

Viewpoint – ITIL Change Management: When is a Change a Change?

A Management White Paper by:

William B. Husselbaugh



COPYRIGHT NOTICE

Copyright © 2008 by William Brett Hussenbaugh. All rights reserved.

This document is protected by the copyright laws of the United States of America. It may not be reproduced, nor may any of its contents used in whole or in part, without the express written consent of the author.

HIGHLIGHT

When considering ITIL Change Management, you can quickly become confused as to when Change Management applies. Does it apply to password changes, for instance? Do all change requests have to be documented by a Request For Change (RFC) and formally approved by the Change Advisory Board (CAB)? If so, that seems overwhelmingly inefficient. Is there some methodology that will be easier to justify yet still conform to the guidelines of ITIL?

DETAILS

Simply speaking, the purpose of Change Management, as defined by ITIL, is to ensure systematic and controlled change to Configuration Items (CIs). Anyone who has ever been on the receiving end of a typical IT “fire drill” knows first hand the impact that uncontrolled change can have.

Typical IT Change Example

Consider, for example, an uncoordinated change to close one port on a firewall in an effort to increase security. That one change can easily generate a call spike of incidents reported at the Service Desk as an entire group of application users – an application that happens to use the port that was closed – reports that they can no longer perform their job since the application they use is not working. This, of course, generates a multitude of responses from various support groups as the problem is triaged – is it an application error, or is the server down, or possibly a network segment or wide area link? The problem is traced on multiple fronts for as long as it takes until someone ultimately thinks of checking the firewall settings. Meanwhile, an entire business department is down and typically applying pressure to IT that increases with every passing hour. Put into perspective, that one small change, while well intentioned, created poor customer satisfaction for IT, required several unplanned resources to respond at high priority – undoubtedly putting work-in-progress aside – and created negative business unit impact that most likely resulted in customer dissatisfaction for the business as well as down time. And, since unplanned IT resources had to be committed at high priority, most likely other IT Service Level Agreements (SLAs) were put in jeopardy, resulting in poor IT customer satisfaction on other fronts as well. All because a network security expert thought it would be a good idea to tighten up the firewall a little more.

If this sounds familiar, then you will quickly see that Change Management is not an *impeding* discipline – something that stands in the way of progress. It is actually an *enabling* and essential discipline. Without it, the possibility of continued customer satisfaction and SLA achievement is

practically non-existent. With it comes the promise of increased customer satisfaction, reduced unplanned interruption to the business, increased IT efficiency and overall reduced cost.

When is a Change a Change

This, of course, is the real question. Per ITIL, a Request for Change (RFC) is generated to request change on any given CI. Keep in mind that a CI can be almost anything, and it can easily differ from a physical asset. For example, a CI could be a Dell Latitude laptop computer with the standard enterprise laptop image installed – that combination could be labeled a CI and could exist currently at baseline version 1.5.0. There may be many thousands of physical units that are managed under that one CI. Therefore, the requested change may affect a large population of physical units, each representing the current baseline of the CI. The change is then approved, or not, by the Change Advisory Board (CAB). In addition to approving or disapproving the RFC, the board may also prioritize various pending changes and determine the appropriate sequence and/or date of release. The composition of the board may vary according to the CI, the urgency of the change request, and the nature of the change request. The question is, do you really need to submit a RFC and route it to all members of the CAB for disposition for every conceivable change to a CI? For example, if a server is considered a CI, then does the changing of a password for an account on the server require an approved RFC? If security policy requires passwords to be changed every 90 days, wouldn't approval of the RFC just be assumed? Is there really a need to add to the CAB's workload, as well as the Configuration Management Database (CMDB) record count, just to track the potentially thousands of password changes in a year? The same question applies to password reset requests – arguably one of the major sources of Service Desk calls. Do they constitute a change or should they be considered incidents? How about swapping out a Read/Write Compact Disc (CD-RW) drive on a PC CI for a Read-only DVD (DVD-R) drive? Does that require an approved RFC?

Considerations for Determining Changes

Chapter 8, Change Management, in the ITIL book entitled “Best Practice for Service Support” mentions the need to *filter* change requests as well as a need to categorize change requests. There is certainly room for expansion and therefore the following considerations are offered:

1. Answer the question, does the requested change impact form, fit, or function of the target CI, or any CIs dependent on the target CI, or will the requested change apply to more than one unit of a CI? If the answer is “yes”, then the RFC must be routed to the CAB for disposition, regardless of how small or “low risk” the proposed change may seem.
2. Consider supporting multiple classes of RFCs.
 - **Class I** – any change request that will impact form, fit, or function of target, parent, or related CIs, or impacts more than one unit of a CI. This class loosely conforms to ITIL's “Major Impact” example category, but differs in that it only considers those changes that will cause material impact – whether small or large.
 - **Class II** – any change request that is not Class I, but will change the auditable state of the CI. The auditable state of the CI will be defined by the granularity of information tracked by the CMDB – the lowest level of information about the CI tracked by the

CMDB will determine what can be audited about the CI. This class loosely conforms to ITIL's "Significant Impact" example category.

