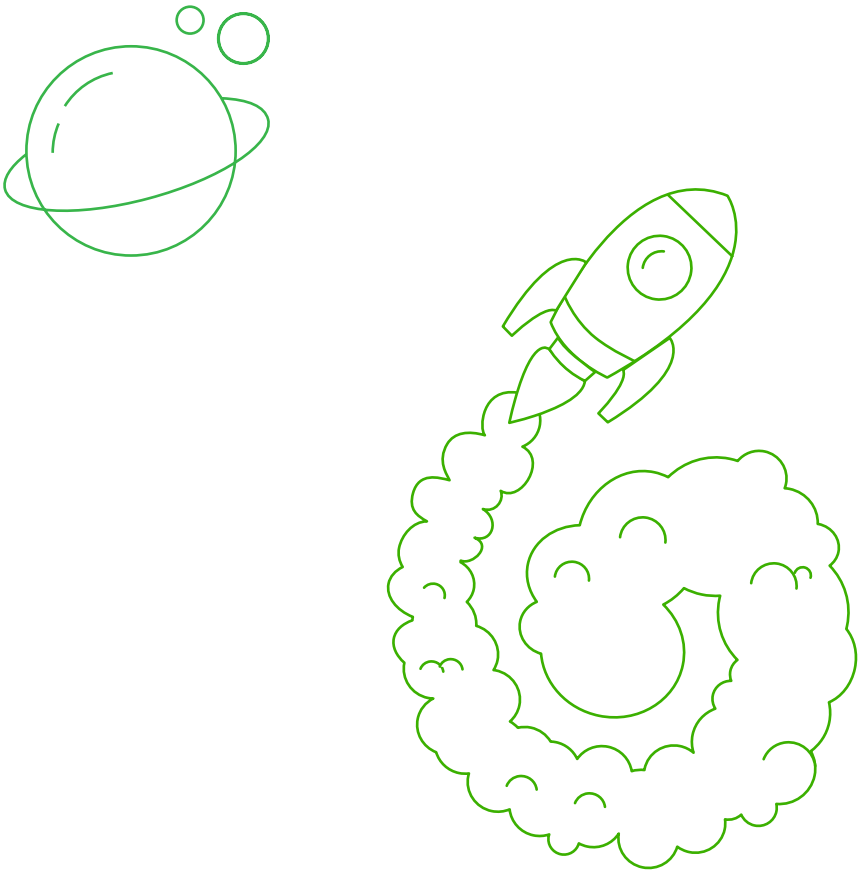
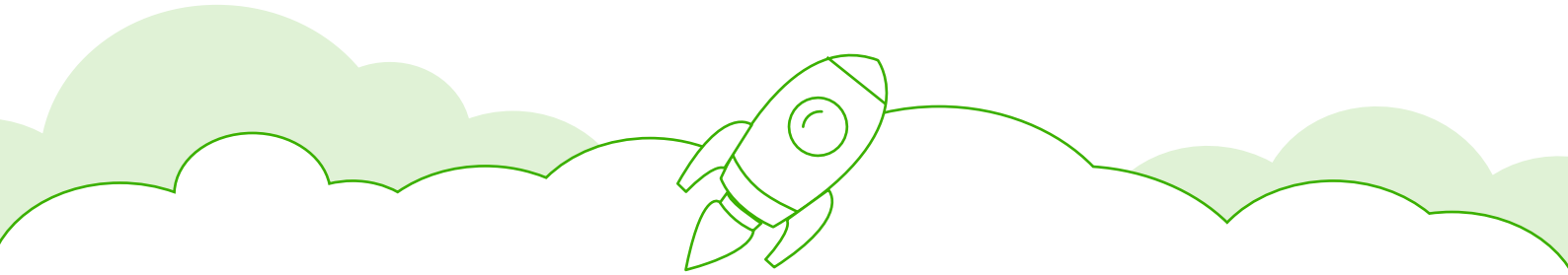
SysAid



6 Tips to Help You Improve Configuration Management

by Stuart Rance



Introduction

Configuration management provides information about what assets you own, how they are configured, and how they are connected together to support your IT services.

