

SECURE YOUR BUSINESS WITH ROBUST IT CONFIGURATION MANAGEMENT

Abstract

The performance of IT service delivery can either make or break a business. For large enterprises, efficient IT services are critical for smooth business functioning without outages or failures. An important element for efficient IT service delivery is the configuration management database (CMDB), a repository of all IT services, applications and IT infrastructure components. An effective CMDB can help enterprises streamline IT service management, reduce IT maintenance cost, and improve service availability, performance and customer satisfaction. It also helps in achieving faster business growth.

This paper provides a roadmap for organizations to build the CMDB. The process mentioned here can be used for either greenfield CMDB implementations or to migrate legacy CMDB systems to an enterprise-wide CMDB system.



Introduction

In today's digital world, efficient IT service delivery is crucial to sustain the business of any organization. IT services support business processes as well as any business transformation initiative. One of the aspects of smooth IT service management (ITSM) is the configuration management database (CMDB). The CMDB stores all the IT services, applications and service components such as servers, databases, network components, etc., along with their requisite details (attributes) that are needed to deliver enterprise IT services. The CMDB also captures the relationships between each service, application and service component. In cases of application or component failure, it is the CMDB that helps analyze the cause as well as the business impact. As a repository of business-critical data, it also helps assess the overall impact of changes across the enterprise so organizations can better plan implementations. Thus, an effective CMDB can slash business costs by reducing the cost of managing IT services and ensuring high quality (less disruption and loss of productive hours). It also boosts IT service performance, yielding higher customer satisfaction and business growth.

