



Step-by-Step Guide to **Building a CMDB**

Updated for ITIL® Version 3

Published by BMC Software

Step-by-Step Guide to **Building a CMDB**

©2008 BMC Software, Inc. All rights reserved.

Published by BMC Software
ISBN 0-9778119-1-3

About BMC Software

BMC Software delivers the solutions IT needs to increase business value through better management of technology and IT processes. Our industry-leading Business Service Management solutions help you reduce cost, lower risk of business disruption, and benefit from an IT infrastructure built to support business growth and flexibility. Only BMC provides best-practice IT processes, automated technology management, and award-winning BMC Atrium technologies that offer a shared view into how IT services support business priorities. Known for enterprise solutions that span mainframe, distributed systems, and end-user devices, BMC also delivers solutions that address the unique challenges of the mid-sized business. Founded in 1980, BMC has offices worldwide and fiscal 2008 revenues of \$1.73 billion. Activate your business with the power of IT.

www.bmc.com

Table of Contents

| | |
|--|------------|
| Introduction..... | III |
| Acknowledgments | VII |
| STAGE 1 Assemble the Project Team and Define the Project..... | 1 |
| Step 1. Assemble Project Team | 5 |
| Step 2. Obtain CMDB Knowledge | 13 |
| Step 3. Create and Agree on CMDB Goals and Mission Statement..... | 21 |
| Step 4. Review and Define Benefits | 27 |
| Step 5. Build a Business Case | 37 |
| STAGE 2 Define Requirements and Create IT Service Model Blueprint | 45 |
| Step 6. Identify and Review Governance Requirements | 51 |
| Step 7. Review and Select Supporting Best Practices | 57 |
| Step 8. Identify Requirements to Address Potential Problems | 63 |
| Step 9. Identify Inventory and Asset Requirements | 69 |
| Step 10. Define Service Catalog Requirements..... | 75 |
| Step 11. Define CMDB Requirements to Support Other Processes..... | 83 |
| Step 12. Define Configuration Item Level and IT Service Model | 89 |
| Step 13. Define Configuration Item Relationships..... | 101 |
| Step 14. Define Configuration Item Attributes | 109 |
| Step 15. Design IT Service Model Blueprint | 119 |
| STAGE 3 Select CMDB Solution and Tools..... | 127 |
| Step 16. Select CMDB Solution | 131 |
| Step 17. Plan the CMDB Population | 145 |
| Step 18. Select Tools to Automate CMDB Population | 157 |
| Step 19. Calculate Project ROI | 163 |

| | |
|---|------------|
| STAGE 4 Construct and Maintain Your CMDB | 173 |
| Step 20. Construct Your CMDB..... | 177 |
| Step 21. Create CI Lifecycle Management Processes..... | 185 |
| Step 22. Build Supporting Processes..... | 193 |
| Step 23. Populate Your CMDB | 201 |
| Step 24. Train the CMDB Team and Users..... | 209 |
| STAGE 5 Driving Ongoing Value | 217 |
| Step 25. Implement Measures and Metrics..... | 221 |
| Step 26. Create a Continual Service Improvement Program..... | 229 |
| Epilogue | 237 |

Introduction

This edition of *Step-by-Step Guide to Building a CMDB* is an update of the original version, which was published in 2006. With the release of the Information Technology Infrastructure Library® (ITIL®) Version 3 (V3), we received numerous requests to update this book to incorporate the most current ITIL guidelines. Please keep in mind two key points: 1) ITIL provides a useful framework for IT service management, but you do not need to implement ITIL in order to build a CMDB; and 2) while this book is updated to better align with ITIL V3, readers who are implementing an ITIL Version 2 (V2) framework will still find the information helpful.

For readers unfamiliar with ITIL V3, the following description will assist you in understanding several of the terms used within this booklet.

ITIL V3 and the Service Management Lifecycle

ITIL V3 introduces the service management lifecycle approach, illustrated in Figure A. Keep this lifecycle approach in mind as you build your CMDB, and remember to view the CMDB in terms of how it can improve business services.

Service management begins with *service strategy*. The goal of service strategy is to design, develop, and implement service management as both an organizational capability and a strategic asset.

Next is *service design*, which is focused on ensuring that IT services offered to the business fulfill the objectives of the business. *Service design* is followed by *service transition*, in which the focus is on creating a smooth ride from design and development to operation.

Service operation then strives to effectively manage operational priorities, such as the availability of the IT services provided to the business; optimize the use of existing infrastructure; resolve issues; and control demand for services.

Improving the quality of existing IT services is the heart of the value delivered by *continual service improvement* (CSI). While the other four parts of the service management lifecycle are best performed sequentially, CSI is actually most effective when it is embedded as part of each of these lifecycle stages.

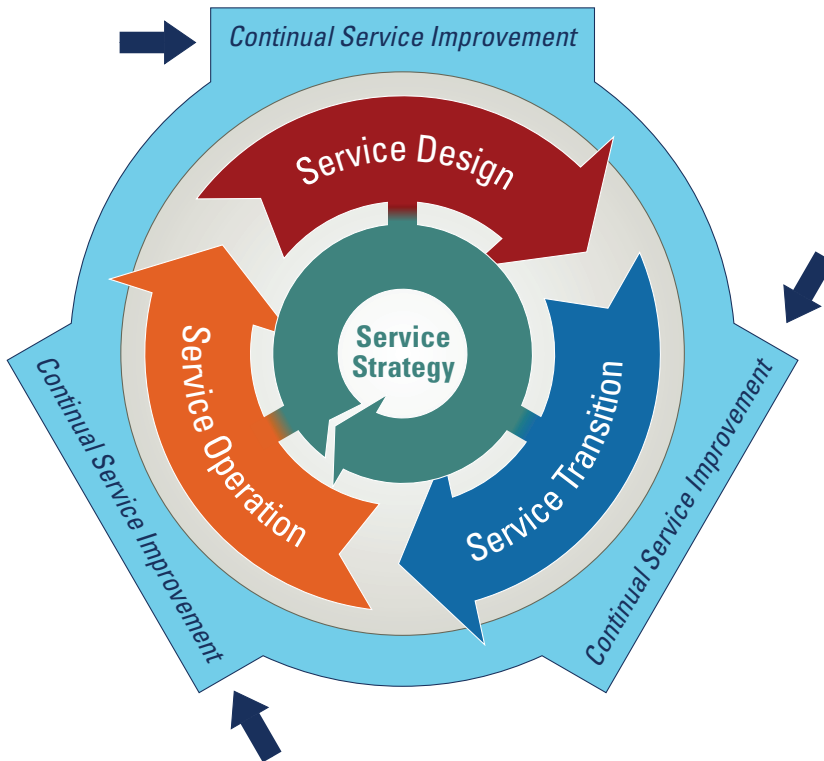


Figure A. ITIL Service Management Lifecycle

The CMDB and the CMS

While this book focuses on building a CMDB, it's helpful to understand how the CMDB fits within the broader configuration management system (CMS). According to ITIL V3, the CMS is comprised of databases and tools that manage configuration data for a service provider. More simply put, the CMS is the foundation that supports a complete service lifecycle across IT.

The goals of the CMS are to:

- » Maximize the value to the business
- » Manage business-critical services
- » Ensure compliance with governance and internal security policies
- » Enable automation to maximize efficiencies
- » Allow for control of assets

A CMS may include various IT management tools and databases, such as an asset database, a change management system, and a CMDB. Refer to the ITIL V3 core publications for a more detailed description of the CMS.

The Single Point of Reference

The CMDB provides a single point of reference, making it the definitive reference mechanism for all IT decisions by providing business-aware visibility into the dependencies among business processes, users, applications, and underlying IT infrastructure. This raises the awareness level for operators of the status of real-time business services, such as e-mail availability, Web site performance, and so on.

The leading CMDB solutions are all built to support a federated CMDB approach, meaning that not all configuration data must reside in a single physical database. Instead, the primary systems and data repositories remain the authoritative source for information, while the CMDB becomes the reference for where this information lives and how to access it. ITIL V3 now recognizes the importance of this federated approach and recommends that it be a core part of the structure of a CMS.

With federation, core data is stored in the CMDB, which is linked to other, more detailed data stores. This linkage provides a CMDB access to all configuration items (CIs). Hence, the CMS includes the CMDB or multiple CMDBs, and through federation, access to all primary data stores and their respective contents. By adding key functions, such as analytics, dashboards, and asset management, the CMS extends the value of the CMDB across IT.

Influencing Cultural and Organizational Change

An important point to keep in mind as you start down the path of building a CMDB is that any successful implementation of process and tools requires a cultural change. One of the most important keys to this is selling the benefits of the CMDB — not just benefits to the organization, but benefits to each individual or group impacted by the change.