Most things that you do in IT service management (ITSM) have a direct and visible impact on your customers. For example, incident management helps you to restore service to users, and capacity management helps to ensure that services provide the throughput and performance that your customers need.

Configuration management is different. It delivers value to customers and users indirectly by maintaining essential data that is needed by other ITSM processes. This makes it really important for you to understand why you are doing configuration management and to focus on how it creates value. I have seen organizations, which do not have this focus, invest a huge amount of effort on time-consuming processes to maintain data that nobody actually uses – what I sometimes describe as a “write-only database”.

It’s very easy to get into a position where you focus on the process and the tool, instead of thinking about customers and service outcomes. This is particularly true if your organization has invested a lot of time and money in the tool and the supporting processes. When a situation like this is allowed to continue, it may damage the reputation of your IT and ITSM, and can eventually lead people to lose respect for all of your essential ITSM processes.

What Is the Purpose of Configuration Management?

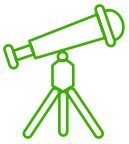
The purpose of your configuration management process is to collect and maintain relevant information about your assets, and ensure that this information is available when and where it is needed. This information about what assets you own, how they are configured, and how they are connected together to support your IT services is required for many different reasons.

Configuration management is important because almost every single IT activity makes use of some configuration information. So effective configuration management can improve how well you carry out every single ITSM process.

Here are some examples:

- Incident management can use configuration information to see what has changed, which is often the vital information needed to understand an incident. You can also use stored configuration information to check that components are configured correctly, and to help you to understand the impact of incidents, for example by showing which services run on a particular server or how many users are connected via a particular router.
- Problem management can use configuration information to see which components may be affected by a particular problem, and therefore to help prioritise problems. Problem management can also make use of configuration information when investigating root causes since it will offer insight into how components interact.
- Change management can use configuration information to help understand the risks of a change. Change management can use historical configuration information to help plan the rollback of failed changes and to identify unauthorized changes.
- Release management can use configuration information to help plan deployments. Detailed information about numbers and locations of components that require a particular release is essential for this purpose.
- Information security management can use configuration information to support audits and to help identify unauthorized modifications, which may indicate a security breach. It also requires configuration information to identify potentially vulnerable components. This requirement may be urgent when a supplier identifies a new vulnerability and you need to react quickly.
- Capacity management requires configuration information to help plan capacity-related upgrades.
- Financial management requires configuration information to ensure that an organization's valuable assets are properly tracked and managed.

I won't list every possible ITSM process here, but you may find it helpful to consider each of your processes to identify what configuration information is needed, and whether that information is currently available whenever and wherever the need exists.



Why Not Just Use a Discovery Tool When You Need the Information?

I have sometimes heard technical staff argue that they don't need configuration management, because they can use tools to find out configuration information when the need arises. This, they say, provides more accurate information, with less effort. There is some truth to this, and you shouldn't just ignore these claims. Instead, make sure you understand what information you need to store and maintain, and what can just be collected when it is needed, and then design a configuration management system that uses both of these approaches as appropriate.

Here are some reasons why just running a discovery tool to collect data when it is needed may not be sufficient.

- IT owns many valuable assets that must be properly protected from loss or theft. A discovery process will tell you what is currently connected to the network, but if you need to carry out an audit to compare the assets you have with the assets you should have, then you also need the configuration information.
- If you are investigating an incident then you may need to compare the current configuration with the configuration when the service was working properly. A configuration management system can maintain a complete history so that you can see what changed, and when, and this information can be essential in helping to understand incidents.
- Comparing the stored configuration information to the actual configuration can help you to identify unauthorized changes. This can help to reduce the number of unauthorized changes, resulting in fewer unplanned outages, as well as helping you to identify unauthorized changes that do happen. A configuration management system can also provide the information needed to recover back to the original situation after a failed change, when needed.
- If you are planning a hardware or software upgrade then you may need detailed information about the configuration of portable assets that are not always connected to the network, such as laptops or tablets. A simple discovery tool cannot get complete information about how all portable devices are configured, unless it is used in conjunction with a configuration management system that stores and maintains this information.