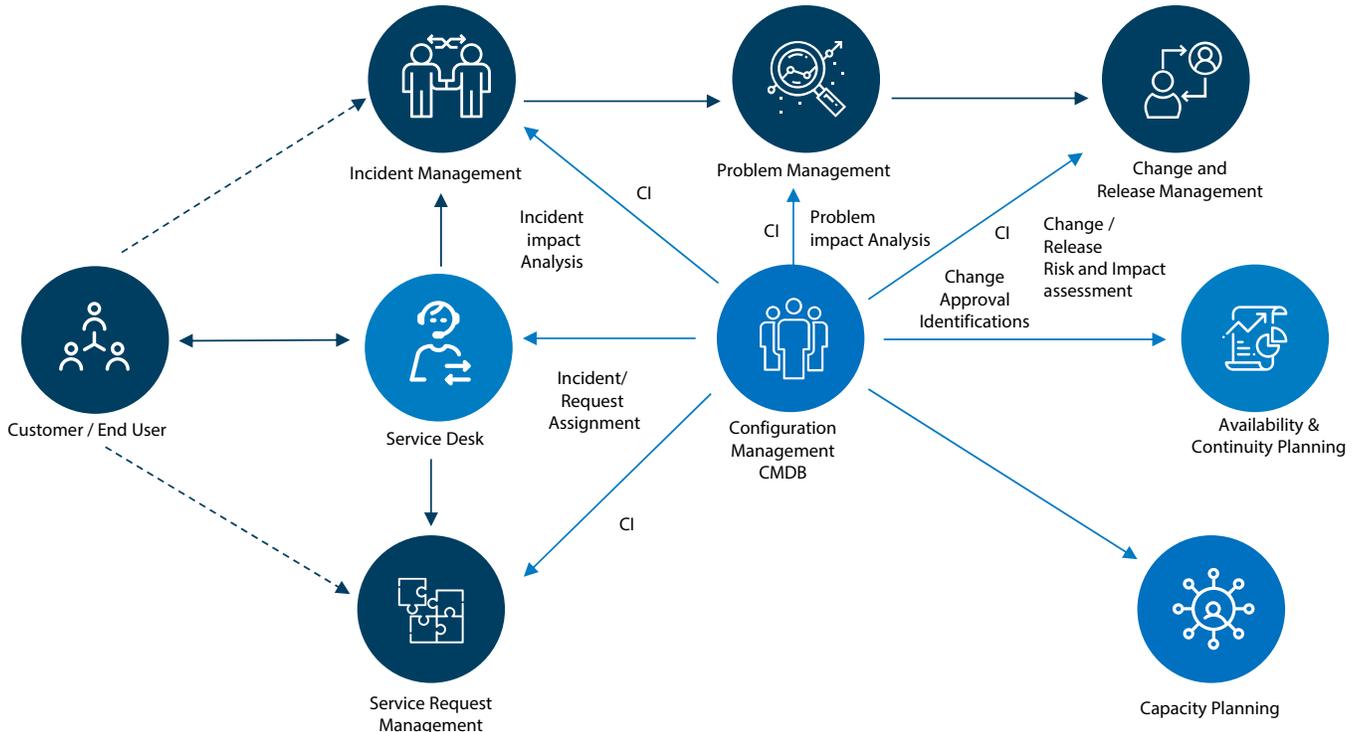


Fig 1: CMDB – Heart of ITSM architecture

Challenges of implementing CMDB

The information technology infrastructure library (ITIL) puts CMDB at the heart of ITSM processes. However, ITIL does not provide instructions on how to implement the CMDB for an enterprise. This, coupled with the lack of industry-standard processes, procedures or guidelines for CMDB implementation, makes it extremely challenging for enterprises with a large number of configuration items (CIs).

The topmost questions for IT departments when it comes to CMDB are:

- How do we implement CMDB for the enterprise?
- What is the first implementation step?
- How do we effectively migrate data out of legacy systems?
- What CIs should be considered for the CMDB?

The following section explains how enterprises of any size can build their own CMDB independent of any specific ITSM tool.

How to build your CMDB

Broadly, building your CMDB involves the following steps:

- Identify key stakeholders and perform data analysis
- Limit the CMDB scope
- Perform CI classification and define basic CI classes
- Identify CI attributes in future systems and map the attributes
- Define CI relationship and CI relationship types
- Design the CMDB blueprint
- Configure the CMDB and migrate data
- Ensure CMDB maintenance

1. Identify key stakeholders and perform data analysis

Enterprises should begin by analyzing their governance and delivery models. They should identify the various business areas/units and key stakeholders in each area along with representatives of partners or suppliers in case of multi-sourcing scenarios. Collaborating with stakeholders will provide vital insights when assessing the current state. It is important to choose stakeholders who have the right knowledge about systems and business processes. These individuals should also possess the requisite decision-making capabilities to finalize the scope of the CMDB.

Once the stakeholders have been identified, conduct interviews or workshops for a 'current state data analysis'. The current state data analysis helps enterprises:

- Determine data in all legacy data sources
- Identify the data type available in multiple data sources. CIs must be classified into different data categories such as whether a particular CI is a service, application, server, or network component
- Identify CI attribute details with each data type
- Understand CI relationship details, i.e., how one item is connected to others

It is also important to have a joint workshop with all key stakeholders at the end of the assessment to validate the information and avoid differences in opinion. The key points discussed during all meetings and workshops must be captured through a current state data analysis report and shared with all attendees and stakeholders.

2. Limit the CMDB scope

Most organizations are unaware of the type of CIs to be included in the CMDB. This step 'Limit the CMDB scope' identifies relevant CI types for the target CMDB. Stakeholders should establish and define the scope of boundaries for configuration management. When including items for the CMDB, enterprises should:

- Determine the importance of each CI to the business process and consider only those items that are needed for effective service delivery. Consumable items like laptops, desktops, mobile phones, etc. (which are not directly related to the service offering) are not recommended to be captured in the CMDB
- Consider CIs that are needed to fulfil data standard compliance, enterprise policy adherence, audit compliance, etc.
- Make sure the business value gained by capturing the CI in the CMDB is greater than the cost of managing the same. Organizations need to allocate funds for governing and managing the components in CMDB (for example, discovery licenses should be procured to ensure CI attributes are up-to-date)
- Ensure that all CIs have a valid owner who is responsible for managing the CI details in the future. Orphan CIs should not be considered for the CMDB
- Consider only those items as CIs that can be controlled via change processes
- Consider the geographical scope and environment to be controlled
- Consider the CI details required to support other ITSM processes like incident management, change management etc.

Ultimately, the final decision-makers must be the stakeholders who know the CIs. A useful tool here is the data analysis report. It helps stakeholders justify which CIs must be moved and why. CIs that are not considered for the target CMDB must be listed as 'out of scope'. Finally, all decisions must be documented and approved by the stakeholders.