For different people and groups, these benefits will vary. For the service desk, the benefits of a CMDB are greater visibility of all the components and relationships that make up the infrastructure, which in turn enables faster resolution of incidents, less stress, fewer errors, and so on. For financial managers, a CMDB allows greater accuracy in accounting for all the costs of IT, and enables calculations of TCO, ROI, etc. For the service level manager, it means visibility into all the components and relationships that make up an end-to-end service. For the business, an effective CMDB results in greater efficiency and effectiveness of IT, which in turn means increased agility in meeting the ever-changing business requirements.

Selling benefits is only the beginning, however. As ITIL points out, just changing IT services isn't enough to transform an organization. In the end, the organization will evolve in ways that allow it to use the changed IT services.

Communication both within IT and between IT and the business can improve dramatically with the implementation of a CMDB. In fact, the CMDB should become an integration platform that ties together multiple processes and tools. The organization should put a comprehensive communication plan into place to assist in managing the cultural changes.

You can identify how the CMDB will impact individuals and groups, and you can facilitate a successful transition by encouraging involvement, acknowledging the emotional cycle of transition, and so on.

Any maturity assessment of process and tools should include an assessment of the people, not only in terms of competencies but also in terms of their attitudes and behaviors. The information gathered from this assessment should then be used to manage the cultural change.

The requirement for improved documentation can change the culture from information hoarding to information sharing. This requirement can have a significant impact on the culture.

Learning As You Go

Although this book provides guidance for building a CMDB, your organization, like all others, will learn as you go. As your organization matures, you will continue to build on that learning through CSI. Without implementing a CMDB, no organization can grow past a certain level of maturity. There will always be growing pains, but the results will be more than worth it. We wish you great success.

Acknowledgments

This book would not have been possible without the contributions of the following individuals:

BMC Software ITIL Expert Contributors to Updated Edition

Dag Blokkum — As the global business process strategist for BMC, Blokkum is responsible for the company's global ITIL practice. Blokkum has more than 20 years of IT experience. He is the program chair and board member of itSMF Houston. Blokkum holds an M.S. in Computer Engineering, an M.B.A., and is certified as an ITIL Service Manager (Masters). He is an ITIL V3 Expert, and ISO/IEC 20000 Consultant/Manager.

Erin Casteel — A solutions architect for BMC, Casteel has more than 18 years of experience in IT. She has worked on IT service management projects in multiple industry sectors, and has trained close to 5,000 people in Foundation, Practitioner, and Masters levels of ITIL, and has also lectured at the University of Sydney. She is the representative from Australia on the international ISO/IEC 20000 Certification Committee.

Atwell Williams — As the solutions architect for the office of BMC's CTO, Williams has been responsible for assuring the overall performance, availability, and recoverability of BMC's internal production applications. He has led the initial implementation of Business Service Management (BSM) within BMC. Williams also is currently responsible for educating customers on ITIL and how to implement service management in their organizations.

Content Contributors (Original Edition)

David Chiu — A senior business technology specialist at BMO Financial Group, Chiu was one of the process architects for many of the ITIL processes implemented at BMO during the past five years, including release, change, configuration, and service level management. He now provides leadership and subject matter expertise for the continuous improvement of the established ITIL processes, using statistical process control techniques and other quality improvement methodology. Chiu won "Best ITIL Case Study" awards at the seventh and eighth Annual International IT Service Management Conferences. He is also a contributing author for the HDI book, *Implementing Service and Support Management Processes: A Practical Guide*.

Troy DuMoulin — An experienced executive consultant for Pink Elephant, DuMoulin has a solid and rich background in business process re-engineering. He holds ITIL Management certification and has extensive experience in leading service management programs of regional and global scope. His focus at Pink Elephant is to deliver strategic and tactical-level consulting services to clients, based upon a demonstrated knowledge of organizational transformation issues. DuMoulin is a frequent speaker at ITSM events and is a contributing author for the ITIL book *Planning to Implement IT Service Management*.

Brian Emerson — A senior solutions marketing manager with BMC, Emerson is currently responsible for the BMC® Atrium™ enabling technologies. He has been with BMC more than six years, coming from the Remedy acquisition where he was product manager for the Remedy® AR System Product Line. Emerson holds a bachelor's degree in Operational Management and Information Systems from Santa Clara University.

Malcolm Fry — A recognized IT industry luminary with more than 35 years experience in information technology, Fry conceptualized this book and was a significant contributor to the first edition. He is the author of four best-selling books on IT service and support, has been published in many journals and magazines, and is regularly contacted as a source of information by technology journalists. In addition, he is the solo performer in a highly successful, best-selling video series made for the Help Desk Institute. Fry is an original contributor to ITIL with ITIL Masters certification.

Rhonda Keller — A senior education developer for BMC, Keller is responsible for the BMC Atrium CMDB curriculum and is the co-developer of the BMC Configuration Management 7.x learning path.

Angie Massicotte — An IT consultant, Massicotte assists organizations with various ITSM/ITIL improvement initiatives. As an IT service quality manager with the Ontario government, Massicotte was key to the advancement of their IT service management program. Her focus with the Ontario government was monitoring and measuring service performance and defining configuration management information model requirements. Massicotte holds an ITSM Masters certification.

Kurt Milne — The managing director of the IT Process Institute, an independent research organization, Milne has more than 15 years of experience in marketing management, alliance management, and engineering positions at leading technology companies. His areas of expertise include IT service management and IT controls, inventory and supply chain management, and computer integrated manufacturing.

Jeanne Morain — With more than ten years experience leading development and deployment of products that enable Fortune 50 companies to comply with regulatory and business directives, Morain led early efforts for integrating discovery tools into asset systems. She has a master's degree from Southern Illinois University.

Michael Nicoletti — Nicoletti is known as a trusted advisor to senior IT executives worldwide. He was the first to attain Forrester Group's TEL certification for the BMC exclusive predeployment value model in April 2004. Nicoletti holds a bachelor's degree in finance from the Rochester Institute of Technology with a concentration in statistics.

Michael Oas — President of Communicating IT Inc., Oas is an accomplished information technology consultant with more than ten years of experience in IT service management. His previous experience includes a successful ten-year career as an air traffic/weapons control officer in the Canadian air force where he was involved in exercise development and contingency planning for NORAD.

Maria Ritchie — A strategic IT management consultant with AdvanciT Consulting Group, Ritchie has ten years of IT management experience in the public sectors. As a key leader and visionary in Ontario's strategic implementation of an ITIL-based Service Management framework, Ritchie was instrumental in defining the strategy and securing executive commitment to advance ITIL strategies in the Ontario provincial government. Ritchie is ITIL Masters certified.

Ken Turbitt — Turbitt has broad experience in best practices management, IT and consulting, has held an ISEB ITIL Manager/Masters qualification for more than ten years, and has been a Gartner-qualified TCO consultant.

Additional Contributors

Bob Balassi, Maryville Technologies

Kevin Behr, IT Process Institute

Carlos Bertaso, CSC Brasil

Gene Kim, IT Process Institute

Jonathan Markworth, CompuCom Systems, Inc.

Tim Mason, TRM Associates

Troy McAlpin, AlarmPoint

Doug Mueller, BMC Software

Javier Leyva Novoa, Quitze Tecnología

Brady Orand, Column Technologies, Inc.

Val Sanford, Singlestep Technologies

David Schrodel, Voyence

Perry Sellars, Strategic Technologies

George Spafford, IT Process Institute

Greg Wilson, Wipro

Frederieke C.M. Winkler-Prins, BMC Software

Marina Yesakova, BMC Software

Executive Editor: **Matthew Selheimer**

Managing Editor: **Elaine Korn**

Editorial Team: **Wendy Assatourian, Lea Anne Bantsari, Linda Donovan, Julia Wasson**

Cover Design: **Patrick Stull, Tekute Design Group**

Special thanks to **Liora Blum and Ayala Bashan, Liora Blum Graphic Design**

Michele Floriani

STAGE 1

Assemble the Project Team and Define the Project

Stage 1. Assemble the Project Team and Define the Project

During the first stage of your CMDB project, you will form a project team, scope the project, gather input from key constituents, and lastly get funding and approval to commit resources to the project. The benefits of a CMDB extend across various IT functions that benefit from access to consolidated and accurate data about the IT infrastructure, its users, and the services it enables. Therefore, ongoing focus on involving key constituents in the process of obtaining knowledge and scoping the problem, solution, and expected benefits is crucial to successfully gaining approval to move forward with the project.

This book intends to provide practical, how-to guidance on organizing a CMDB project and building a successful CMDB. All of the steps in this stage examine general IT project management best practices with an emphasis on the CMDB. If your organization already follows rigorous IT project management, then you may not need to spend a lot of time on these steps. But do take the time to review them, because these steps contain a great deal of useful CMDB information that will help you achieve success. Figure 1 shows the steps for assembling the project team and defining the project.

