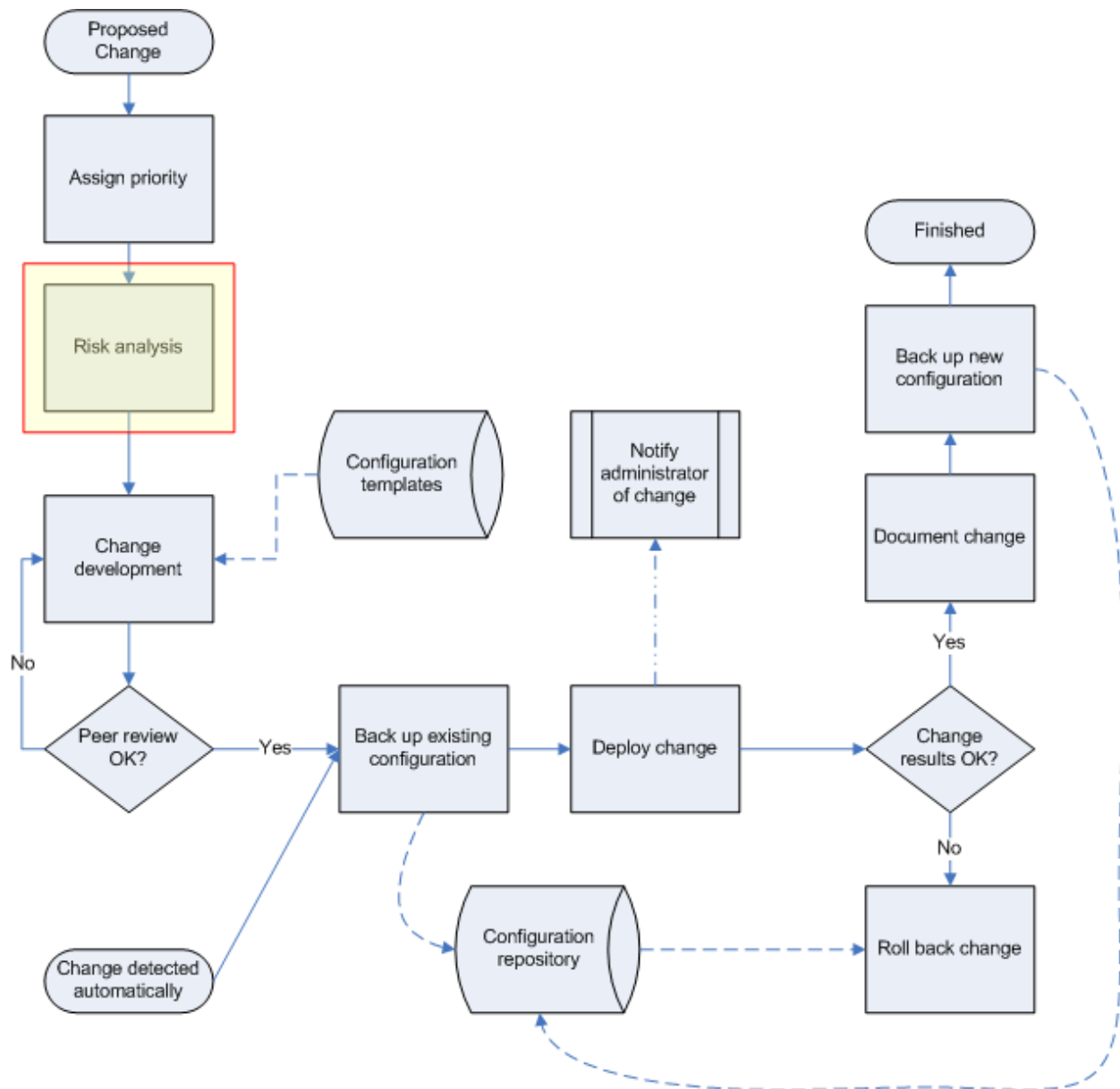


Figure 6.6: Risk analysis in a configuration management process.

Evaluation Criteria

The change model and risk analysis features you'll need include:

- The ability to document and assign risk information to a proposed change and to document and archive plans for mitigating those risks—This functionality might simply include additional text or embedded documentation in your Help desk ticket-tracking system or it might be more specific functionality in a configuration management package. Whatever tool provides it, you'll want to have clear, readily available documentation of the risks you believe could occur with a change as well as a handy way to access your “Plan B” documentation in case a problem does crop up.
- A way to create preliminary proposed changes, for the purposes of risk assessment without the danger of those changes being deployed—A role-based security model as well as a built-in workflow system that prevents changes from being deployed without some external approval sub-process can help with this feature.

Figure 6.7 provides a handy evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Assign risks					
Document mitigation plans					
Create proposed changes					
Review proposed changes					

Figure 6.7: A tool evaluation checklist for the modeling and risk analysis functionality.

Change Implementation and Deployment

Actually implementing a change can be a tricky process. Some changes might require a device restart, which means you might want to implement the change during off-hours. You might also have a strict schedule about when changes can be deployed, and a tool can help enforce those changes.

Contribution to the Configuration Management Process

Implementation is, of course, where the rubber meets the road in a configuration management process. As suggested in the highlighted area of Figure 6.8, implementation should include its own internal workflow or scripting process so that change deployment can involve multiple steps, such as backing up the existing configuration.

