itSM Solutions® DITY™ Newsletter Reprint

This is a reprint of an itSM Solutions® DITY™ Newsletter. Our members receive our weekly DITY Newsletter, and have access to practical and often entertaining articles in our archives. DITY is the newsletter for IT professionals who want a workable, practical guide to implementing ITIL best practices -- without the hype.

become a member

(It's Free. Visit http://www.itsmsolutions.com/newsletters/DITY.htm)

Publisher
itSM Solutions™ LLC
31 South Talbert Blvd #295
Lexington, NC 27292
Phone (336) 510-2885
Fax (336) 798-6296

Find us on the web at: http://www.itsmsolutions.com.
To report errors please send a note to the editor, Hank Marquis at hank.marquis@itsmsolutions.com
For information on obtaining copies of this guide contact: sales@itsmsolutions.com
Copyright © 2006 Nichols-Kuhn Group. ITIL Glossaries © Crown Copyright Office of Government Commerce. Reproduced with the permission of the Controller of HMSO and the Office of Government Commerce.

Notice of Rights / Restricted Rights Legend
All rights reserved. Reproduction or transmittal of this guide or any portion thereof by any means whatsoever without prior written permission of the Publisher is prohibited. All itSM Solutions products are licensed in accordance with the terms and conditions of the itSM Solutions Partner License. No title or ownership of this guide, any portion thereof, or its contents is transferred, and any use of the guide or any portion thereof beyond the terms of the previously mentioned license, without written authorization of the Publisher, is prohibited.

Notice of Liability
This guide is distributed "As Is," without warranty of any kind, either express or implied, respecting the content of this guide, including but not limited to implied warranties for the guide's quality, performance, merchantability, or fitness for any particular purpose. Neither the authors, nor itSM Solutions LLC, its dealers or distributors shall be liable with respect to any liability, loss or damage caused or alleged to have been caused directly or indirectly by the contents of this guide.

Trademarks
itSM Solutions is a trademark of itSM Solutions LLC. Do IT Yourself™ and DITY™ are trademarks of Nichols-Kuhn Group. ITIL ® is a Registered Trade Mark, and a Registered Community Trade Mark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office, and is used here by itSM Solutions LLC under license from and with the permission of OGC (Trade Mark License No. 0002). IT Infrastructure Library ® is a Registered Trade Mark of the Office of Government Commerce and is used here by itSM Solutions LLC under license from and with the permission of OGC (Trade Mark License No. 0002). Other product names mentioned in this guide may be trademarks or registered trademarks of their respective companies.

itSM Solutions®
IT Experience. Practical Solutions.

Subscribe PDF Download Back Issues

Vol. 2.29 JULY 26, 2006

"IT -- Heal Thyself!"

DITY Weekly Reader

The workable, practical guide to Do IT Yourself

Most Service Desk calls result from failed changes, making IT its own worst enemy and largest customer. This makes IT the #1 preventable cause of IT service outages! The solution is Release Management...

hank MARQUIS

Articles
E-mail
Bio

By Hank Marquis

Various industry analysts claim that about 80% of Service Desk calls result from change-related failures self-inflicted by IT. By definition a failed change is a failure of IT, and they come at a massive cost.

The accepted cost per contact — answering the call and not including resolution, lost profit or productivity — is about $30 per incident. Since an average user calls the Service Desk 1.25 times per month a modest IT organization with just 4000 users pays an astounding $1,440,000 annually **just to answer the phone** for largely preventable change related failures.

Clearly, any IT organization looking to improve customer satisfaction and reduce costs should consider healing themselves first.

Release Management (RM) is an often misunderstood IT Infrastructure Library® (ITIL®) process. Simply put, the objective of RM is the protection of the live environment and its services by using formal procedures and checks.

Following I explain a how Release Management should work and how to create an effective Release Policy that can dramatically improve service quality and reduce costs.