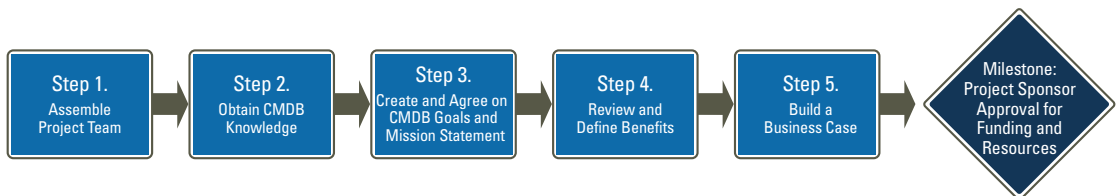


Figure 1. Steps for assembling the project team and defining the project

Each step contains specific goals that you should meet before moving forward.

You may want to use the following as a checklist.

Step 1.

Assemble Project Team

Goal: Identify the team members, including key constituents from affected IT functions whose input and support you'll need. Also scope your resource requirements for the other stages of the project. The right team members will increase your probability of project success.

Completed: ____

Step 2.

Obtain CMDB Knowledge

Goal: Educate team members on all aspects of a CMDB, including structure, terminology, scope, relationships with other IT processes, benefits, and the tasks required to build a CMDB. Investment in education now will dramatically reduce confusion and increase your ability to clearly describe the reasons for pursuing the project.

Completed: ____

Step 3.

Create and Agree on CMDB Goals and Mission Statement

Goal: Clearly describe the objectives of the CMDB project. Include a review of the current environment and problems to be solved, and a statement of the goals and mission for your CMDB project. Clearly defining goals is a critical early step toward getting the project approved, and then achieving the desired results.

Completed: ____

Step 4.

Review and Define Benefits

Goal: State the expected benefits and resulting organizational impact in terms of key metrics and measures. You want to link the problems and goals identified in the previous step to the specific benefits expected from pursuing the CMDB project. Detailed information about how the project will improve the organization will be used to build a business case and estimated ROI for the project.

Completed: ____

Stage 1. Assemble the Project Team and Define the Project

Step 5.

Build a Business Case

Goal: Use the output of the previous steps to build a solid business case so you can gain the buy-in and approval of key constituents, the project executive board, and the overall project sponsor. You need a logical, simple business case to win approval over other projects requesting scarce resources.

Completed: ____

Milestone: Project Sponsor Approval for Funding and Resources

Congratulations. You've made your case. You outlined the resources and funding required to implement a CMDB, and you clearly described how the organization would benefit. It's time to move forward and define solution requirements in stage 2.

Milestone achieved: ____

Step 1

Assemble Project Team

Goals and Objectives

In this step, you will select a project team that has the required skills, experience, and knowledge to justify, design, and implement a CMDB that meets all of the other goals described in this step-by-step guide. The team's capabilities should include not only technical and configuration management skills but also project design and project management skills.

Overview

This step focuses on what you need to do to get started on your CMDB project, including selecting a project manager and project team. This step, while rudimentary, is critical to your CMDB project and cannot be ignored. A skilled and well-managed project team will greatly increase your ability to create a successful CMDB. Starting with step 2, "Obtain CMDB Knowledge," the guidance is targeted directly to the project manager and project team.

Task-by-Task Guide

Figure 1.1 shows the tasks to assemble the project team.

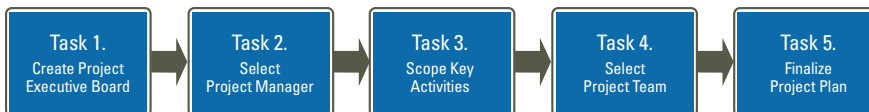


Figure 1.1. Tasks to assemble the project team

Task 1. Create Project Executive Board

The main objective of the project executive board (PEB) is to act as a guardian to the project, ensuring that:

- » The project stays on schedule
- » The project meets its objectives and goals
- » The project stays within budget
- » Decisions are made at key project gateways
- » Exceptions are reviewed and resulting actions are taken
- » The authorizations and change requests necessary for the project are provided

Creating a PEB can be contentious, particularly if a number of groups want to have a representative on the PEB. Here are a few tips:

- » Keep the PEB small enough so that it can move quickly when required and you can avoid project delays.
- » Make sure the PEB has the authority to make decisions on matters such as budgets and purchasing, as well as what constitutes an acceptable ROI.
- » Verify that the PEB members have the time to dedicate to the project, including reviewing regular project updates.
- » Select PEB members who can provide guidance to the project manager when required.

Project team members probably will not be part of the PEB, although having the project manager on the PEB can be advantageous. You'll need to weigh the benefits of having the project manager on the PEB against the possibility that the project manager may not be able to be truly independent from the project, and may influence PEB decisions accordingly.

At a minimum, the PEB should include representatives from the following:

- » **Executive sponsor** — An executive sponsor is critical because building the CMDB will affect numerous IT and business organizations. An executive-level sponsor with responsibility for multiple IT departments gives the project both credibility and focus. Keep in mind that the more senior the sponsor, the higher the level of credibility that is associated with your project. The CIO is an excellent candidate; alternatively, look to others such as the vice president of IT support, director of infrastructure, or director of information communications. The key is to find an executive sponsor who shows a clear interest in the project and who will benefit from the CMDB.
- » **Key stakeholders** — Those organizations that plan to change their functions or processes to leverage the CMDB are key stakeholders, and they should be represented on the PEB since they will depend on the success of the project. Organizations that are contributing budget and resources to the CMDB project may also want to be represented on the PEB. These stakeholders might be from governance; the service desk; or asset, change, release, incident, problem, configuration, financial, availability, capacity, or service level management. These key stakeholders may select a deputy to represent them, but the deputy should have the authority to make decisions on behalf of the sponsors.

- » **CMS owner** — Include the future CMS owner on the PEB; that is, the person who owns and manages the complete CMS, which is comprised of all the individual CMDBs. This representative is responsible for ensuring that the overall CMS and the individual CMDBs can be managed and maintained to the level defined in the CMDB goals and mission statement you will develop in step 3, “Create and Agree on CMDB Goals and Mission Statement.”
- » **Main resource contributor** — One team or organization will probably provide the bulk of the labor required to complete the CMDB project, and should be represented on the PEB. At heart, this is a project to define and build a database, so the sponsor and other stakeholders likely will not complete much of the work. This PEB member is responsible for ensuring that the necessary resources are supplied when required, and for making decisions if some tasks take longer than anticipated or require more resources.

Take great care to select the correct PEB members. Remember: The PEB should be an asset to the project — not an encumbrance.

Task 2. Select Project Manager

Selecting the project manager for this CMDB project is, as with all projects, a critical decision. Ideally, the project manager will have technical skills and will be able to contribute to the project as well as manage it.

This project focuses on successfully implementing a CMDB, which is a core component of ITIL® V3. Although you don’t have to implement ITIL best practices to have an effective CMDB, a project manager with a Manager’s (Master) certification level in ITIL will provide a distinct advantage. This certification proves that the project manager understands the role of the CMDB and has the knowledge to implement ITIL components. In addition to basic project management skills, you should look for the following skills, experience, and attributes in the project manager:

- » Planning skills
- » IT service management background
- » Previous involvement in building a database
- » ITIL Manager’s certification
- » Ability to make decisions
- » Capabilities to motivate staff
- » Ability to blend a team
- » Strength of character to lead a team
- » Capacity to present results and status of project to sponsors and stakeholders
- » Self-motivation
- » Ability to communicate instructions to the project team

Stage 1. Assemble the Project Team and Define the Project

Use these ideas to help you create a profile of the ideal project manager for your CMDB project. Make sure that the PEB either chooses or is actively involved in selecting the project manager, since the project manager and the PEB will work closely together.

Task 3. Scope Key Activities

Designing a project from scratch is always a challenge, and this step-by-step guide has been developed with that in mind. This guide includes 5 stages, subdivided into 26 steps and more than 100 tasks, in a project-oriented structure:

Stage 1. Assemble the Project Team and Define the Project

Step 1. Identify Project Team Members

- Task 1. Create Project Executive Board
- Task 2. Select Project Manager
- Task 3. Scope Key Activities
- Task 4. Select Project Team
- Task 5. Finalize Project Plan

Step 2. Obtain CMDB Knowledge

Step 3. Create and Agree on CMDB Goals and Mission Statement

Step 4. Review and Define Benefits

Step 5. Build a Business Case

Stage 2. Define Requirements

Each stage comprises a number of steps, and each step contains multiple tasks. It is a simple matter to convert this structure to a Microsoft Project document, as shown in figure 1.2.

| STAGE, STEP, OR TASK NAME |
|---|
| Assemble the Project Team and Define the Project |
| Identify Project Team Members |
| Create Project Executive Board |
| Select Project Manager |
| Scope Key Activities |
| Select Project Team |
| Finalize Project Plan |
| Obtain CMDB Knowledge |
| Create and Agree on CMDB Goals and Mission Statement |
| Review and Define Benefits |
| Build a Business Case |
| Define Requirements |

Figure 1.2. Convert stages, steps, and tasks into a Microsoft Project document

By following this book, you can quickly create a Microsoft Project file. Look at each stage, step, and task, and decide whether it needs to be included in your project. At the same time, remain vigilant for any extra stages, steps, or tasks that you may need to add.