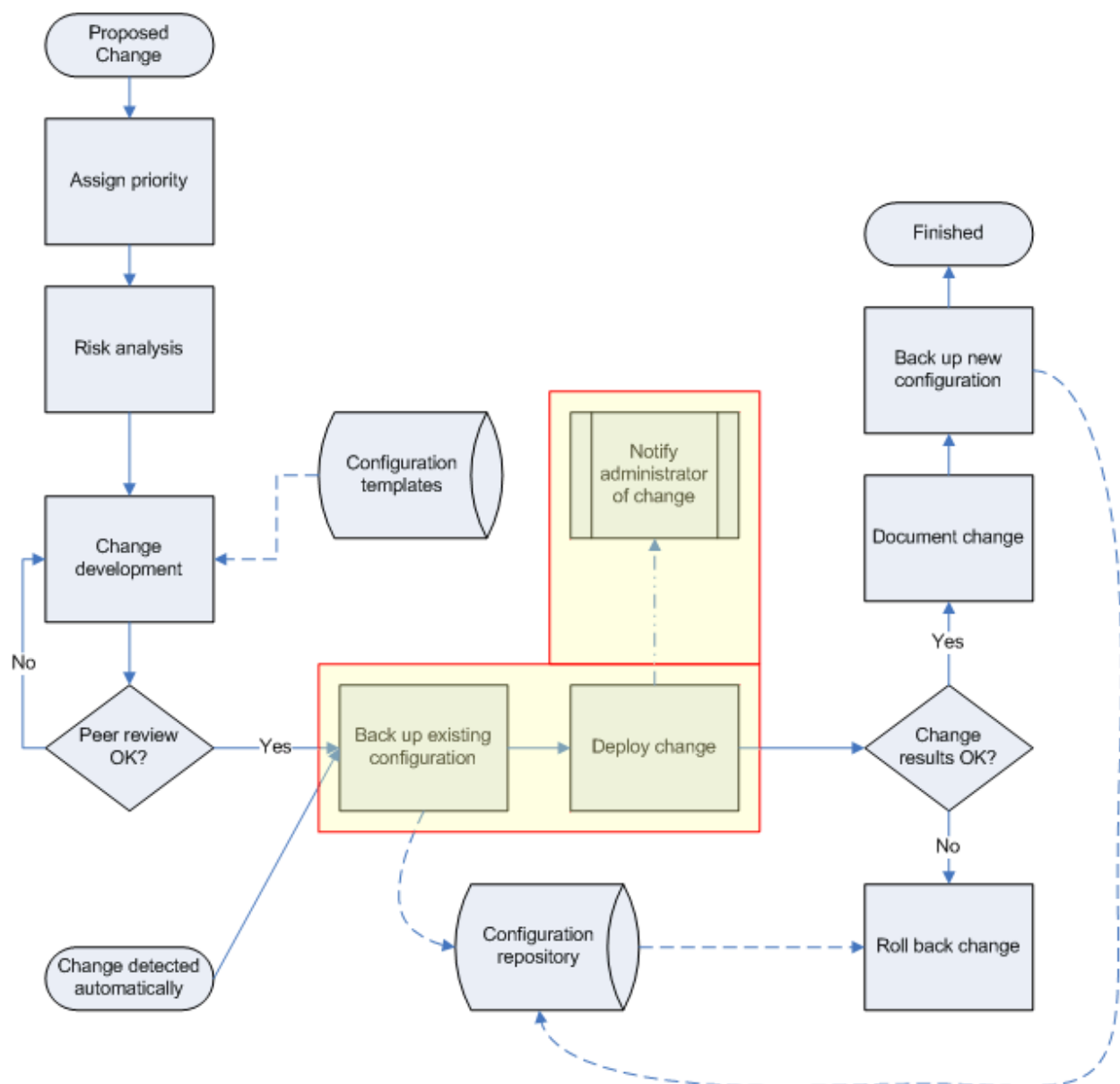


Figure 6.8: Implementation within a configuration management process.

Evaluation Criteria

Change implementation and deployment is perhaps the most important area of configuration management. Specific features to look for include:

- Multi-vendor compatibility—As with inventory capabilities, the ability to deploy changes to any vendor’s device is a must if a solution is to be useful across your entire enterprise network.
- Role-based security is a key component of deployment, ensuring that only authorized administrators can make the final decision to “push the button” and send a change out to your network devices—At the same time, role-based security allows other administrators to review and update proposed changes, if necessary, prior to deployment.
- The ability to deploy not only configuration changes but also device OS patches and upgrades—Maintaining a secure, operational network depends heavily on quickly deploying critical updates, and your configuration management solution should make it easier to do so.
- Automatic notification tied into the inventory component allows a tool to notify you of unauthorized changes—This capability requires tight integration between a deployment tool and the inventory tool (ideally, having both capabilities in a single tool) so that authorized deployments don’t trigger an alert but unmanaged or unauthorized changes do trigger an alert.
- A scripting or workflow process that allows a deployment to consist of multiple sub-steps within an overall atomic process—In other words, the implementation phase must be able to perform operations such as backing up the existing configuration, ensuring that the device targeted is online and reachable, deploying the change, notifying an administrator that the deployment completes, and so forth. These steps should either be atomic—meaning all of them complete or none of them do—or be able to incorporate logic, so that, for example, a device that can’t be backed up will halt the process, preventing the change from being deployed.
- Scheduled deployment that automates the process of deploying changes to one or more devices and does so at a designated time—This functionality can be a great labor-saving feature because approved changes can be deployed off-hours, without every network engineer on staff having to be present for the event. Scheduled deployments can be a bit scary, though, because you’re modifying critical network infrastructure components when nobody’s around to deal with possible problems. Thus, a scripting or workflow facility within the deployment tool is critical: You should be able to configure the tool to automatically roll back to a known-good configuration if it is unable to successfully deploy your scheduled change. The tool might also have some basic deployment-validation capability, such as the ability to ping a device and make sure it is still reachable, and to take basic corrective action (or at least notify someone via pager) if the validation fails.