Change, Release, and the CAB

It is important to understand the relationship between changes and releases in order to fully understand RM. A change is "The addition, modification or removal of approved, supported or baselined hardware, network

A change is "The addition, modification or removal of approved, supported or baselined hardware, network, software, application, environment, system, desktop build or associated documentation." A *release* is "*A collection of new and/or changed Configuration Items (CIs) which are tested and introduced into the live environment together."*

Change Management brings together interested parties via the Change Advisory Board (CAB) to makes sure a change is well thought out. The CAB is also responsible for advising and recommending release content and scheduling. Change Management and the Change Advisory Board (CAB) define and agree what changes to make to infrastructure components not yet introduced into production.

While many consider RM a subset of Change Management, it is a vitally important process in its own right. Release Management manages the introduction of the changed CIs into the live or production environment. Essentially, Release Management oversees the work required to implement the change.

As a member of the CAB, under Change Management control, RM is responsible for documenting the specific details of how to implement a change and making sure the change follows a stringent review process. Not all changes need to use formal RM, however you should always use RM for:

- large or critical hardware rollouts, especially when there is a dependency on a related software change
- major software rollouts, especially for new applications with new distribution and support procedures
- collecting related changes into manageable units

As a process, RM is responsible for implementing the release according to CAB directives (which, as a member of the CAB, RM helps define.) Without this tight connection between Change and Release Management, the success of the release and its associated change becomes haphazard.

Release Management In Action

Practitioners often get confused and try to create a functional organization for the RM process. RM does not define an organization, but rather defines what various organizations must do together as a team in order to modify the physical infrastructure. RM describes much more of a workflow than most ITIL processes.

As a process, the activities of RM execute within existing functional organizations, and normally span multiple organizational boundaries and technology silos. This means RM has both distributed and centralized process responsibilities and activities making it one of the most difficult processes to comprehend and implement for new practitioners.

Many problems with Change Management are really problems of RM, as described in the introduction. The "vicious" cycle here goes something like this:

- The CAB comes up with a plan
- Everyone is so busy and there are so many problems they skip essential parts of the plan
- No one follows up to make sure everyone is following the plan
- The Change then fails and generates Incidents
- The "fix" is the make a Change
- Repeat

The only way to break this cycle of failed changes is to make sure all parties follow the instructions of the CAB. This makes the real purpose of RM to prepare a release based on a Request for Change, and to make sure implementation of the release occurs as defined and agreed by the Change Management process.

Consider RM a supervisor of required work, and the secret to success with RM is to realize that it mostly defines oversight and describes a series of check-offs designed to validate completion of CAB and Change Management dictates. The next secret to success is to realize that the method and location of work completion is not as important as making sure the work completes, and that it completes according to the Release Policy and CAB instructions.

It is important to understand the difference between a *building a release* and *building software or hardware*. A release is the managed introduction of one or more changes into production as a unit, not the development of a piece of software or hardware. Part of the confusion with RM is that new practitioners confuse these concepts. This results in so-called "turf-battles" between RM and functional groups.

For example, part of RM includes responsibility for release "design, build and configuration." Of course, designing and building software is often a task assigned to a software or applications development group. Release does not "take over" development, but rather development must follow the Release Policy as they build the software. Then, a Release Manager (person following the Release Policy) checks to ensure the built application aligns with the Request for Change and Release instructions from the CAB.

RM must leverage existing tools and Auality Assurance best practices or methodologies already in place. Many software development groups use Capability Maturity Model (CMM) to manage software quality, since RM and CMM share several objectives there is no need to duplicate them. Instead, RM just needs to validate and "check off" that release criteria as described in the Release Policy, and instructions from the CAB, are complete -- in this case by examining CMM documentation in the development organization.

Release Management validates each step required, and submits confirmation to Change Management for review and approval. After Change Management approval, the release deploys. As the release deploys, RM updates Configuration Management with modified CI details.

Getting Release Management of the Ground

Getting started with RM means preparing the process – establishing an owner, manager, team, etc., and one of their first tasks is to define the Release Policy. The Release Policy defines how RM conducts its business within and between other processes and functional organizations. The Release Policy documents roles and responsibilities, and defines authorities. It describes how RM is to operate, and as such, it becomes an integral part of the Change Management process as well.