Task 4. Select Project Team

Your project team should consist of:

- » **Dedicated project team members** — These people will be dedicated to this project from the outset through completion.
- » **Subject matter experts** — You will need these specialists to support certain tasks of the project. For example, you might need an expert to help build the ROI or to help identify governance requirements. In addition, you may need someone from the information security team when applications and technologies are being selected, or a solutions architect to help write the RFP.

For this CMDB project, your dedicated project team members will come from a small, select group, and you should consider the following:

- » **Future CMDB owners** — To ensure consistency after implementing your CMDB, include a representative from the team that will ultimately manage and maintain the CMDB.
- » **Skills, knowledge, and experience** — Look for representatives that have extensive skills related to building and implementing databases. Also seek representatives who have knowledge of asset management and other ITIL processes. If the project team will consist of external resources, a prerequisite should be that at least one of them has built and implemented a CMDB.
- » **Workload** — Much of the workload will occur in the middle stages when you build and populate the CMDB. Consider other workload commitments when selecting dedicated project team members.

These characteristics are easier to identify when you add the resources required to perform the stages, steps, and tasks listed in the project plan, as shown in figure 1.3.

| STAGE, STEP, OR TASK NAME | RESOURCE GROUP |
|---|---|
| Assemble the Project Team and Define the Project | Names of Groups Supplying Resources for each Stage, Step, and Task |
| Identify Project Team Members | |
| Create Project Executive Board | |
| Select Project Manager | |
| Scope Key Activities | |
| Select Project Team | |
| Finalize Project Plan | |
| Obtain CMDB Knowledge | |
| Create and Agree on CMDB Goals and Mission Statement | |
| Review and Define Benefits | |
| Build a Business Case | |
| Define Requirements | |

Figure 1.3. Identify resources for each stage, step, and task

Stage 1. Assemble the Project Team and Define the Project

From these resource groups, identify potential dedicated and subject matter expert project team members. Consult with them or with their managers concerning their availability to help finalize the project plan. Assuming they are available, they can help identify the workload and the target start and end times. If they are not available, then the project manager must perform these calculations and estimates. Either way, start by identifying the workload, as shown in figure 1.4.

| STAGE, STEP, OR TASK NAME | RESOURCE GROUP | WORK |
|---|--|---|
| Assemble the Project Team and Define the Project | Names of Groups Supplying Resources for each Stage, Step, and Task | Number of Hours Required to Complete each Stage, Step, and Task |
| Identify Project Team Members | | |
| Create Project Executive Board | | |
| Select Project Manager | | |
| Scope Key Activities | | |
| Select Project Team | | |
| Finalize Project Plan | | |
| Obtain CMDB Knowledge | | |
| Create and Agree on CMDB Goals and Mission Statement | | |
| Review and Define Benefits | | |
| Build a Business Case | | |
| Define Requirements | | |

Figure 1.4. Identify workload for each stage, step, and task

Next, the project manager, with the help of the potential project team members, will need to add the target start and finish dates for the stages, steps, and tasks (figure 1.5). Keep in mind that these dates might be affected by a scheduled delivery date of the new CMDB.

| STAGE, STEP, OR TASK NAME | RESOURCE GROUP | WORK | START | FINISH |
|---|----------------|------|--|---|
| Assemble the Project Team and Define the Project | | | Target Start Date for each Stage, Step, and Task | Target Finish Date for each Stage, Step, and Task |
| Identify Project Team Members | | | | |
| Create Project Executive Board | | | | |
| Select Project Manager | | | | |
| Scope Key Activities | | | | |
| Select Project Team | | | | |
| Finalize Project Plan | | | | |
| Obtain CMDB Knowledge | | | | |
| Create and Agree on CMDB Goals and Mission Statement | | | | |
| Review and Define Benefits | | | | |
| Build a Business Case | | | | |
| Define Requirements | | | | |

Figure 1.5. Identify target start and finish dates for each stage, step, and task

In addition to subject matter experts and project team members, you will need to talk with the business to identify business needs.

Don't forget to identify all relevant governance requirements and address them in your project plan.

Once the resources, workloads, and start and finish dates have been identified, the project manager should seek approval from the managers of the potential project team members for their staff to work on the project. The project manager might need to perform some skilled negotiating, but the necessary levels of staff are important. Appropriate resources will help you complete the project on time and meet all of the project's objectives. If you use outside resources, then the project manager will have all of the data to negotiate with confidence.

Task 5. Finalize Project Plan

Finally, note the estimated cost of the workload forecast for the project, as shown in figure 1.6.

| STAGE, STEP, OR TASK NAME | RESOURCE GROUP | WORK | START | FINISH | COST |
|---|----------------|------|-------|--------|--|
| Assemble the Project Team and Define the Project | | | | | The Cost of the Resources for each Stage, Step, and Task |
| Identify Project Team Members | | | | | |
| Create Project Executive Board | | | | | |
| Select Project Manager | | | | | |
| Scope Key Activities | | | | | |
| Select Project Team | | | | | |
| Finalize Project Plan | | | | | |
| Obtain CMDB Knowledge | | | | | |
| Create and Agree on CMDB Goals and Mission Statement | | | | | |
| Review and Define Benefits | | | | | |
| Build a Business Case | | | | | |
| Define Requirements | | | | | |

Figure 1.6. Identify the cost of the workload for each stage, step, and task

This estimated cost represents the cost (number of hours x average hourly rate) of performing the actions required to complete the project on time and to meet the project's objectives and goals. It is not the cost of the tools and technologies that may be required to implement the CMDB.

The basic project plan should now be complete, as shown in figure 1.7.

The project manager should now present the project plan to the PEB for approval. If the PEB asks for some changes to the project plan, then the project manager should integrate these changes and resubmit the project plan for approval.

Stage 1. Assemble the Project Team and Define the Project

| STAGE, STEP, OR TASK NAME | RESOURCE GROUP | WORK | START | FINISH | COST |
|---|--|---|--|---|--|
| Assemble the Project Team and Define the Project | Names of Groups Supplying Resources for each Stage, Step, and Task | Number of Hours Required to Complete each Stage, Step, and Task | Target Start Date for each Stage, Step, and Task | Target Finish Date for each Stage, Step, and Task | The Cost of the Resources for each Stage, Step, and Task |
| Identify Project Team Members | | | | | |
| Create Project Executive Board | | | | | |
| Select Project Manager | | | | | |
| Scope Key Activities | | | | | |
| Select Project Team | | | | | |
| Finalize Project Plan | | | | | |
| Obtain CMDB Knowledge | | | | | |
| Create and Agree on CMDB Goals and Mission Statement | | | | | |
| Review and Define Benefits | | | | | |
| Build a Business Case | | | | | |
| Define Requirements | | | | | |

Figure 1.7. Completed project plan

Summary

Getting the right people involved at the outset is critical to the success of your CMDB project. You need an effective PEB to help keep the project on track and to make key decisions along the way. A strong project manager and a skilled project team are also essential elements.

Use the guidance here to create a project plan to help you select the right project team members. Although this takes time, creating a project plan ultimately does not delay the delivery of a CMDB. In fact, delays and frustrations occur when the correct resources are not available when they are required.

The steps and tasks in this book are intended to be a guide. If you decide you don't need a particular step, just leave it out of your plan. Alternatively, you may need to add some steps to meet the needs of your unique situation.

As the project progresses, you may need to update the project because of new ideas, lessons learned, revised emphasis, change of deliverables, lack of finances, and demands of sponsors and stakeholders. If this is the case, then use this step to help you re-engineer your project and to help you determine your resource needs.

Step 2

Obtain CMDB Knowledge

Goals and Objectives

The goal of this step is to quickly educate the key members of the CMDB creation team so that they are fully conversant in all aspects of a CMDB, including structure, terminology, scope, relationships with other IT processes, and benefits, and they understand the tasks required to build a CMDB. The key members should pursue at least ITIL Foundation certification level of education so that they understand the concepts of a CMDB and how it relates to the service lifecycle and other service management processes and functions. The output of this step is a well-informed team that will dramatically reduce confusion and increase the chances of project success.

The CMS owner should have a high degree of knowledge with advanced certificates in ITIL and/or ISO /IEC 20000. It is also helpful to have process knowledge, such as Prince 2. In addition, project management experience is essential to have on the team, as setting up an efficient CMS can be a complex project.

Overview

The goal in this step is to educate team members to the level required so each can contribute successfully to this CMDB project. Practical knowledge is vital in this project. Therefore, do not worry too much about extensive certifications or qualifications for all team members. It's important to separate the knowledge required to create and build a CMDB from the education required to work in IT service management on a daily basis. The CMDB team members probably will represent different areas in IT. A database designer, for example, will bring his or her own knowledge and may not have a comprehensive knowledge of ITIL education. And that's fine.

Although the ultimate goal is to obtain CMDB knowledge, you may want to quickly review any complementary skills, such as database design or knowledge of service management processes, that might be useful. Also, IT service management staff with little project management experience may need project training. Keep in mind that the need for further skills often emerges when a new requirement is identified or when the original depth of knowledge must be expanded.