Figure 6.9 provides an evaluation checklist for this functionality that you can use during your tool evaluations.

	Product	Product	Product	Product	Product
Vendor-neutral					
Role-based security					
Deploy changes					
Deploy OS patches & upgrades					
Automatic notification					
Scripting or workflow					
Scheduled deployment					
Atomic deployment					

Figure 6.9: A tool evaluation checklist for the change implementation and deployment functionality.

Change Archival and Tracking

Tools can also provide a sort of automatic documentation, archiving both your change and the device configuration as it existed *before* the change, providing a historical trail of the device's configuration. This archival and tracking feature can be useful from both a reporting and educational standpoint. For example, if you discover that your network has a security vulnerability that you thought had been closed up, the archive of device configuration changes can help you track exactly where the vulnerability was re-introduced. You can then modify your processes and documentation appropriately to prevent the problem from occurring again.

A tool should also be able to automatically detect changes, even ones made outside of any tool that you have. This automatic detection will allow the tool to alert you that out-of-scope changes have been made to a device (which might represent a security vulnerability or a direct attack) and allow you to take appropriate remedial action.

Contribution to the Configuration Management Process

For auditing purposes, disaster-recovery purposes, and several additional reasons, you need configuration archival and tracking. The highlighted area of Figure 6.10 shows that archiving is an integral part of the configuration management process.

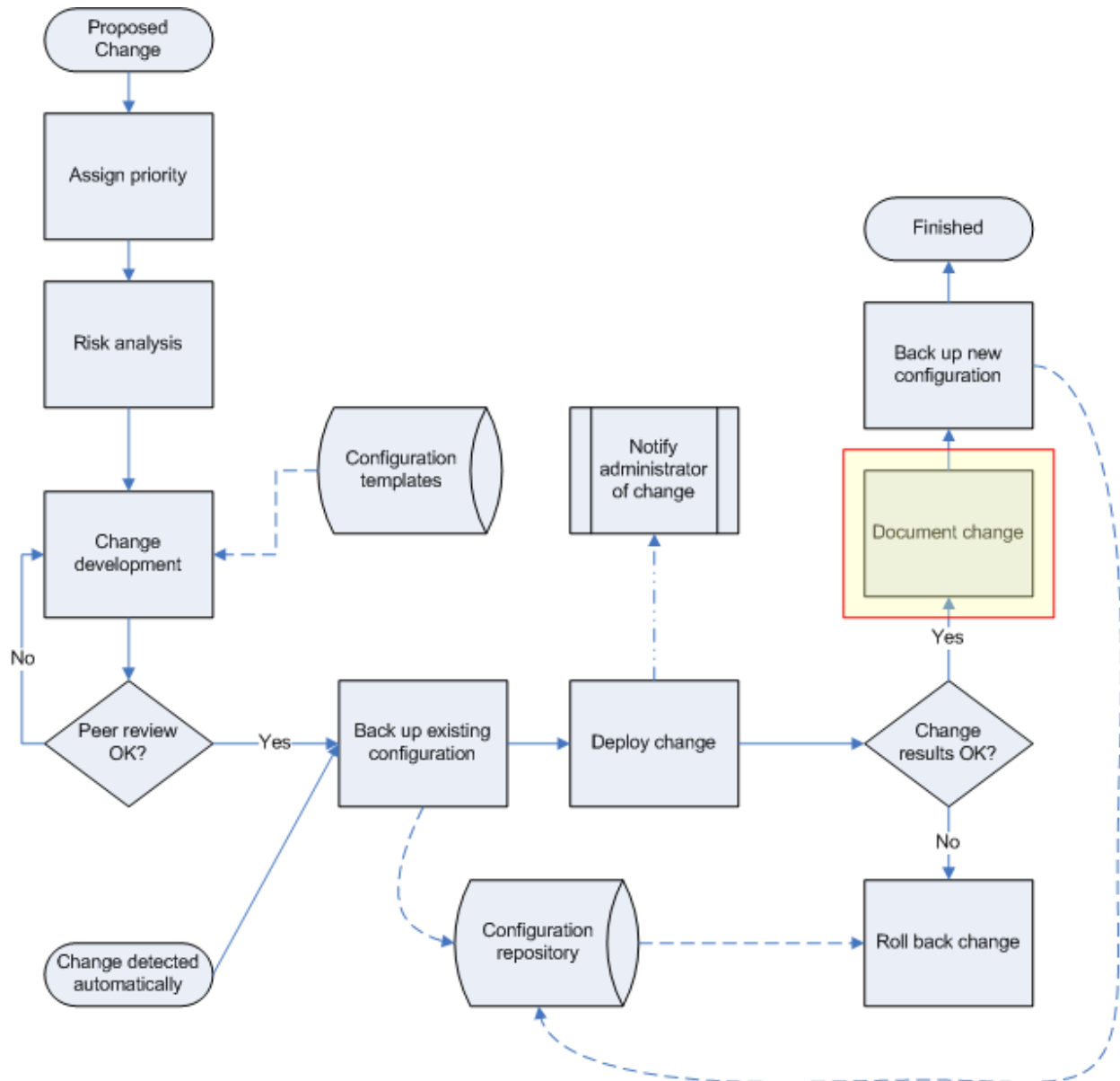


Figure 6.10: Archive and tracking within a configuration management process.