A Release policy should include:

Definition of the level of change to the IT infrastructure under RM control
Not every change requires creation of a formal release under RM control, so exactly what does constitute a release is an important element of the Release Policy. Consider changes to application software for example. Are modifications to a single file under RM control? Or do only modifications to more than one file require invocation and usage of RM? Setting the bar higher (fewer under RM control) is easier, lower the bar (more under RM control) as you mature and gain experience with RM. Remember that combined hardware and software changes, large software changes, and bundles of changes should always follow RM.

Release naming and numbering conventions
To prevent confusion and ease communications each release requires a sensible and easy to understand name. Many schemes are possible. Commonly the release will have a number related to the RFC date, or the date planned for the release, for example, YYMMDD. A release slated for July 27, 2006 becomes 060727. If there are many releases, inclusion of a sequence number is useful, for example, 60727-001. Some companies further expand this to include an identifier for the technology or type of the release, appending L for LAN, N for Network, A for Application, and so on. For example, 060726-001L for a release scheduled for July 26, 2006, for the LAN infrastructure. Some go even further, using an X to indicate a "patch" or "one off" release. For example, 060726-001LX to indicate a patch release to LAN infrastructure. Just remember to keep it simple and effective.

Release acceptance responsibilities
Many RM tasks occur within different functional departments. It is very important to define the title, position, and role within these distributed departments that are to perform these tasks. A central RM role (e.g., Release Manager) oversees completion of required tasks. The coordination and reporting between the distributed RM activities and centralized oversight is critical. Of course, this also requires a high enough level or management commitment to RM a priority within the distributed functional departments.

Definition of Major, Minor, and Emergency Releases
The ITIL mentions that Releases can comprise three classes, and that there is a relationship between them. Categorizing into these types helps reduce miscommunications and establishes a shared sense of release purpose. They are:

- Major Release -- contains new functionality and supersedes preceding Minor and Emergency Release.
- Minor Release -- contains enhancements and fixes, and supersedes preceding Emergency Release.

- Emergency Releases -- (or “fixes”) contain corrections to resolve Problems.

Release frequency
The expected future or annual schedule of Major and Minor Releases. This should take into account support capability, not just development ability to produce new or changed products.

Business and Customer concerns
RM has the responsibility to work closely with Customers (via the CAB or otherwise) to make sure that any planned releases do not occur during important periods. For example, taking down a print system for maintenance during the period when a key customer report is to print.

Release contents
Each release should have minimum and optional components. For example, every release should have notes that describe it. Optionally, a Major release might always require a set of instructions for installation. Document whatever a release must include to meet RM acceptance criteria.

Back-out plan policy
Key to successful changes is post implementation testing of a release, and if required, restoring the changed CIs back to their original state. The Back-out Plan policy defines which release requires such plans, and when and how to produce, test, and document them.

Release Management process description
It (almost) goes without saying that the Release Policy requires a complete description of the RM process itself. Be sure to include time and resource commitments from distributed functional departments and individuals who are to carry out RM tasks. Include meetings, audits, assessments, escalations, etc.

DSL and DHS
The Definitive Software Library (DSL) and Definitive Hardware Store (DHS) are vital repositories. RM must define, organize, and operate these repositories. The Release Policy must define where and how they reside, as well as the process for using and maintaining them.

Summary

I hope that you now understand the true role of RM, and have an idea of how it should operate. There are many tasks to RM, developing a Release Policy is just one of them. However, developing a Release Policy forces you to address virtually all of the other activities to some degree.

Key to your success is to realize that RM is a distributed process that relies upon your existing quality and production systems as much as possible. The central RM process tasks are supervisory – overseeing and checking off distributed RM tasks completed by the functional departments.

The steps of defining a RM process, creating a Release Policy, and making sure all involved do as they are supposed to do will have a dramatic and positive effect on operations. Reducing failed changes through controlling releases will improve service quality, and deliver real cost savings as well.

--

Where to go from here:

- Subscribe to our newsletter and get new skills delivered right to your Inbox, click here.
- Download this article in PDF format for use at your own convenience, click here.
- Browse back-issues of the DITY Newsletter, click here.

Entire Contents © 2006 itSM Solutions LLC. All Rights Reserved.