Stage 1. Assemble the Project Team and Define the Project

As the project team amasses knowledge, make note of, and keep in a central repository, potential problems that could affect either the building of a CMDB or the ongoing management of the CMDB once implemented.

A vast amount of knowledge and information is available on CMDBs, so be sure to explore all sources. This step will provide information to get you started in a number of areas, including education, use of the Internet, useful books and publications, societies, and conferences and seminars.

Task-by-Task Guide

Figure 2.1 shows the tasks for obtaining CMDB knowledge.

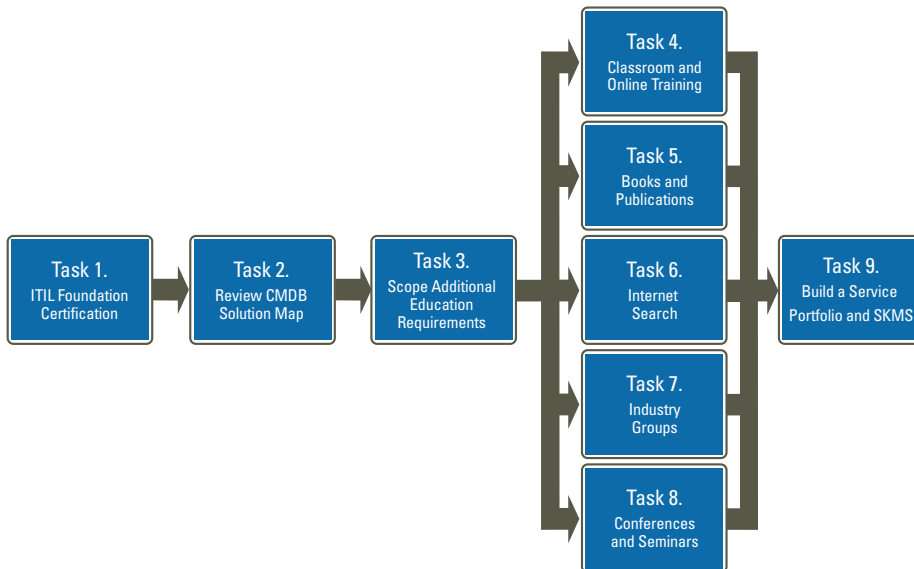


Figure 2.1. Tasks for obtaining CMDB knowledge

Task 1. ITIL Foundation Certification

Your project will run much more smoothly if key project team members achieve the basic ITIL education — and are certified at the ITIL Foundation level. The Foundation Level Certification includes an understanding of not only the CMDB, but also the basics of all of the ITIL Service Management processes and the standard ITIL terminology. This common knowledge is essential to establishing consistency and better communication. It is best if the team members are all on the same page from the very beginning.

Many organizations around the world provide ITIL education. A good place to start is with software vendors, such as BMC Education (www.bmc.com/ITILeducation), or industry groups, such as the IT Service Management Forum (itsmf at www.itsmfi.org). Review their learning paths to see which courses would be relevant for you.

Task 2. Review CMDB Solution Map

After key members have attended ITIL foundation training (preferably ITIL V3), the team should review and ensure common understanding of the primary components of a CMDB solution before they scope additional education requirements. If you have in-house experts for specific solution functions, then that will reduce the scope of CMDB education required by the team.

The CMDB solution market is maturing, but vendors may not all share a common vocabulary. Be sure to categorize solution elements by basic function so you can prevent terminology confusion. Figure 2.2 shows the core elements of a modern CMS to allow for better visualization and decision process of all the data the CMDB manages. Note that this figure aligns with the Data-Information-Knowledge-Wisdom (DIKW) methodology discussed in ITIL V3.

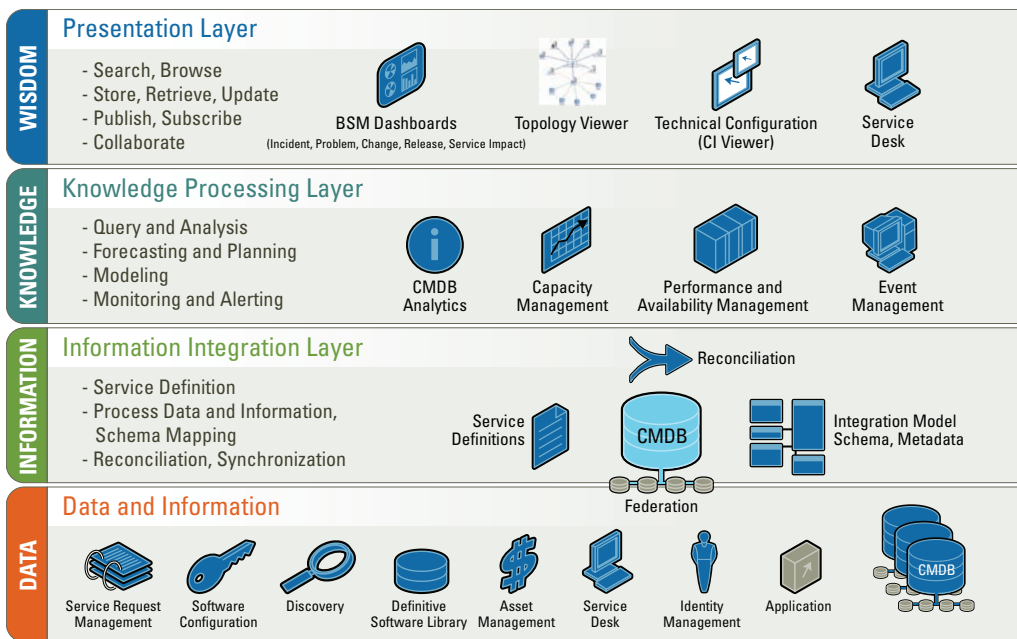


Figure 2.2. Sample Configuration Management System (CMS)

USING ITIL IN THE REAL WORLD

An IT manager for a leading provider of healthcare services discusses the importance of using ITIL in a CMDB implementation.

From the outset of our CMDB implementation, we understood that to realize the many benefits of a CMDB, we needed to create a framework to help ensure consistency across the enterprise. We also decided that we would do more than just become familiar with ITIL; we would study it closely and use an ITIL-based framework for our deployment. The flexibility of the ITIL specification made it easy to adapt it to our specific needs.

Most large organizations already have some form of change management and incident tracking system in place, as was the case with us. Ironically, existing systems — which can be plentiful, as you may find that separate groups have created their own solutions — make it all the more important for you to invest in the upfront commitment to understanding ITIL principles and creating a common yet flexible framework.

Without a framework, there can be a temptation to try to confederate existing systems, which tends to fuel a resistance to change. But without the discipline of a formal, centralized CMDB with precise configuration item (CI) definitions, you can't gain the consistency required to operate across the enterprise.

Additionally, while organizations may have change management systems in place, not as many have configuration management solutions, and even fewer have configuration data integrated seamlessly with change management and incident management data. A well-planned ITIL-based framework accommodates all.

Excerpt from "The CMDB: Relief for Your IT Headaches," VIEWPOINT, Focus on: CMDB, published by BMC Software.

Task 3. Scope Additional Education Requirements

Obviously, knowledge and education are key components for the successful building of a CMDB. You'll want to strike the right balance between having enough knowledge and over-investing or delaying time-to-value for your project. A best practice learned from past CMDB projects is to assign one project team member to manage education for the entire team.

Every team member should have an understanding of chapter 4 in the *Service Transition* book, which includes a discussion of service asset and configuration management (SACM). Team members also need to understand how the CMDB relates to other ITIL processes. You can use this level of understanding as a basis, and analyze the education level of each member of the project team. Then you can identify gaps where further education is required.

Task 4. Classroom and Online Training

After you have decided that key project team members should pursue additional education, you should first focus on selecting a company to provide that training. Many companies, including BMC and Pink Elephant, specialize in ITIL and CMDB education. Evaluate each before making your selection.

Create a description of your exact training requirements to present to the vendor before booking or attending any courses. Each team member who attends a course should report back to the team and evaluate the course. The first course you try may not meet your needs, so you may need to try other vendors until a suitable one is found.

Task 5. Books and Publications

All members of the project team should identify books, publications, and media such as podcasts and DVDs that contain potentially useful information concerning configuration management and the CMDB. The five core ITIL V3 books provide a good starting point. It is strongly recommended that all team members read all five core ITIL books. However, if you choose not to follow this recommendation, you should at least have one team member read each book and present a summary to the rest of the team. The five core ITIL V3 books include:

- » *ITIL V3 Service Strategy* provides insights into understanding and making strategic choices that allow IT organizations to outperform competing alternatives.
- » *ITIL V3 Service Design* offers guidance on designing processes to ensure that the IT infrastructure can effectively provide services aligned to business needs.
- » *ITIL V3 Service Transition* provides guidance on taking the output of the service design phase and creating usable services to the business.
- » *ITIL V3 Service Operation* provides guidance on all aspects of managing the day-to-day operation of an organization's IT services. It also discusses the operational aspects of many processes.
- » *ITIL V3 Continual Service Improvement* provides guidance on all aspects of continual service improvement for IT organizations.