Some Definitions

There are a number of terms used in configuration management, and it is important to distinguish these. Here are some terms that you will find used in this white paper, with their definitions. These definitions are based on the definitions in the ITIL Glossary, but have been modified for use in this document.

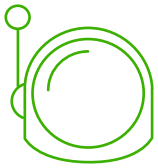
Configuration Item (CI)
A configuration item (CI) is anything that needs to be managed in order to deliver an IT service. For example: a server, an application, a network router, a software license, or a contract with a vendor. An IT service is also a type of CI.

Configuration Management Database (CMDB)
A configuration management database (CMDB) is a database that stores configuration records. Each configuration record in the CMDB stores information about one CI. The CMDB stores attributes of the CIs, for example: the name, size, type, or model number. It also stores relationships between CIs, for example which servers are connected to a particular network segment, or which service owner is responsible for a particular IT service.

Configuration Management System (CMS)
A configuration management system (CMS) is the complete set of tools, data and information that are maintained by configuration management. The CMS includes tools for collecting, storing, managing, updating, analysing, and presenting data about all CIs and their relationships. The CMS may include many different tools and CMDBs.

A well-designed configuration management system can provide significant value to your organization; here are some tips to help you make sure your configuration management system delivers this value.

Tip# 1



Focus Configuration Management Design on Use Cases

Many organizations design configuration management based on data that they think is needed, or that they think is important. Unless there is a plan for someone to actually use this data, then this can easily turn into the situation I have described as a "write-only database", where data is collected and updated, but never used to create any value.

The best way to avoid configuration management becoming a write-only database is to focus the design on how the data will be used. To do this you should:

- Identify all of the different stakeholders in your organization that may need to use configuration information.
- Work with the stakeholders to understand what information they need to do their jobs, how they could use the information, what format they would find most useful, and when and where they need to access the information.
- Document these uses of configuration information as use cases. Each use case should describe one way in which the information will be used to create value for one or more clearly identified stakeholders. Discuss these use cases with the stakeholders to ensure that they are appropriate for their needs.
- Base the design of your configuration management tools and processes on these clearly defined use cases, and carry out tests to ensure that each use case can be met by your solution.
- Train IT staff in how to use the configuration management system based on the agreed use cases.
- Carry out regular reviews to ensure that the users are actually making use of the configuration data in the ways that you expect. If necessary, update the use cases and modify your design so that it continues to meet the needs of your stakeholders.

Tip# 2

Don't Just Store Technical Information

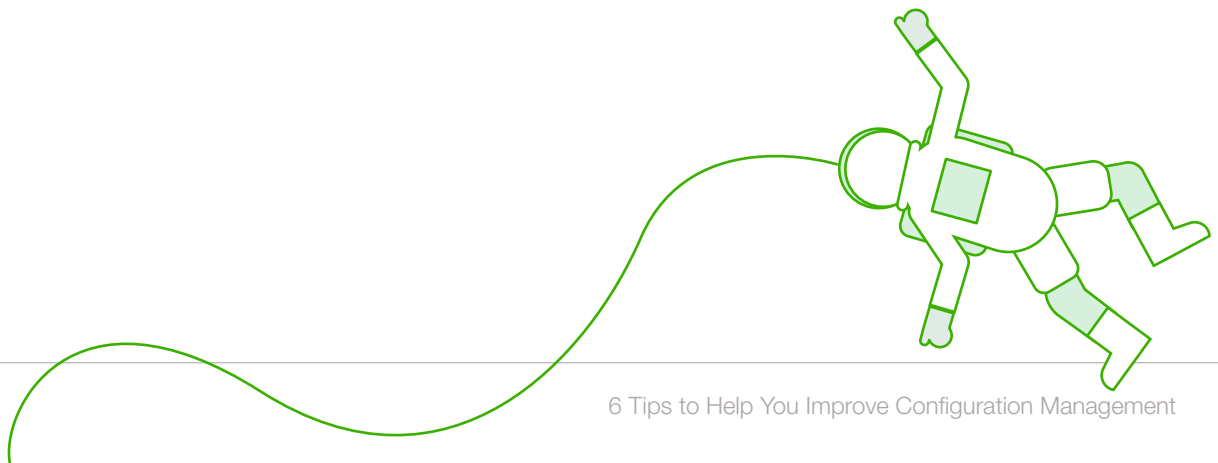
If you base the design of your configuration management system on discovery tools, then it is quite likely that all of your configuration information will be about infrastructure and applications. To be really useful a configuration management system should also include information that can't be collected with a tool. Examples of this type of data include:

