DITY™ Weekly Newsletter
itSM Solutions® IT Experience. Practical Solutions
The workable, practical guide to Do IT Yourself Vol. 4.49 • December 10, 2008

IT - Heal Thyself!

By Hank Marquis
Hank is EVP of Knowledge Management at Universal Solutions Group, and Founder and Director of NABSM.ORG. Contact Hank by email at hank.marquis@usgct.com. View Hank's blog at www.hankmarquis.info.

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...

[Editor's Note: This DITY about Release Management originally ran prior to the release of the new ITIL V3 guidance. Although some of the terms have changed in V3, the concepts remain unchanged. Substantive differences between the V2 and V3 guidance are noted within the text.]

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 $1,800,000 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 is an often misunderstood IT Infrastructure Library® (ITIL®) process. Simply put, the objective of Release Management 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 Release management. 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 make sure a change is well thought out. Change Management and the CAB define and agree what changes to make to infrastructure components. The CAB is also responsible for advising and recommending release content and scheduling.

While many consider Release Management 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, Release Management 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 *[ITIL V3 identifies 'Standard Changes']* need to use formal Release Management, however you should always use Release Management 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, Release Management *[ITIL V3 introduces the planning functions of Transition, Planning & Support and the deployment functions of Release & Deployment Management]* is responsible for implementing the release according to CAB directives (which, as a member of the CAB, Release Management helps define.) Without this tight connection between the Change and Release Management processes, 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 Release Management process. Release Management does not define an organization, but rather defines what various organizations must do together as a team in order to modify the physical infrastructure. Release Management describes much more of a workflow than most ITIL processes.

As a process, the activities of Release Management execute within existing functional organizations, and normally span multiple organizational boundaries and technology silos. This means Release Management 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 Release Management, 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 Release Management 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 Release Management a supervisor of required work, and the secret to success with Release Management 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.

It is important to understand the difference between 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 Release Management is that new practitioners confuse these concepts. This results in so-called "turf-battles" between Release Management and functional groups.

For example, part of Release Management 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.

Release Management must leverage existing tools and Quality Assurance best practices or methodologies already in place. Many software development groups use Capability Maturity Model (CMM) to manage software quality, since Release Management and CMM share several objectives there is no need to duplicate them. Instead, Release Management 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 *[ITIL V3 Service Validation & Testing]*, and submits confirmation to Change Management for review and approval. After Change Management approval, the release deploys. As the release deploys, Release Management updates Configuration Management with modified CI details.

Getting Release Management of the Ground

Getting started with Release Management 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 Release Management conducts its business within and between other processes and functional organizations. The Release Policy documents roles and responsibilities, and defines authorities. It describes how Release Management 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 Release Management control**
  Not every change requires creation of a formal release under Release Management 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 Release Management control? Or do only modifications to more than one file require invocation and usage of Release Management? Setting the bar higher (fewer under Release Management control) is easier, lower the bar (more under Release Management control) as you mature and gain experience with Release Management. Remember that combined hardware and software changes, large software changes, and bundles of changes should always follow Release Management.

- **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 Release Management 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 Release Management role (e.g., Release Manager) oversees completion of required tasks. The coordination and reporting between the distributed Release Management activities and centralized oversight is critical. Of course, this also requires a high enough level of management commitment to make Release Management a priority within 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 Releases
- Minor Release -- contains enhancements and fixes, and supersedes preceding Emergency Releases
- Emergency Releases -- (or "fixes") contain corrections to resolve urgent issues (aka 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**
  Release Management 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 Release Management process itself. Be sure to include time and resource commitments from distributed functional departments and individuals who are to carry out Release Management tasks. Include meetings, audits, assessments, escalations, etc.

- **DSL and DHS**
  The Definitive Software Library (DSL) *[ITIL V3 – Definitive Media Library (DML)]* is a vital repository. Release Management must define, organize, and operate this repository. The Release Policy must define where and how it resides, as well as the process for using and maintaining it.

Summary

I hope that you now understand the true role of Release Management, and have an idea of how it should operate. There are many tasks to Release Management, 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 Release Management is a distributed process that relies upon your existing quality and production systems as much as possible. The central Release Management process tasks are supervisory – overseeing and checking off distributed Release Management tasks completed by the functional departments.

The steps of defining a Release Management 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.