Other useful publications include *VIEWPOINT, Focus on: CMDB*, and *VIEWPOINT, Focus on: CMDB Leadership*, published by and available through BMC Software.

Other relevant publications can be found at:

- » www.isaca.org
- » www.litgi.org
- » www.iso.org
- » www.itsmfi.org

Task 6. Internet Search

The Internet is a great source of CMDB information. Software vendor Web sites often contain a large amount of useful information, including excellent white papers, blogs, and podcasts about the CMDB and configuration management.

You can obtain CMDB and configuration management information at the following Web sites:

- » www.bmc.com/cmdb
- » www.bmc.com/thoughtleadership
- » www.itimassociation.com

Task 7. Industry Groups

Knowledge gained from networking with people who have already implemented a CMDB is invaluable. Important sources of networking opportunities are societies and user groups. Identify and evaluate as many societies and groups as possible, and select those most appropriate for your purposes. One example is the itSMF, a leading organization for IT service management networking. itSMF also works closely with the Office of Government Commerce (the creator of ITIL) to produce the ITIL publications.

Project team members should join these groups and attend meetings, and should report any useful information to the rest of the team.

Task 8. Conferences and Seminars

Conferences and seminars combine education, relevant presentations, networking opportunities, and vendors all in one package. Seminars can be a great resource, because they offer a lot of practical experience — often for free. Personal recommendations and the contacts gained from networking at society meetings can help you select conferences. Here are four of the best Web sites for IT service management-focused conference information:

- » www.itimassociation.com
- » www.itsmfi.org
- » www.pinkelephant.com
- » www.thinkhdi.com

Each member who attends a conference or seminar should report any useful information to the rest of the team.

Task 9. Build a Service Portfolio and a Service Knowledge Management System (SKMS)

Ensuring that all the materials, project plans and past experience are kept in a systematic order is the job of the service portfolio management (SPM) process. The SPM keeps track of the entire project life cycle, from strategic analysis to policy setting, gathering the requirements, and getting approvals. It tracks the development, architecture, inputs/outputs, resources, costs, time lines, and all project dependencies. The SPM also draws on past experiences of similar projects, which are found in the service knowledge management system (SKMS). All of the repositories can be fairly simple.

Following the best practice principles of ITIL V3 SPM and SKMS greatly enhances the chances of project success by helping keep the project on track, on time, and on budget, and delivering the requirements identified by the business.

Summary

Investment in education at this stage will dramatically reduce confusion and increase the chances of project success. Remember that the project team will continue to require knowledge as the project progresses, so keep in mind that some of these tasks may have to be repeated.

Also remember that the SKMS is a valuable source of information for the team who will be responsible for managing and maintaining the CMDB when it is successfully installed. A key best practice from successful CMDB projects is to always keep your SKMS current to ensure all team members have access to the same information resources during the project.

Step 3

Create and Agree on CMDB Goals and Mission Statement

Goals and Objectives

The purpose of this step is to create a clear set of goals and objectives for your CMDB project. These goals must be linked to the overall business strategy. Doing a strategic analysis will determine which business-critical services to include in the CMDB and the overall CMS. In addition, the strategic assessment will determine what controls should be in place as you plan for the depth and breadth of the CMDB. From this you can create a mission statement. Next, you can determine your critical success factors and compile a list of key performance indicators (KPIs) against which to measure the success of the CMDB project.

Overview

The overall goal of the CMS is to support the business-critical services. Ultimately you need to be able to keep your IT services under the control of the CMS. You also need to integrate all the other IT Service Management processes to the CMS to allow for the most efficient and effective management of these business-critical services. A good service strategy will allow you to designate the most business-critical services and focus on those that bring the best value back to the business.

Due to the cross-functional nature of a CMDB, you can attribute a wide variety of goals to a CMDB. However, the goals of your project should support the needs of your business and align with the current priorities within IT.

Various IT functions and processes can be improved if they link to and utilize the data in a federated CMDB. However, there is no point in building a CMDB if it isn't going to be leveraged to improve the different IT processes that help deliver and support IT services. Therefore, it is important to engage the owners of the various processes to get their input and support on the CMDB project goals.

Stage 1. Assemble the Project Team and Define the Project

Your CMDB goals should be clear, concise, and plainly describe the overall deliverables of the project. Once you are satisfied with your goals, it will be easy to create the mission statement, which naturally flows from the established goals.

Task-by-Task Guide

Figure 3.1 shows the tasks for establishing your CMDB goals.

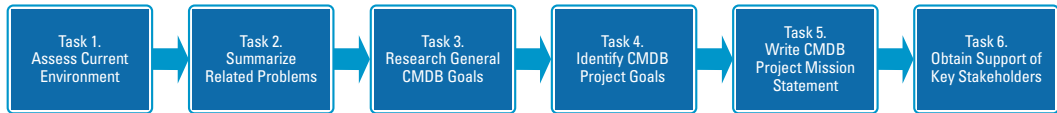


Figure 3.1. Tasks for establishing CMDB goals

Task 1. Assess Current Environment

The CMDB can drive improvements across many functions and processes in IT. However, overall project success depends on the CMDB being aligned with current business, IT executive, and process owner priorities.

You should start your process of defining project goals by assessing the current environment at two key levels. First, assess the overall IT initiatives prioritized by the IT executive team. Second, identify and assess the IT functions that can benefit from a CMDB and determine the current priorities of those individual groups.

Start at the top by researching your CIO's and project sponsor's top priorities. What are the key strategic initiatives that your CIO is pursuing? What is important to your CMDB project sponsor? What are the key measures that are being tracked? What are the important initiatives and goals for the IT functions related to your project? How do these relate to the overall corporate strategy?

Make sure your CMDB project supports the goals and objectives of your IT executives, which should in turn align with the needs of the business. This will increase the chances that financial and people resources will be allocated to your CMDB project. If your CMDB project does not support the highest-level goals and objectives of the organization, you should ask yourself why you are pursuing a CMDB right now.

If a corporate goal is to increase Web sales by 25 percent, and IT's goal is to help facilitate this goal by ensuring the Web store remains operational 99.99 percent of the time, then consider ways the CMDB can help. If a corporate goal is to improve customer satisfaction by ensuring timely delivery of orders, and IT's goal is to enable prompt order fulfillment, then how can the CMDB assist in meeting this goal? Look for ways that you can align the goals of your CMDB project with overall IT and company goals and initiatives.

Now identify the various IT groups and functions that may benefit from a CMDB. Reach out to the owners of various IT processes and services, and get them involved in discussions about how a CMDB may benefit them.

You also need to be sure that the scope of your CMDB is defined in conjunction with the scope of change and release management.

The key to a successful CMDB and CMS is to have an integrated IT service management that can do the following:

- » Manage the flow of incidents, problems, changes and releases
- » Give information to the service desk
- » Manage your overall service level agreements
- » Control your financial assets
- » Manage recovery options
- » Provide information for availability, capacity, and supplier management

All of your technology components should be integrated. The business-critical technology components should be integrated into the CMDB. Then all the various technology teams need to work with the CMS. This may include the mainframes, the databases, the networks servers, applications development, and data centers.

Be methodical. Use this list to identify key stakeholders throughout IT and other areas of the organization. Remember to tap into business managers, compliance officers, information security managers, auditors, and accountants.

Meet with them one by one and tell them about your CMDB project. Ask them what their current priorities are. Ask if they would benefit from access to consolidated information about the IT infrastructure and the components' relationships to each other and the services they support. Also ask about their personal goals. Can a CMDB help them achieve their goals?

CMDB GOAL FOR A LARGE INSURANCE COMPANY

A large insurance company provides an excellent example of setting a CMDB goal. The company's IT organization has been in the process of implementing ITIL for several years and is continually looking for areas to improve upon. One year the focus was on improving problem management and knowledge management. They set a goal of improving the first-call resolution rate from 65 percent to 85 percent (the ITIL benchmark for best-practice organizations). Now, their main goal is to try to keep the number of unauthorized changes to a minimum, and a strong CMDB is the only way to do that.

Excerpt from VIEWPOINT, Focus on: CMDB, published by BMC Software.

Task 2. Summarize Related Problems

Now that you know what's important to your CIO, project sponsor, and other key stakeholders, summarize the problems that underlie these strategic initiatives. Have there been issues with the stability of the Web store? Were these issues caused by a less-than-perfect change management process that didn't evaluate the impact of proposed changes?

Stage 1. Assemble the Project Team and Define the Project

Has customer satisfaction declined because of shipping delays? Were the shipping delays the result of a manufacturing system delay, or a problem in the shipping department? Did an IT system or infrastructure component contribute to the delay?

Take the time to gain a clear understanding of the challenges that your IT organization and overall company are facing. This will give you a basis for developing the goals of the CMDB project.