This archival function not only provides a means of rollback, should that be required, but also ongoing documentation of how your network has changed and grown—required in many organizations that are governed by security and auditing regulations (see the sidebar “The Law on Security and Auditing”).

Evaluation Criteria

Change archival and tracking is another key component for any configuration management solution. Specific features to look for include:

- Security, which is an absolute must—Your network configuration data is in many ways the “key to the kingdom,” and archival databases must be password-protected, encrypted, or otherwise secured so that your configuration information can’t be easily read by unauthorized individuals.
- The solution should include auditing capabilities, and in some industries, *must* include those abilities—Several new pieces of legislation require organizations in specific industries to maintain detailed auditing records of sensitive information, which can, in certain circumstances, include networking configuration data. Government organizations are also subject to federal regulations regarding the auditing of configuration data.

The Law on Security and Auditing

Which laws or regulations is your organization subject to? Government agencies and their contractors are usually subject to 21 Consolidated Federal Regulations (CFR) Part 11, which mandates several requirements for information systems security and auditing. In fact, stringent new rules allow the Office of Management and Budget (OMB) to suspend funding for any IT programs that do not meet federal security guidelines.

Healthcare-related companies and organizations must comply with the Health Insurance Portability and Accountability Act (HIPAA), a sweeping and comprehensive set of regulations dealing with any systems that store or transmit patient data. Financial institutions must comply with the Sarbanes-Oxley Act, another set of strict regulations dealing with customer privacy and accountability.

An effective network configuration management solution will help you comply with all of these regulations by providing an accurate history of your network’s configuration changes and the ability to pinpoint all changes, whether authorized or not. A role-based security model is essential for compliance with most of these regulations and should be another important criteria consideration. Before considering any solution for network configuration management, ask the vendors how their solutions comply with any regulations that might affect your organization.

- A means of readily accessing archived configuration data, whether for rollback purposes or simply for enterprise reporting, trend analysis, and so forth—For example, if a change is made and you later realize that your routers seem to be running somewhat slowly, you should be able to access past configuration versions for those routers to further analyze the changes that have been made over time to see where the performance problem might have crept in.

Figure 6.11 provides an evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Secure storage					
Auditing capabilities					
Access to archived configurations					
SOX compliance*					
HIPAA compliance*					
21 CFR 11 compliance*					

* if desired or required by your organization

Figure 6.11: A tool evaluation checklist for the change archival and tracking functionality.

Change Rollback and Recovery

Any device change, no matter how well-reviewed in advance, presents a potential failure of network functionality. For example, you might propose a change to your firewall that locks down a specific UDP port. Every engineer in your company might review that change and agree that it is perfectly valid and won't cause any harm; only when it's deployed do you discover that the port was open to allow the company CEO to plan an online golf game, and he's hopping mad that he can't log on anymore. Tools offering an instant rollback capability can restore the device's previous version, which is a valuable function—especially if the problem caused by the change is more serious than just upsetting the CEO's lunch hour. Configuration rollback can also help recover from unauthorized changes that were made outside the scope of your change management process.

Contribution to the Configuration Management Process

Rollback is a necessary safety net in any configuration management process. As shown in the highlighted part of Figure 6.12, you must always have this capability, although a properly managed configuration management process will hopefully render this feature unnecessary most of the time.