- **Class III** – all other changes. This class loosely conforms to ITIL's "Minor Impact" example category.
3. Will the requested change impact one unit of an existing CI, a group of serial numbered units, or all units? If one or a group, what is the "waiver" process in order to allow units within the CI group to exist in an alternate state? A "waiver" process should specifically be considered for Class I changes that will be applied to only a subset of units in a CI. Also, can the CMDB handle documenting individual units in a CI in an alternate approved state (alternate to the CI baseline)?
 4. Will the requested change be applied to all units of the CI from the approval date forward, or will existing units be retrofitted with the change? This decision needs to be documented in the CMDB.
 5. Sometimes a RFC results simply from a user's lack of knowledge of the CI's functionality – asking for a change when the CI is currently capable of performing the requested function. Consider vetting these change requests via existing user-groups / communities. That is, have RFCs for specific applications or CIs originate from a users' group, after having been vetted by the group. This will cut down on unnecessary RFCs.

ITIL's examples for categorizing change are aligned primarily to the predicted impact of the proposed change and/or the magnitude of the resources required to affect the change. The classes recommended above are designed primarily to vet change requests in order to streamline how changes are handled – and avoid sending a majority of non-risk change requests to the CAB and/or CMDB for documentation. The recommended classes are designed to offer a quick methodology to determine if a change request must go to the CAB, creating a more streamlined path for those changes that do not require CAB approval. Theoretically, Class I changes pose the most risk to the organization while Class III pose little to no risk.

Each class definition should also pre-determine the extent of the RFC response. The pre-determined response for changes in each class should be documented in the Change Management plan. Also, typical change requests can be pre-categorized and documented in the Change Management plan (such as password reset requests), so that Service Desk agents can quickly spot Class III changes and can act to affect those changes without burdening the official Change Management process. As an example, consider the following:

- A Class I change must always have CAB approval to be implemented, must show proof of full regression testing prior to release, requires a release and test plan along with roll-back considerations, must pass a pre-release CM audit to compare "as-built" to "as-authorized", must pass a security audit, and requires full documentation in the CMDB and Definitive Software Library (DSL).
- A Class II change does not require full CAB approval, can be approved by the Change Manager, does not require full regression testing, does not require a release plan or roll-back, does not require a pre-release CM audit, may require a security audit (to be decided

by the Change Manager at approval), and does require documentation in the CMDB (but not the DSL).

- A Class III change can be authorized by the Service Desk agent and does not require documentation in the CMDB – essentially a “non change” from the perspective of Configuration and Change Management. It can be handled as an “incident” or a “change request” in terms of how it is classified at the Service Desk, but essentially is a non-change.

Therefore, applying the previous examples to the classes above, a password change may be considered a Class III change as it does not affect form, fit, or function of the CI or related CIs, applies to a single unit for each change event, and it does not change the auditable state of the CI. A single unit swap of a DVD-R drive in place of a CD-RW drive may be either a Class I or Class II change – depending on if a related CI (such as an application) depends on the Read/Write capability of the existing drive. A complete swap of the DVD-R in place of the CD-RW for all units comprising the CI is a Class I change as the proposed change impacts more than one unit. The single unit addition of a commercial off-the-shelf piece of software that is not on the current image would generally constitute a Class II change – it will not change form, fit, or function of the existing CI (assuming no interference in operation between the proposed package and the existing image – otherwise it might be classified as Class I), but it will most likely change the auditable state since installed software is likely being tracked by the CMDB.

Finally, for all Class I changes that are to be applied to multiple units, the CAB must decide on whether the change will be applied to all new units, or if existing units in the field are to be retrofitted. For Class I and Class II changes that apply to single units or groups of units within a CI, the CAB must explicitly document the unique identification numbers (serial numbers or some other unique ID) of the units to be targeted for the change. All of this should be documented in the CMDB to enable downstream audits of “as-authorized” to “as-built” states.

About the Author

Brett Husselbaugh has over 20 years of experience primarily in the IT industry. He has consulted with over 25 of the leading Fortune 500 companies on strategies for optimizing the IT investment. With experience as both a CIO and a CEO, Brett brings a unique and practical perspective to IT management, promoting the concept of operating as a "business within a business" to deliver measurable value. Brett is a proven business leader, an innovative thinker, a highly effective writer, and an enthusiastic and motivational public speaker.

Brett has experience as founder and CEO of TOBEK Technical Services, an IT Asset Management firm which he started with no outside investment and grew to 80 people in three years. He then positioned the firm and sold it to Inacom, a Fortune 500 company. Brett also has experience as a CIO, Managing Partner for Managed Services, VP of Strategic Development, VP of Services R&D, Principal Consultant, Industry Analyst, and Program Manager.

Brett has published several magazine articles as well as over 50 industry white and position papers. He has spoken on numerous occasions to audiences of senior and executive management teams on optimizing IT investment, developing strategy, and effective IT management.

Brett holds a Masters of Science in Electrical Engineering from the University of Texas at Arlington and a Bachelors of Science in Electrical Engineering from the University of Maryland at College Park. He is currently a member of American Mensa.

William Brett Husselbaugh
<http://www.husselbaugh.com>