

These guidelines are based on the article "[Implement change management with these six steps.](#)" by Change Tech Solutions.

Change management deals with how changes to the system are managed so they don't degrade system performance and availability. Change management is especially critical in today's highly decentralized, network-based environment, where users themselves may be applying many changes. A key cause of high cost of ownership is the application of changes by those who don't fully understand their implications across the operating environment.

In effective change management, all changes should be identified and planned for prior to implementation. Back-out procedures should be established in case changes create problems. Then, after changes are applied, they are thoroughly tested and evaluated. Here are the process steps for change management and factors critical to its success.

Table of contents

Define change management process and practices	2
Receive change requests	2
Plan for implementation of changes	2
Implement and monitor the changes; back out changes if necessary	2
Evaluate and report on changes implemented	2
Modify change management plan if necessary.....	3
Evaluate and test all changes.....	3
Make sure all changes are covered.....	3
Document all changes	3
Communicate the benefits.....	3

1 Define change management process and practices

As you would with other systems management disciplines, you must first craft a plan for handling changes. This plan should cover:

- **Procedures for handling changes**—How changes are requested, how they are processed and scheduled for implementation, how they are applied, and the criteria for backing out changes that cause problems
- **Roles and responsibilities of the IT support staff**—Who receives the change request, who tracks all change requests, who schedules change implementations, and what each entity is supposed to do
- **Measurements for change management**—What will be tracked to monitor the efficiency of the change management discipline
- **Tools to be used**
- **Type of changes to be handled and how to assign priorities**—Priority assignment methodology and escalation guidelines
- **Back-out procedures**—Actions to take if applied changes do not perform as expected or cause problems to other components of the system

2 Receive change requests

Receive all requests for changes, ideally through a single change coordinator. Change requests can be submitted on a change request form that includes the date and time of the request.

3 Plan for implementation of changes

Examine all change requests to determine:

- Change request prioritization
- Resource requirements for implementing the change
- Impact to the system
- Back-out procedures
- Schedule of implementation

4 Implement and monitor the changes; back out changes if necessary

At this stage, apply the change and monitor the results. If the desired outcome is not achieved, or if other systems or applications are negatively affected, back out the changes.

5 Evaluate and report on changes implemented

Provide feedback on all changes to the change coordinator, whether they were successful or not. The change coordinator is responsible for examining trends in the application of changes, to see whether:

- Change implementation planning was sufficient.
- Changes to certain resources are more prone to problems.
- When a change has been successfully made, it is crucial that the corresponding system information store be updated to reflect them.

6 Modify change management plan if necessary

You may need to modify the entire change management process to make it more effective. Consider reexamining your change management discipline if:

- Changes are not being applied on time.
- Not enough changes are being processed.
- Too many changes are being backed out.
- Changes are affecting the system availability.
- Not all changes are being covered.

7 Evaluate and test all changes

Changes should be evaluated and tested prior to implementation. It is practically impossible to predict the outcome of all changes, especially in a complex, interrelated system architecture. You must carry out a thorough evaluation of all changes, especially those dealing with critical system resources.

We also highly recommend that you test all changes prior to full-scale deployment. For minimum impact on the system, test with a user not on the critical path, with test data, during off hours, and on a test system.

8 Make sure all changes are covered

Cover all changes, big and small. Minor changes can have major effects on system performance and availability. A simple change in a shared database's filename could cause all applications that use it to fail. An additional software utility installed in the user's workstation could cause the user's system to become unstable. Or a move of a user's workstation from one department to another could prevent it from properly accessing the network. You might occasionally need to bypass certain change management processes, such as in the case of emergency changes required to recover from a fault condition. But even in these situations, document the change thoroughly and have it approved after implementation to ensure that system records are updated.

9 Document all changes

Perhaps the hardest part of change management is documenting all actions performed before, during, and after the change has been applied. Technical people often fail to document changes, and we have seen many problems caused because not everyone knew about earlier changes. Many IT organizations are familiar with the Monday Morning Crisis—that most problems occur on Monday mornings because someone implemented a change over the weekend without following correct change management procedures.

10 Communicate the benefits

Many people mistakenly view change management as more IT red tape. They fail to realize that good change management acts like a traffic light that regulates the smooth flow of changes and does *not* stop all change from happening. With a well-planned and well-deployed process, you can ensure that changes do not negatively affect system performance as a whole.

The Enterprise Computing Institute (www.ecinst.com) helps IT professionals solve problems and simplify the management of IT through consulting and training based on the best-selling Enterprise Computing Institute book series. Founded by Harris Kern (www.harriskern.com), the industry's foremost expert on simplifying IT and world-renowned American author, publisher, lecturer, and consultant, the Institute has focused on providing practical guidance for tackling current IT challenges since its inception in 1998.

Additional resources

- TechRepublic's [Downloads RSS Feed](#) **XML**
- Sign up for our [Downloads Weekly Update](#) newsletter
- Sign up for our [CIO Issues Newsletter](#)
- Check out all of TechRepublic's [free newsletters](#)
- ["Keep documentation updated with change management"](#) (TechRepublic article)
- ["Changing things too late: Why change management feels like an uphill battle"](#) (TechRepublic article)
- ["Project Proposal and Special Clarification forms"](#) (TechRepublic download)

Version history

Version: 1.0

Published: September 26, 2005

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Content Team