- Who owns each asset
- Who can approve changes to the asset
- Who can approve new or changed access rights to the asset
- Who uses the asset
- What services the asset is part of
- What maintenance contract is in place to support the asset

As you define use cases, remember to consider all the information you might need, not just the technical information that is easy to collect with tools. For each piece of information that you need you should think about:

- How will you collect this information?
- How will you verify that this information is correct?
- How will you find out about updates or changes to this information?
- How will you audit this information to ensure that it remains correct?

A configuration management system that is based on use cases, and that provides all the information that is needed for each use case, not just the technical configuration information, can be extremely valuable.



Tip# 3

Don't Collect More Configuration Information Than You Need

There is a great temptation to collect and store all of the information that you possibly can, because it might be useful at some time in the future. This tends to happen when the design of the configuration management system has been based on the data collection capabilities of the selected discovery tools, rather than on the needs of the organization.

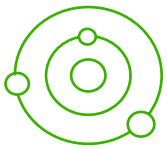
Collecting, storing, and managing too much data can lead to large and unwieldy configuration management databases, with costs that spiral out of control, and that deliver very limited value.

A much better approach is to identify the configuration information that you actually need, and to just store and maintain this information. Even if your tools are capable of collecting much more information, there is no need to do so unless you have a use for it.

There are two implications of this tip:

1. You should only consider an asset to be a separate CI if it needs to be separately managed. For example, you may need to have a CI for every disk drive if you manage each of these separately, but it is usually sufficient to have one CI that represents the server and to have the number and types of disk drives as attributes of that CI.
2. You should only record something as an attribute or relationship for a CI if it is information that you need to manage the service. For example, you probably don't need to record the unique serial number of every computer mouse and keyboard, even if this information is available to you, as these items are usually considered to be "consumable" and recording the serial number has no value. You may however need to store the serial numbers of expensive or unusual keyboards and mice that you need to manage in a different way.

Tip# 4



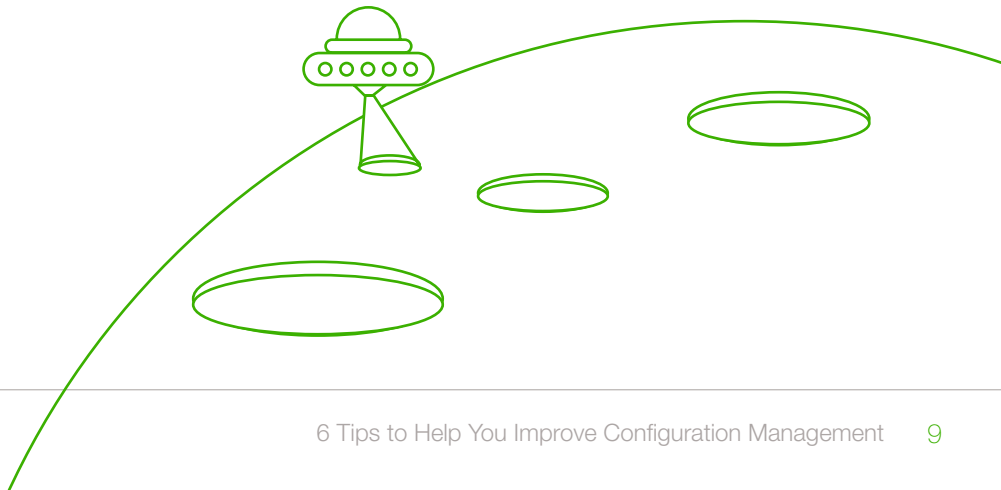
You Can Have as Many Configuration Management Databases as You Need

Your configuration information needs to be stored in a way that enables staff to access the information they need, when and where they need it. It should also provide the protection required to prevent inappropriate access, which might introduce security vulnerabilities, or potential leaks of personal data, or allow unauthorized modifications. Your CMDB should provide the access controls needed to enable this protection.

