

Implement Change Management with these Six Steps

Change management deals with how changes to the system are managed so they do not 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 do not fully understand their implications across the operating environment.

Change management means that 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. This article describes the process steps for change management and discusses factors critical to the success of change management.

Step 1: Define change management process and practices

As with other systems management disciplines, 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 what the criteria are 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.

Step 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.

Step 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

Step 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.

Step 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 if:

- 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.

Step 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

Other Process Issues

There are other process-related issues that critical to the success of change management. Changes are evaluated and tested prior to implementation. It is practically impossible to predict the outcome of all changes especially in a complex, interrelated system architecture. Hence, a thorough evaluation of all changes must be carried out, especially those dealing with critical system resources. We also highly recommend that all changes be tested 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, on a test system.

All changes, big and small, are covered. Minor changes can have major effects on system performance and availability. A simple change in a shared database's file name could cause all applications that use it to fail. An additional software utility installed in the user's workstation could cause his 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, as for emergency changes required to recover from a fault condition. But even in these cases, document the change thoroughly, and have it approved after implementation, to ensure that system records are updated.

All changes are documented. 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 — wherein most problems occur on Monday mornings because someone implemented a change over the weekend without following correct change management procedures.

Conclusion

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 impact system performance as a whole.