3. Perform CI classification and define basic CI classes

Each CI type is considered as an 'entity'. Similar types of CIs from multiple data sources should be grouped together. For example, applications gathered from different data sources can be grouped under the 'application' entity. Entity details and the data sources must be documented. Each CI entity will also be associated with basic CI classes. It is a good practice to keep CI classifications as simple as possible. Some of the CI classes in the ServiceNow data model include business processes, services, service offerings, applications, servers, and network gear.

4. Define attributes for each CI class and map the attributes

Determine the attributes that are required for each Configuration Item class.

CI attributes for each entity/class collected from different sources (and documented in the current state data analysis report) must be analyzed to determine the importance of CI attributes to the business as well as service delivery. This will help delineate the attributes of entities in the source system that should be migrated to the new CMDB.

For CMDB migration, the attribute/field details for the new CMDB may not be similar to those of the source systems. Stakeholders should identify the fields in the target CMDB to accommodate the

legacy system attributes and work on the attribute/field mapping between legacy systems and the new CMDB. In cases where it is not possible to map some of the attributes from the legacy systems to any of the fields in the target CMDB, custom fields can be created. However, as a best practice, we should avoid creating custom fields. All field mapping-related decisions with business justifications need to be documented along with the reasons for creating custom attributes (if any are created).

Sometimes it may be necessary to transform, enrich or clean data during migration from legacy data sources to the target CMDB. In such cases, data cleansing, transformation or enrichment activities must be documented as well.