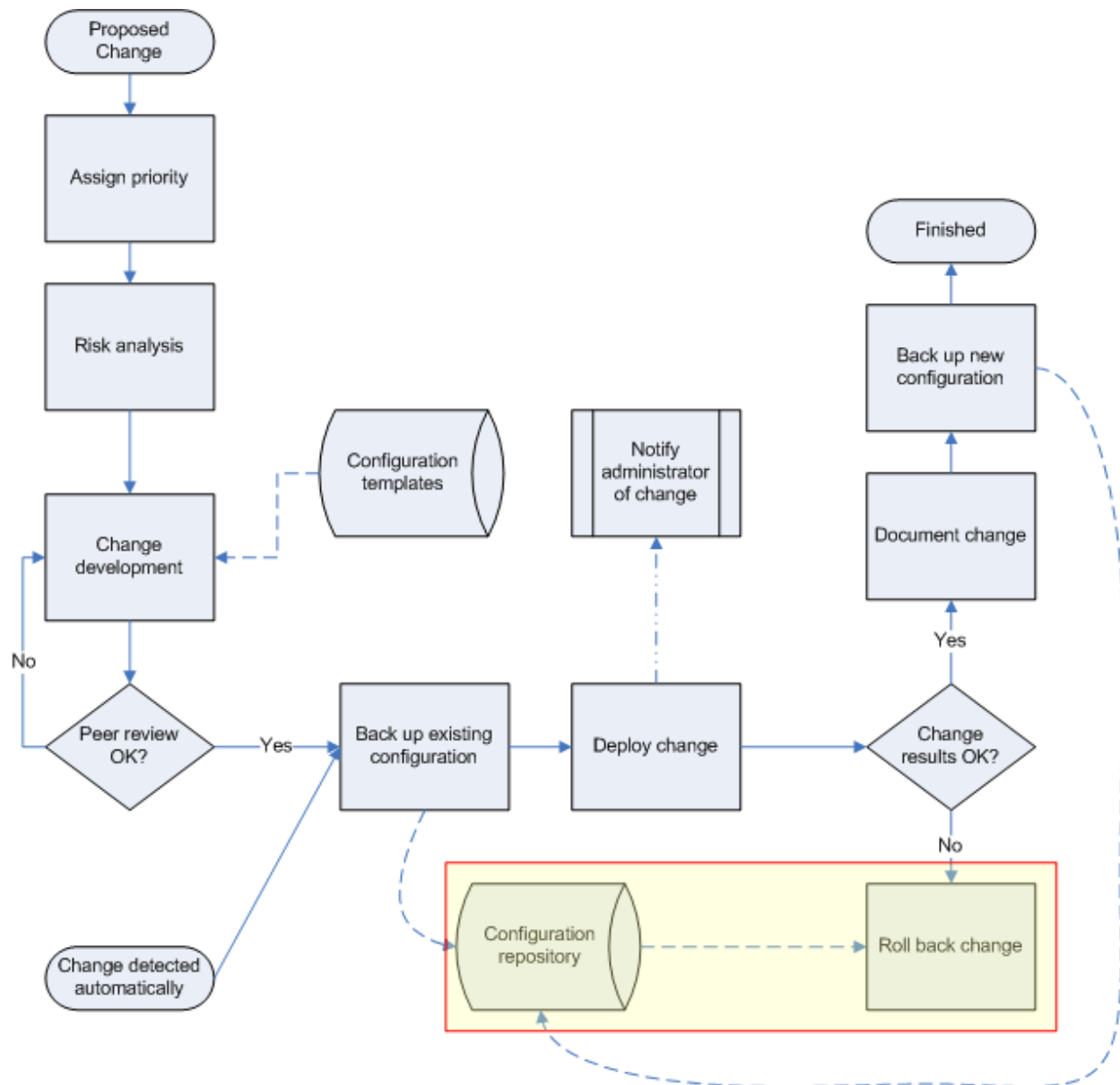


Figure 6.12: Rollback and recovery in a configuration management process.

Rollback also has a role in an enterprise disaster recovery plan. Ideally, your configuration repository should be backed up using offline media, allowing you to take that media to an off-site recovery facility and restore your device configurations on new, unconfigured devices, effectively allowing you to more quickly rebuild your production environment, if required.

Evaluation Criteria

Rollback and recovery features should include the following capabilities:

- Once again, multi-vendor support is paramount—Using a different solution for each vendor’s products will simply increase management overhead; one solution must be able to deal with the broadest possible range of network devices.
- Rollback should be an automated process that can be initiated only by authorized administrators under a role-based security model—Rollback should *not* require an administrator to select specific pieces of a configuration to roll back; administrators should simply select a prior configuration version from the database and have the solution deploy it either immediately or on a schedule.
- The ability to deploy archived configuration files to *any* device—At the very least, you should be able to retrieve an archived configuration, modify it, and deploy it to a device. This step is essential in a disaster recovery plan in which you might need to deploy slightly modified configurations to devices at a disaster recovery facility.

Figure 6.13 shows a useful evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Vendor-neutral					
Automated rollback					
Role-based security					
Deploy to any device					

Figure 6.13: A tool evaluation checklist for the change rollback and recovery functionality.

Management Reporting

Reporting is an often-overlooked aspect of many tools that contribute to the configuration management process. Reporting is, however, an essential capability. On one level, reporting can provide a more accurate picture of your network management workload than anecdotal evidence can provide. Telling the boss “We need more headcount because we’re *really* busy” is less convincing than “This report shows that we’ve been producing three times as many configuration changes as we should be able to physically handle; can we get some help?” On another level, reporting can provide additional archival data and can be useful in holding post-mortem meetings to discuss the results of recently made configuration changes.

Contribution to the Configuration Management Process

Reporting is an obvious part of any management process. As I’ve mentioned already, many organizations have legal requirements around auditing and reporting, making a good reporting function an absolute necessity.

The fact is, though, that most companies wouldn’t know which reporting features to look for in a network configuration management system. Companies don’t tend to run reports on changes made, inventory, and other items that a useful reporting system can provide; now is the time to start thinking about how you could use that information to more effectively manage your network. These reports should be viewed not only by management but also someone with a firm technical grounding. For example, a pattern of changes to a set of routers might lead you to the conclusion that those routers aren’t ideally placed on your network, and you might begin a project to re-deploy them more effectively, saving future time and management effort.

Evaluation Criteria

Management reporting is perhaps one of the most subjective areas when it comes to developing evaluation criteria; every organization has slightly different reporting needs. Some common requirements include:

- Ad-hoc reporting capability, whether intrinsic to the product or provided by some third-party reporting tool—This capability will give you all the necessary reports, although you’ll have to take the time to design them yourself.
- Basic inventory information should be reportable, including device model numbers, serial numbers, and OS revision levels—This information is important in planning ongoing maintenance as well as complying with audit reviews.
- Auditing reports are a must and should detail each change made to a device, the user that made or authorized the change, the date it was deployed, and its current status (deployed, pending, and so forth)—These reports are required in organizations that must comply with government or industry auditing regulations and in organizations that maintain rigorous internal audit procedures.
- Reporting should also include some kind of automated notification system—This system usually ties in with the solution’s inventorying system, allowing it to automatically detect changes to network devices and either perform an automated action (such as rolling back the device’s configuration) or at least create a notification or report. Figure 6.14 shows this separate process in an overall configuration management process.