A CMDB will often be provided as part of an ITSM toolset, along with discovery tools to collect configuration data and populate the database. Some people believe that all configuration information for the entire organization should always reside in a single CMDB. At best, this is an idealistic view of how configuration management should work, and is often entirely inappropriate. It is important to have a single CMS, but within this CMS there may be a need for many tools, and many CMDBs; it depends on the nature of your organization, your infrastructure, and your IT services. The decision about how many CMDBs to create is not a strategic one, it is simply a matter of understanding the best way to collect, store, and manage the data that you need.

If you have more than one CMDB, then your CMS design will need to consider the following issues:

- How will people be directed to the right location to find the data they require?
- Do you need to federate the CMDBs to allow a single view or are they independent?
- Do you need to support cross references so that a CI in one CMDB can have a relationship with a CI in a different CMDB? If so, then how will you enable this?



Tip# 5

Integrate Configuration Management with Other Processes and Data

The configuration information that is stored in your CMDBs won't provide any value until you make use of it. This information isn't used by configuration management, which is simply the process that maintains the information. Rather it is used by a wide range of other ITSM processes.

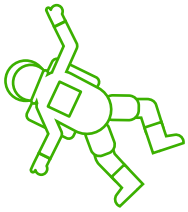
It is important when designing any ITSM process to consider how it will interact with other processes and activities within yourorganization, but this is even more important for configuration management. This means that you cannot design configuration management by itself, the design of your configuration management process needs to be considered as you design all of your other ITSM processes. Configuration management should be designed so that it supports the way these processes work, but also these processes may need design changes to support what is feasible, and cost effective for configuration management. For example, your incident management process may need to access information about users, so that when a user logs an incident the service desk agents are aware of where that user works and what devices they normally use to access the services. The incident management process and the configuration management process both need to be designed to support this need.

Your organization will store much information and many records that are not part of the CMDB, but that may need to be associated with information in the CMDB. For example, it is essential that incident records, problem records, and change records are associated with CIs, and that the history of any CI can be quickly established by following links from the CMDB.

There may also be a need for configuration information to be shared outside of the IT organization, and for external information to be made available within IT. Examples of this include:

- A financial asset register that stores information about costs and depreciation of organizational assets. Some of these assets are also configuration items, but others may have nothing to do with IT. Information such as the owner, purchase date, current status, and location may need to be shared between financial management and IT.
- Human resource records that store information about employees and contractors. If your CMS includes information about IT staff and their competencies, or about users and their contact details, then these may need to be shared between HR and IT.

Tip# 6



Report on Business Value, Not Technical Content

When I talk to customers about their measurement, reporting, and improvement activities, I am often very disappointed when I see what they consider important for configuration management. The vast majority of configuration management measurements and targets are very internally focussed, and are about the completeness and accuracy of the configuration information. This may be important, but it is not nearly as important as the things these organizations should be measuring and improving.

One of the best methods I know for improving targets like these is to ask "so what", and to keep asking that until you hear about something that creates value for the customer. For example:

"We measure and report 'Percentage accuracy of CI information during annual audits'."

"So what?"

"Accurate configuration information is important because people use this information when they are solving incidents."

"So what?"

"If the configuration information is wrong, then it causes delays in incident management."

"So what?"

"Delays in incident management cause increased cost to the business and to IT, and result in reduced ability to service our end customers."

Now, we have come to the important fact that we can think about how to measure. Maybe we should be measuring "number of incidents impacted by incorrect or unavailable configuration information", or "total business impact of incident management delays caused by incorrect or unavailable configuration information".

This was just one example. The important thing to note is that you really need to measure and report the things that impact business performance and that matter to your customers. If you are currently using internal metrics that just consider configuration management itself, then ask yourself "so what" and keep asking until you get to something that really matters. Then you can define metrics and KPIs that are worthwhile, and focus your improvement activity where it will have the greatest value.

Talk to us