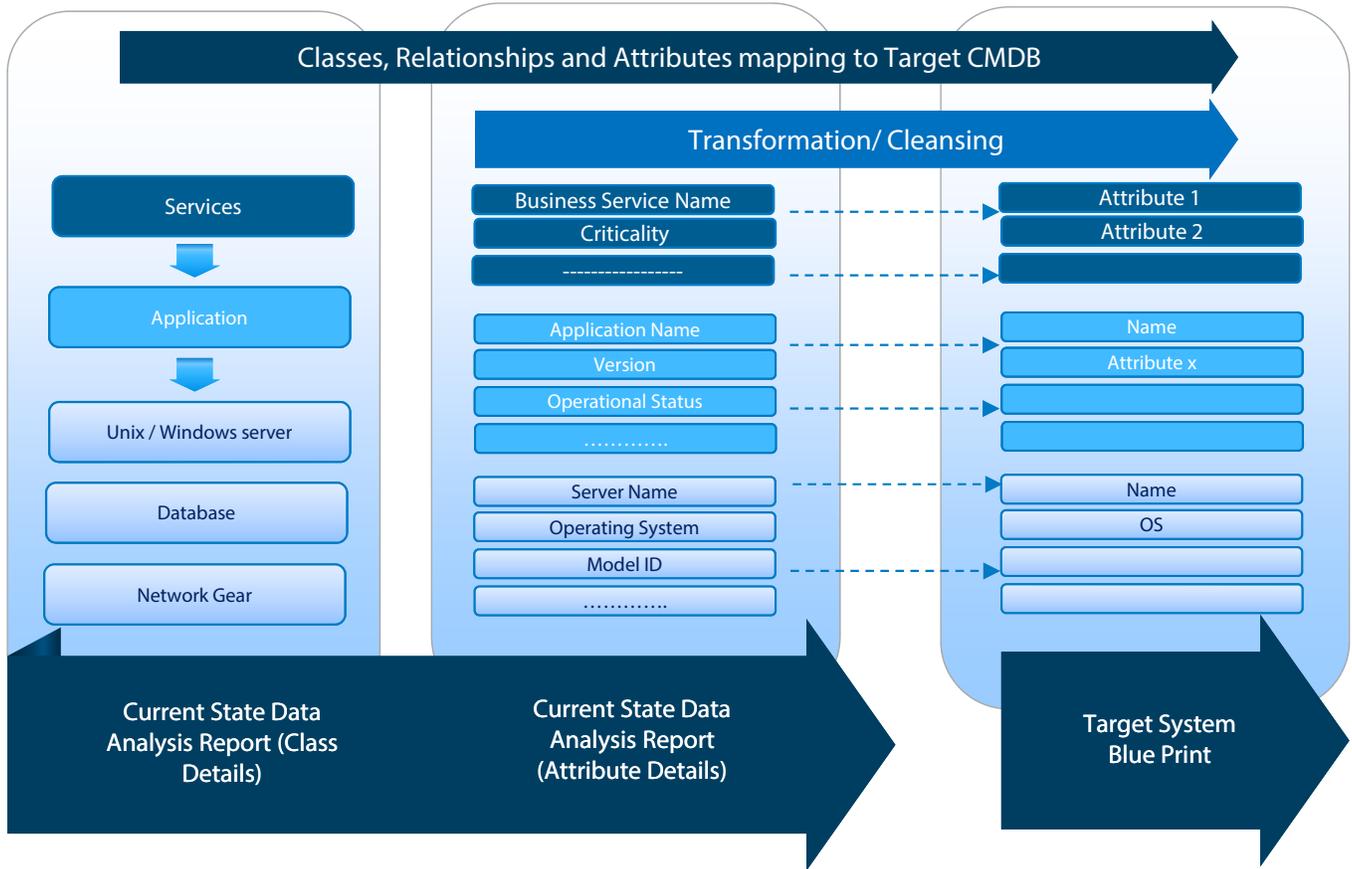


Fig 2: Attribute mapping details

5. Define CI relationship and CI relationship types

Determine the Relationships that are required between Configuration Items

The key steps here are to analyze the existing CI relationship details, design the target CMDB CI class relationships, define the relationship types, and define the relationships among CIs to users, groups, etc. CI relationship details (as recommended by ServiceNow) can be documented as shown below:

Base class	Relationship	Dependent class
Business service	Depends on : Used by	Application
Application	Runs on : Runs	Server
Application instance	Depends on : Used by	Database instance





6. Design the CMDB blueprint

In this step, enterprises should design the CMDB blueprint, a key priority for the enterprise CMDB. The CMDB blueprint should contain the following information:

- Details of all CI entities in the environment
- Relationship details among CI entities as well as a complete view of the relationships between all elements, i.e., how they are tied together and how they impact other elements
- Details of fields and attributes against each CI type
- Details of custom attributes created (if any)

This document should be meticulously used to update the CMDB in the future. It should be controlled through proper change management processes.

7. Configure the CMDB and migrate data

Here, enterprises must design and configure the new CMDB as per the blueprint. The process begins by loading some sample data in a couple of iterations (following agile methodologies) and involves appropriate stakeholders to verify the data. If changes need to be made, the blueprint document should be updated accordingly. Next, all the required normalization, reconciliation and data cleanup activities need to be performed while migrating the data into the CMDB. Enterprises should also ensure integration with other sources (if any) to automate the data feed.

8. Ensure CMDB maintenance

Once the CMDB is ready with the relevant CIs and corresponding data, enterprises should decide on a strategy to maintain the CMDB. Here, it is important to ensure that

the CMDB is always in sync with the actual business environment. Below are some best practices for CMDB maintenance:

1. Define the right process to update the CMDB including adding or deleting a CI. Updating the CMDB should be controlled (wherever required) by change management processes
2. The configuration management team should be responsible for updating critical attributes of the CI while other attributes can be updated by the CI owners
3. Discovery can be used to update the CI attributes in the new CMDB
4. Create CMDB Health Dashboards to facilitate control of CMDB and activities

While performing the health check ensure that all the attributes required for service delivery are available. There should be no duplicate or orphan CI existing in the CMDB.

Conclusion

Today, IT is critical to business growth. As IT service management (ITSM) becomes a key business driver, organizations must improve the efficiency with which they consume IT services. Failures and outages compromise service delivery and can lead to negative business impact. CMDB is an important component of the enterprise IT infrastructure library (ITIL). However, many companies are unaware of how to build or implement CMDB. The right implementation approach must be carefully planned with clear steps that include identifying stakeholders, limiting the scope, classifying and mapping the CIs based on their attributes and relationship types, designing the blueprint, configuring the CMDB, migrating data, and maintaining the CMDB. While the above guidelines help enterprises build their own CMDBs, one or more project managers must be assigned to manage the implementation based on the size and complexity. Organizational change management (OCM) is a useful strategy to spread awareness about the new CMDB and its usage to streamline adoption. With this approach, organizations can benefit from IT service assurance, reduced IT maintenance cost and higher business growth through greater user and consumer satisfaction levels.



About the author



Supratik Dey

Supratik Dey is an ITIL v3 expert with Infosys Technologies Limited. With nearly 15 years of experience in areas such as ITSM consulting, ITSM transformation and service governance, he has enabled several CMDB implementation programs for large enterprises.

For more information, contact askus@infosys.com



© 2019 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.