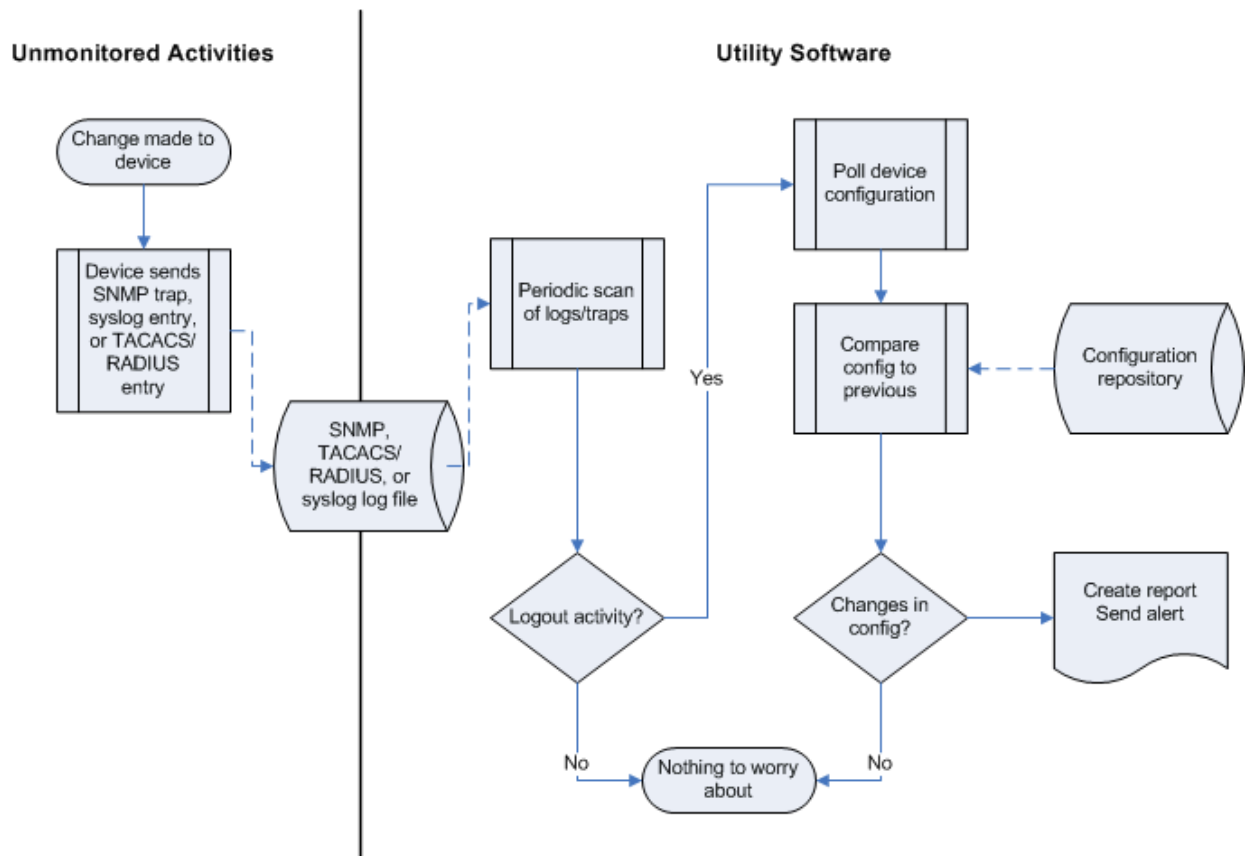


Figure 6.14: Automatic change detection and notification.

Figure 6.15 offers a handy evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Ad-hoc reports					
Inventory reports					
Auditing reports					
Automated reports or notifications					

Figure 6.15: A tool evaluation checklist for the management reporting functionality.


Knowledge Bases

Knowledge bases are often seen as a collection of past problems matched with solutions that have worked in the past. On that level, a knowledge base can be a useful tool. But knowledge bases can also provide a more proactive role. For example, searching for past problems can help you create new changes that specifically avoid those problems: “Look, the last time we modified the routing table, the network was down for a day. Let’s learn from the past and not do it again.” Knowledge bases can also be a valuable tool for new network administrators, providing them with an easier way to solve problems that have already been solved once in the past.

Contribution to the Configuration Management Process

Anyone who has never used a knowledge base will tell you that they aren’t necessary; technical professionals who have used them will tell you that you can’t live without them. Too many organizations allow critical knowledge about their network’s infrastructure to remain locked up in administrators’ heads, creating havoc when an administrator leaves the company and making it more difficult to promote those individuals for fear of losing their day-to-day involvement. A knowledge base is a simple, straightforward way of making information more readily accessible.

Typically, knowledge bases are populated from Help desk tickets or, more rarely, from supporting documents such as network diagrams and descriptions. Help desk tickets—with a detailed description of a problem and a detailed description of the solution to that problem—not only aid in knowledge transfer but also make it easier to train new administrators. Coupled with linked or embedded supporting documents, a searchable knowledge base provides a central point for finding information about the network.

