

Viewpoint – ITIL Configuration Management: Choosing Configuration Items (CIs)

A Management White Paper by:

William B. Hesselbaugh



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

Much forward thought must be given when initially choosing CIs, in order to minimize the cost of the ensuing ITIL Configuration/Change/Release management practices, increase the tractability of implementation, and to minimize business risk.

DETAILS

Briefly, a CI is any entity for which change is to be controlled, documented, applied, and audited. A CI is not just a physical entity, such as a PC or monitor. A CI can be a database, an internally developed application or code module, a business process, or even a network.

Classifying an entity as a CI infers quite a bit of ongoing CM, Release Management, and Change Management activity. Therefore, CIs should be chosen carefully – don't just choose to make everything a CI. There are cost implications to consider.

A good rule of thumb to use when choosing CIs is to ask the question - "If uncontrolled change were to happen to this entity, would it negatively impact the business?" If the answer is "yes", then the entity is a good candidate to be classified as a CI. If no, then don't classify it as a CI. When working through choosing CIs, therefore, it is best practice to involve representatives of the business as they should be consulted in determining the business impact of uncontrolled change.

For example, if uncontrolled change were to happen to PC keyboards, would that cause negative impact to the business? The first and obvious question to ask is what change can be made to a keyboard – not a keyboard swap, because that would be a change to the PC – but the keyboard itself? If the answer is little or none, then classifying a keyboard as a CI might not be a good move as it will create unnecessary clutter in the Configuration Management Database (CMDB), and will require future audit against authorized state as well as authorized state and change tracking. These are all costs that should not be incurred because there is little to no threat to the business due to highly unlikely keyboard changes.

The same might be said for monitors and possibly even printers. However, the same most likely cannot be said for PCs. That is, if uncontrolled change were to happen to a PC, it is likely that it could negatively impact the business. Therefore, a PC is usually a good candidate to be classified as a CI. However, the combination of the PC and its image might be considered the CI, so that change is managed at the *combined* level, as opposed to classifying the PC by itself as a CI. This is a slightly different way of looking at the PC than a typical ITAM implementation (which tends to look at things as physical assets).

That concept – of considering a PC and its image as a single CI – shows how CIs can and should be chosen at various levels or hierarchies. For example, a network may have several PCs connected to it, which in turn tend to have several peripherals attached to them (as well as software loaded on them). ITIL CM requires the maintenance of several key relationships among CIs in the CMDB, including parent-child relationships. This is to allow the CM and Change Management practices to quickly identify all CIs that stand to be impacted by a requested change. Properly managing the inter-CI relationships is critical, and possibly time-consuming (depending on the capabilities of chosen toolsets.) This is another thing to consider when choosing CIs

Also consider the level in the system hierarchy at which change needs to be controlled, documented, applied, and audited. In the network example above, it may be at the PC-image level in the hierarchy – everything above that level will be considered a CI (that includes the network and servers and storage farms, for instance). Everything below will not be considered a CI (to include mice, keyboards, monitors, modems, etc.) Alternatively, it may be that the *network* is the lowest level at which change needs to be managed, and, in this case, PCs would not be considered as CIs. This may be a perfectly valid consideration for an environment of thin clients, or heavily “locked-down” PCs. The same concept applies as applied in the previous keyboard example – if change is not possible then there is really no reason to apply controls at that level. Consider, however, that it is entirely possible for ITAM and CM to diverge at this point – ITIL Configuration/Change/Release Management may be completely satisfied with choosing the network as the lowest CI, however that concept is very likely to be foreign to current ITAM thought.

The same applies to internally developed applications. Is the CI the entire application itself, or are individual modules that comprise the application considered CIs?

Choosing the CI level may also have release implications. If the entire application is the lowest CI chosen, then the entire application may have to be released each time a release is made. If it is possible and more cost effective to release individual modules, then this must be taken into consideration when choosing CI levels.

Peers in any given hierarchy should be considered independently when determining if they will be classified as CIs. For instance, from a PC’s perspective, the network may be chosen as the lowest level CI (meaning the PC is not considered a CI), however a server may be considered a CI – even though both appear and the same level relative to the network CI.

Another consideration when choosing CIs is to look at current Service Level Agreements (SLAs). Arguably SLAs are a strong indicator about what the business values. If there are SLAs specifically targeting PCs (return to service time SLAs, for instance), then PCs (or possibly the PC-image combination) must be considered CIs. However, it may be a great time to re-visit SLA choices, especially if the SLAs are contractually binding with a managed services vendor, as the independent analysis of where in the hierarchy a CI should be chosen may have created questions for the need for some SLAs. Synchronize these two activities when settling on CI choices.

Finally, consider that your choice in CIs has upstream implications – that is, everything above that level in the hierarchy will generally be classified as a CI as well. For instance, if uncontrolled change to the PC-image combination is thought to yield negative business impact,

then certainly the same goes for all levels above the PC-image combination. You should therefore attempt to choose the highest level in the hierarchy, below which you feel the risk to the business is acceptable for uncontrolled change. Be forward in your thinking. Although you can adjust your chosen level(s) later, there are obvious one-time cost implications for populating the CMDB and extending Change and Release management practices to the newly chosen level. However, you may want to consider tending toward the side of taking acceptable risk and minimizing, especially initially, the number of items chosen as CIs. This will help keep recurring costs down and, initially, make the implementation of ITIL Configuration/Change/Release management more tractable.

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>