This information will be summarized as the problem description in the business case you develop in step 5, “Build a Business Case.” In that step, you’ll link problem, solution, and benefits, so it’s important that the problem description be accurate and compelling.

The problem description should include a summary of how the organization currently functions without a CMDB. This section will be as detailed as required and can include process diagrams and statements from stakeholders in the business or technology group about the challenges currently faced by various siloed IT functions working without a CMDB.

Task 3. Research General CMDB Goals

Now conduct research on the areas in which a CMDB can address the problems you identified in the previous task. Assessing general CMDB goals is similar to assessing general CMDB benefits, which is covered in step 4, “Review and Define Benefits.” You may want to reference the CMDB benefits detailed there to help review potential CMDB goals.

If you are implementing ITIL V3, refer to the *Service Strategy* and *Service Transition* books for more information.

Assign project team members to obtain the relevant data from these sources. Convert the information into goal statements to address the problems identified in task 2.

Think about two sets of goals: goals for the project team and goals for the overall project, which should align with the CMDB benefits that will result from various IT functions improving their IT processes with better access to key information in the CMDB.

Following are some points to consider for creating draft goals:

- » Maximizing the value back to the business in managing the most business-critical services
- » Controlling the IT environment to ensure governance compliance
- » Assuring company security policies are adhered to in the CMS and under CMS control
- » Setting up an environment for automation of the most routine management tasks
- » Establishing basic asset controls of your most critical assets

Task 4. Identify CMDB Project Goals

Now, evaluate all of the goal statements you drafted and prioritize them. Pare down the list to five to ten goals that you will pursue, and turn them into SMART goals. (As a reminder, SMART goals are **s**pecific, **m**easurable, **a**ttainable, **r**elevant, and **t**ime-based.)

Following are examples of goals for the CMDB project:

- » Provide a sound basis for incident management to improve the first-call resolution rate on the help desk to 90 percent by Nov. 15.
- » Improve the rate of successful changes to 98 percent by Dec. 15 by offering necessary information to support the change management process.
- » Enable the IT controls required to meet Sarbanes-Oxley requirements by Dec. 31.
- » Provide a central repository for asset data to optimize controls for software licenses, leases, warranties, retirement, depreciation, and total cost of ownership (TCO) by Sept. 30.
- » Develop and populate a CMDB by March 1 to support the data center move by June 1.

Task 5. Write CMDB Project Mission Statement

Once you have agreed upon the goals for the CMDB project, it's time to write the mission statement, which captures the essence of your project goals and the thinking behind them. The mission statement should include why you are undertaking the CMDB project, the benefits of the project, the deliverables, and how you will accomplish your goals. Be sure that your mission statement reflects the views of senior management and the project sponsor and focuses on the needs of the business.

The best mission statements are developed with input from everyone on the team. Just going through this process can help the CMDB team understand why you are pursuing this project, and the mission statement itself can serve as inspiration for the project team. Having a mission statement that articulates the desired outcome ensures a coordinated approach from the various people and teams that are contributing to the project.

The mission statement should be brief — two to three sentences. It should be easily understood, and team members should be able to readily articulate it. While this task is important, don't spend an inordinate amount of time creating your mission statement — there is much work to be done on your CMDB project.

The following are examples of mission statements:

- » To create a CMDB and supporting processes that meet and surpass ISO/IEC 20000.
- » To create a CMDB and processes that effectively support business requirements, compliance, and the requirements of all ITIL processes.
- » To create a CMDB that achieves visibility and control in a multisourced environment.

Stage 1. Assemble the Project Team and Define the Project

These examples meet most of the objectives stated earlier. Keep in mind that the goals and the mission statement must complement each other, as shown in figure 3.2. Once these goals have been achieved, they will be replaced by the next set of goals. All goals need to be prioritized in order of importance to the business.

| Mission Statement | Goals |
|--|--|
| To create a CMDB and processes that effectively support business requirements, compliance, and the requirements of all ITIL processes. | <ul style="list-style-type: none">• Discover all infrastructure CIs and their relationships, verify the configuration records against the infrastructure, and correct any exceptions by June 30.• Provide accurate version control of all non-desktop system configurations by Sept. 1.• Connect data producers and consumers in incident management, problem management, change management, and release management by Nov. 1.• Provide a complete list of infrastructure CIs that support financial applications for IT audit by Jan. 1. |

Figure 3.2. Example CMDB mission statement with goals

Task 6. Obtain Support of Key Stakeholders

Distribute the goals and mission statement to key stakeholders for comments and input. This distribution must include everyone who was consulted as a resource for the goals in task 1. Also distribute to anyone else who will be influenced or affected by the CMDB when it is implemented.

Review comments and input as you receive them from the stakeholders. To reduce the chances of conflict later in this project, follow up with all of the stakeholders — especially those who did not provide any comments or input. After the project team reviews these comments, update the goals and mission statement as necessary. Repeat this cycle until the key stakeholders agree on a set of goal statements. If you can't reach agreement then ask senior management, including the sponsor, to be arbitrators.

The goals and mission statement will be incorporated into your business case in step 5. However, you may want to get buy-in from the PEB at this point, before you define the expected benefits in step 4, to make sure you are on the right track.

Summary

Reviewing and agreeing on the goals and mission statement for your CMDB project is an important step. Going through this process will ensure all project contributors and stakeholders are on the same page as to what you plan to achieve, why, and how you will get there.

Step 4

Review and Define Benefits

Goals and Objectives

This step involves linking the problems and goals identified in the previous step to the specific benefits expected from your CMDB project. By improving processes, the CMDB will help you provide more consistent services to the business. It's important to involve key users and stakeholders in the process of linking problems to goals, and then to specific benefits, so that you gain user support. At the end of this step, you'll have detailed information that you can use to build a business case.

Overview

If you are implementing ITIL, keep in mind the V3 service management lifecycle. All of the processes described in this chapter support one or more of the service lifecycle stages.

You will need to be able to specify the benefits of your CMDB project so you can get it funded. The CMDB project may be an obvious win to the project team, but many good project proposals don't get funded. To build a hard-hitting and obvious business case, you must link the goals outlined in the previous step 3, "Create and Agree on CMDB Goals and Mission Statement," to particular anticipated benefits.

In this step, you will review a broad range of potential benefits of a general CMDB project, and then narrow them down to those things that relate to the specific goals and expected impact of your project. Involve and win the support of the key stakeholders that own the processes that will be improved by the CMDB. It will be difficult to get the project funded if your key stakeholders are not supportive. Involving them early and throughout the development of goals, mission, and benefits is a key to success.

Task-by-Task Guide

Figure 4.1 shows the tasks for reviewing and defining the benefits of the CMDB.

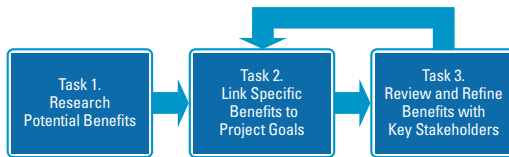


Figure 4.1. Tasks for reviewing and defining the benefits of the CMDB

Task 1. Research Potential Benefits

A CMDB can create a broad range of potential benefits. Knowing the potential benefits will help the CMDB project team identify the specific benefits of the CMDB project in your organization as you work with key stakeholders in various IT functions. You need to understand how the CMDB changes and improves organizational practices, so that you can articulate the benefits of a CMDB as they relate to your specific goals.

In general, the benefits of a CMDB are related to better access to information. The CMDB helps consolidate and federate data that previously was distributed across the IT organization. Through a federated CMDB, the project will connect those who consume information with those who produce the information.

The CMDB also helps improve data accuracy. The ongoing collection, reconciliation, and maintenance of CI data make the data dependable and usable. Putting supporting processes in place is mandatory. Identifying owners of specific types of CI data also helps. Finally, a federated CMDB can provide information to improve existing processes and also enable new capabilities that were not previously possible with disparate data stores.

To understand the potential benefits of a CMDB, you should research various sources of general information about how a CMDB enhances IT processes. Consider hard and soft benefits for each process. The following summary describes the potential benefits you can expect from a CMDB.

Incident management. The CMDB provides a rich source of information that you can use to turbocharge incident management. Incident managers can quickly access CI status, determine impact by reviewing the relationships between CIs and the business applications they support, and identify related CIs to restore service. CMDB data can automatically populate incident records. Also, the ability to roll back to a known good configuration baseline enables an effective service restoration strategy.

- » **Potential hard benefits:** Higher first-fix rate, less time and aggravation asking questions to assess the situation with the impacted user, reduced mean time to restore, lower incident management training costs
- » **Potential soft benefits:** Improved customer satisfaction because you can avoid asking them questions about the configuration of their system and you can provide faster response time, improved service-personnel morale, and improved business perception of IT at a key interface point with business users

Problem management. The CMDB gives you a rich source of data for proactive problem management, accelerating and simplifying root-cause analysis and problem resolution. It provides the immediate status of CIs affected by the problem. The CMDB links incidents to problems, and helps you visualize the problem CI and related CIs and their dependencies. It also shows the history of changes that may have caused the problem. The CMDB enables proactive problem prevention by analyzing aggregate problem data and trending information for a specific CI type, so you can identify asset classes that should be replaced to avoid escalating incidents. The CMDB may automatically populate problem records. The CMDB provides CI ownership information as well.