 Most effective Help desk ticket-tracking systems have integrated knowledge base capabilities, automatically moving selected Help desk tickets into the knowledge base for future use. The key is in training employees to provide detailed information in their Help desk tickets so that the tickets can eventually become a valuable part of the knowledge base.

Evaluation Criteria

Knowledge bases are often incorporated in other products, but you should look for the following features:

- The ability to create “pending” documents that will eventually become part of the knowledge base after peer technical review.
- Easy search capabilities, allowing junior administrators to quickly retrieve relevant documents to aid in problem resolution.
- Problem categorization—For example, being able to categorize knowledge base documents as related to a router or switch will make it easier to review those documents when proposing future changes to those types of devices. A quick review of past problems is an excellent part of a peer review for new changes and helps to ensure that problems of the past aren’t repeated in the future.
- Web-based access—Although not strictly required, this feature makes the knowledge base more accessible, and an accessible system is more likely to be used and useful.

Figure 6.16 provides an evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
Transfer from help desk tickets					
Create pending KB documents					
Problem categorization					
Web-based access					
Search					

Figure 6.16: A tool evaluation checklist for the knowledge base functionality.

Problem Tracking

Any change, as I've said, represents a potential problem, no matter how carefully the change is researched and reviewed before being implemented. When problems do occur, a problem-tracking system is necessary to document the details of the problem, document the resolution, and eventually (ideally) become a part of your growing knowledge base. Problem tracking is often handled by the same ticket-tracking system that you use for change requests—the theory being that a network problem is just a really, really important request that needs to be taken care of immediately. That said, your ticket-tracking system needs to offer additional features if it is to accommodate problem tracking as well as change-request tracking.

Contribution to the Configuration Management Process

Problem tracking is another front-end to the configuration management process. Generally, problems are resolved through some kind of configuration change, but must be accomplished more quickly—because it's a *problem*—than a change *request*. Problems can also be generated as a part of the configuration management process: If, for example, a change is deployed and it breaks a part of your network, then a problem ticket should be opened to deal with the corrective action. This process might seem roundabout—why not just fix the configuration error?—but tracking problems in this fashion provides benefits:

- It helps you perform a post-mortem analysis to discover why your configuration management process allowed a bad change to be deployed in the first place. Was the peer review insufficient? Were steps in the process skipped for expediency?
- Tracking problems individually can improve future configuration management steps. For example, when considering a change that impacts switch port configurations, past problem tickets might indicate that changes of that nature generally result in performance problems. The individuals developing and reviewing the changes can first review past problems and try to build the new change to avoid the same problems that occurred in the past.

The result is beneficial feedback built-in to your configuration management process, making it gradually more efficient and less prone to error over time.

Evaluation Criteria

Most problem tracking can be handled by the same ticket-tracking system you use to track change requests. However, because problems tend to represent a more immediate and urgent issue than a change request, some additional features in your tools will make those tools more useful:

- Ability to assign a priority and, ideally, the ability to define your own table of priorities—This table will help you better manage problems so that the most urgent ones can be worked first.
- A notification system for problems that have been sitting idle for a period of time (which you would define)—This feature helps prevent problems from being forgotten. A means of escalating problems can also be useful in case a problem is mistakenly assigned to an engineer who is too busy to address it immediately; the escalation will move the problem to someone else for faster resolution.
- A categorization system that allows you to easily group problems—This system is especially useful if the problems are going to contribute to future changes because categories make it easier to retrieve and review relevant problems when looking at what has gone wrong with past configuration changes.

Figure 6.17 shows a checklist you can use to evaluate this functionality.

	Product	Product	Product	Product	Product
Prioritize					
Notification for idle problems					
Escalation for idle problems					
Categorization of problems					

Figure 6.17: A tool evaluation checklist for the problem tracking functionality.

Compliance and Enforcement

I've hinted at this category of functionality elsewhere in this chapter, but it's important enough to be considered independent of other feature sets. The basic idea is that any process is only effective if it's followed, and the very best configuration management tools can't stop someone from logging on to a device manually and making changes outside the scope of configuration management. In fact, instead of trying to prevent that from happening, you should assume it *will* happen, and act accordingly by providing policing functionality to catch these changes and deal with them appropriately.

Contribution to the Configuration Management Process

An important function of any process is to recognize and deal with external-process activity, such as manual changes to a device's configuration. Today's network management technologies make this possible in almost any environment, but you must have the right tools in order to make it happen. It's not sufficient to simply perform an automated scan of device configurations on a periodic basis; you need real-time monitoring that can catch changes before they have the opportunity to create major problems on your network. Ideally, you also need the ability to quickly undo those changes by rolling back to the last authorized configuration. The areas highlighted in Figure 6.18 illustrate where this functionality falls in the configuration management process.

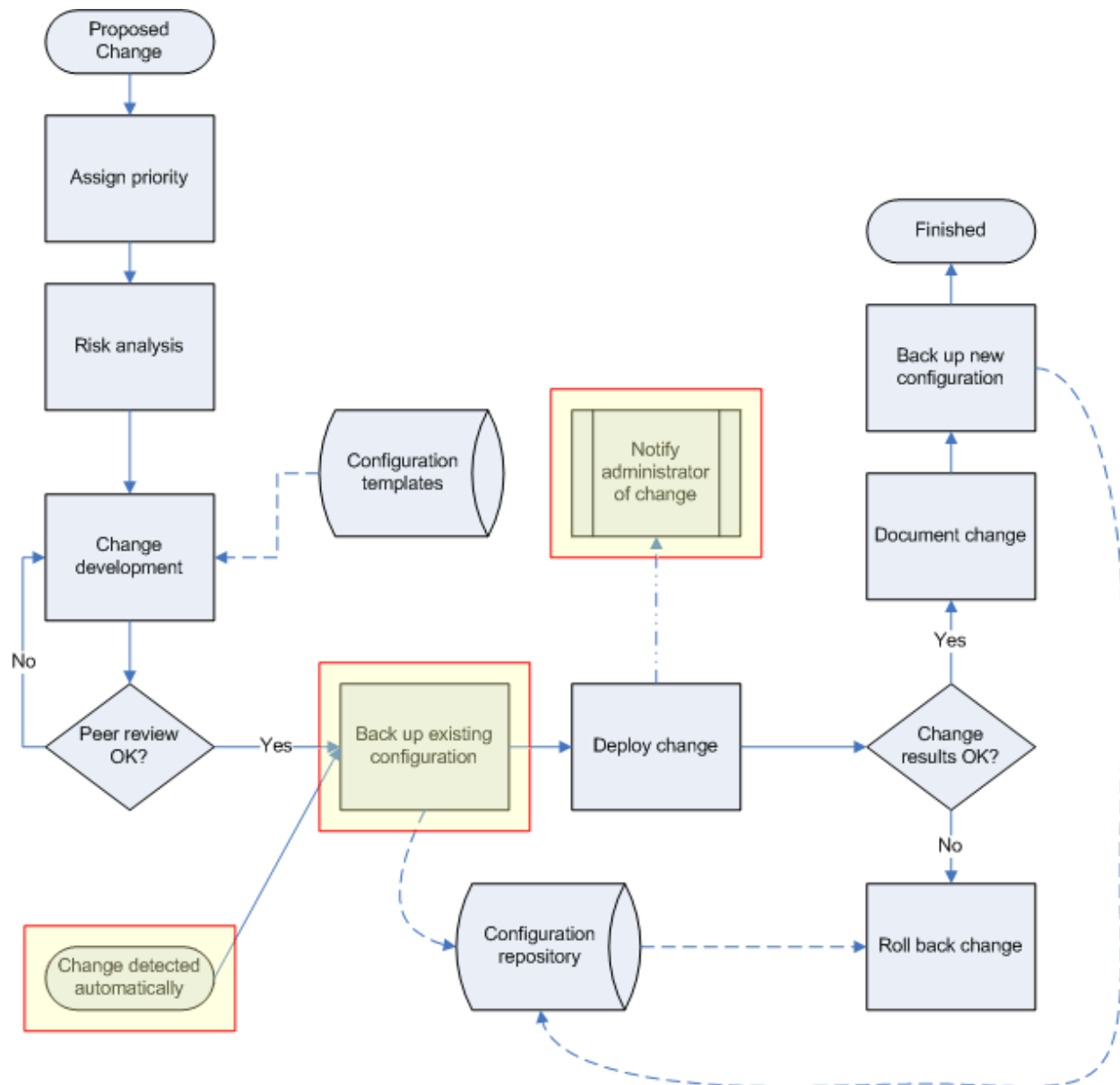


Figure 6.18: Ensuring compliance in the configuration management process.

Evaluation Criteria

Different vendor technologies provide different ways of dealing with real-time notifications; as a result, you need to select a solution that is vendor-neutral and flexible enough to use almost any system. Look for these features:

- Support for TACACS+, RADIUS, and Syslog—These technologies offer, in addition to other functionality, solid logging capabilities that can provide notification of trigger events. A trigger event is something—like an administrator putting a router into configuration mode—that indicates a change has taken place and tells your management solution to immediately poll the device’s configuration and look for unauthorized changes.
- Support for SNMP, an event protocol that can also act as a trigger event.
- Integration support for enterprise management frameworks—Because these frameworks also utilize TACACS+, SNMP, RADIUS, and Syslog, your configuration management solution must be able to work with the framework so that both systems can consume the events and logs provided without interfering with each other.
- Automated notification when an unauthorized change is discovered.
- Configurable automated corrective actions when an unauthorized change is discovered, such as automatically deploying the last authorized configuration back to the device, effectively undoing the unauthorized change.

Figure 6.19 provides a handy evaluation checklist for this functionality.

	Product	Product	Product	Product	Product
RADIUS support					
TACACS+ support					
Syslog support					
SNMP support					
Integration capabilities					
Automated notification					
Automated corrective action					

Figure 6.19: A tool evaluation checklist for the compliance and enforcement functionality.

Summary

In this chapter, I've described the various pieces of functionality that an enterprise configuration management process demands from a toolset. I've provided you with evaluation criteria and checklists so that you can evaluate existing and new tools that you're considering for the features necessary to implement a solid configuration management process.

In the next chapter, I'll spend some time exploring best practices for configuration and change management. I'll focus on the best practices published in the ITIL, a set of industry best practices that includes information about configuration management. I'll also describe how those ITIL best practices can be implemented in the real world and provide some sample process flowcharts to illustrate how it all fits together. Finally, we'll explore how some major consulting companies are beginning to provide best practice-based configuration management services, and how those can benefit larger organizations.