- » **Potential hard benefits:** Faster root-cause analysis and problem resolution, problem avoidance through trending and CI class analysis, lower repeat incident rate, more aggressive vendor discount when armed with aggregate failure data
- » **Potential soft benefits:** Improved customer satisfaction through proactive problem management

Change management. The CMDB provides a range of information that significantly improves the change management process, including better change-request risk assessment; identification of high-risk changes based on past success rate of similar CI types; and understanding upstream and downstream dependencies with other infrastructure components. The CMDB also enables forward scheduling of multiple changes that all relate to dependent CIs. The CMDB links user information to CIs, facilitating proactive and focused communication to users about pending changes. The CMDB shows the CIs that are in the change management process and reflects the status immediately after a change.

- » **Potential hard benefits:** Higher change success rate, lower overhead cost per change, lower risk of system outage caused by change, better coordinated change planning during maintenance windows
- » **Potential soft benefits:** Accelerated change management through a well-controlled process, higher customer satisfaction due to better communication, faster responsiveness to business changes

Configuration management. The CMDB is integral to configuration management, enabling the consistent, accurate, and cost-effective identification, control, status accounting, and verification of all CIs in the CMDB. Detailed and effective configuration versioning that enables an effective rollback or rebuild strategy is almost impossible without a CMDB. Without the ongoing update of CI attribute data through discovery and other sources reconciled in a CMDB, it is difficult to keep accurate information about the wide range of CIs in the production environment. Current configuration information facilitates automated rollout and update of security patches and application changes. A CMDB also enables version compatibility management and upgrade planning.

- » **Potential hard benefits:** Low-risk infrastructure upgrades and changes, increased security, reduced service outages
- » **Potential soft benefits:** Reduced security patch and application upgrade burnout

Release management. CMDB information supports automated rollout across distributed locations by providing accurate, detailed information about hardware, software, and current configurations and their compatibility with changes that are incorporated in a release. This CMDB information

Stage 1. Assemble the Project Team and Define the Project

also supports back-out procedures. The CMDB keeps version details for software, verifies tested configurations, and enables project scheduling. The relationship data in the CMDB can help you assess the potential system and financial impact of a planned release, and enables you to link CI changes together as a release and maintain the release-rollout status.

- » **Potential hard benefits:** Reduced release cost by enabling greater automation, reduced service impact, reduced business risk
- » **Potential soft benefits:** Reduced staff burnout due to manual software and patch installation

Service desk. The CMDB enables significant improvement in a range of service desk functions by providing detailed information about CIs related to service requests. Information about CI status, current configuration, configuration baseline, dependencies to other CIs and to business services, and planned changes all can help service desk managers meet service requests. The CMDB also provides the data the service desk needs to notify users of outages and the status of problem resolution.

- » **Potential hard benefits:** Reduced service costs; improved service levels by reducing errors, reducing manual data collection, and reducing the risk of failure due to changes that affect vital business functions
- » **Potential soft benefits:** Improved customer satisfaction, improved service manager morale

Service level management. The CMDB allows end-to-end service level management, providing detailed information about CIs, their relationships to each other, and their relationships to the underlying infrastructure linked back to IT services. It provides CI relationship data that links service level agreements (SLAs) to customers and to all related CIs that enable the service. It allows dynamic referencing of SLA components. The CMDB also enables operating level agreements (with internal IT groups and external service providers) and underpinning contracts (with external service providers) showing customer ownership. The CMDB provides a rich source of data for a data-driven service improvement program.

- » **Potential hard benefits:** Provides data that enables IT organizations to write and meet comprehensive SLAs
- » **Potential soft benefits:** Improved customer satisfaction through better reporting and confidence that service levels will be met

Financial management. The CMDB provides information that is critical to the effective financial management of IT. It contains a complete list of CIs, from which you easily can produce expected maintenance costs and license fees, maintenance contracts, license renewal dates, and CI replacement costs. Unlicensed copies of software can be identified easily, for erasure or destruction. The CMDB also enables service-based cost calculations by linking delivered services to supporting CIs. Detailed dependency data enables accurate charge-back programs, provides a vital source for inventory and asset audits, and serves as an important tool for service-based budgeting and forecasting. The CMDB also creates links to the enterprise resource planning (ERP) system holding the fixed asset register.

- » **Potential hard benefits:** Reduced maintenance contract costs, reduced license fees, more accurate service costing, charge-back to IT users
- » **Potential soft benefits:** Understanding true cost of delivering services

Business continuity management. The CMDB stores information about the IT infrastructure components, their configurations, and their dependencies to each other and key business processes. It also identifies the priority and the agreed-upon minimum level of business operation following a major service disruption. The CMDB contains vital recovery data and shows how a change in CI could mean a change in continuity requirements, such as when a low-priority system becomes higher priority due to a change of system functionality. The CMDB configuration baseline and CI current status information prevent the aging of response plans by providing ongoing updates of detailed infrastructure information, including CI and business service dependencies. The CMDB is also a source of feedback to customers during outages, and shows the status of CIs as they become active after an outage.

- » **Potential hard benefits:** Significantly faster disaster recovery time
- » **Potential soft benefits:** Higher business confidence in IT disaster recovery plans

Availability management. The CMDB provides a central information repository that links availability, reliability, and maintainability of services to the underlying IT components. The CMDB provides important business impact data, shows related components in an availability string, provides risk-analysis data, and helps isolate which CIs are the root cause of availability failures. Understanding dependency relationships also greatly simplifies linking CIs to the business process or service that is dependent on the IT infrastructure. This knowledge provides a great advantage by allowing IT to manage and prioritize work based on business impact. The CMDB also links IT components to service level agreements, operating level agreements, and underpinning contracts.

- » **Potential hard benefits:** Reduced service outages
- » **Potential soft benefits:** Improved prioritization of IT resources to business requirements, improved customer satisfaction and confidence in IT

Capacity management. The CMDB is essential for comprehensive business capacity management, service capacity management, and resource capacity management and modeling. Information about CIs, their relationships with each other, and their relationships to business functions is necessary for automated capacity management and real-time computing frameworks. The CMDB shows related CIs in a capacity grouping, provides vital risk-analysis data, and reduces the time needed to resolve capacity-related incidents and problems. The CMDB relationship data also helps prioritize capacity based on business needs.

- » **Potential hard benefits:** Reduced cost of redundant systems by capacity planning at the group or organization level instead of the individual system level, reduced time to resolve capacity-related incidents and problems
- » **Potential soft benefits:** Higher confidence in high availability

In addition to the ITIL service management processes, organizations perform many other IT business processes that rely on accurate IT service and component data. You should understand how these processes might need IT service and component data from the CMDB. You also should know which processes are most critical and therefore worth your effort to collect and maintain CI data, including the correct links and relationships.

THE IMPORTANCE OF THE CMDB AND THE CONFIGURATION MANAGEMENT PROCESS

Service operation ensures that your customers have access to appropriate IT services that enable the business. Service operation also ensures consistent functioning of the IT infrastructure upon which the business depends. With an effective and accurate CMDB, you can greatly enhance service operation. The service desk is a key part of your service operation strategy, because it provides your IT organization's face to the customer. Service operation processes help you deliver high-quality service, enabling consistent availability of the applications that customers want and more immediate response time to failures if they occur. Without configuration management, these service operation functions cost more and deliver less value since you must repeat steps to solicit or hunt down component information.

Service operation processes are more effective when they have accurate component-level information. However, these processes require access to information created in many places and functions within IT. Consolidating all of this information in a single CMDB enables faster resolution of problems, higher success rates for changes, and fewer unplanned outages. You'll likely have happier customers, as you're more likely to achieve higher service levels, and you'll also reduce the labor required to continually hunt down component-level information.

Many service operation processes cannot succeed without accurate configuration information. However, this information is created in a broad range of places and functions within IT. Consolidating all of the information in a single CMDB enables system-level capacity and availability management, continuity and risk planning, and better overall financial management of IT resources. You also can more effectively utilize scarce IT resources to improve service levels and service quality. As a result, you can spend less on IT back-office functions and more on new strategic initiatives to better deliver service.

Asset management. The CMDB provides a continually updated relational data repository for CIs that are considered IT assets. Those responsible for IT asset management can use the CMDB to understand the relationship of assets to organizational units, employees, cost centers, solutions in use, and so forth. This also allows you to understand not only what CIs are considered assets, but also who is using what assets, where the assets are located, who is paying for the assets, and what solutions are running on those assets. The CMDB enables you to automatically reconcile and manage data about your deployed assets and their dependencies with business services.

- » **Potential hard benefits:** Lower asset TCO and acquisition costs, elimination of unnecessary duplicate asset purchases, more efficient allocation, more accurate budgeting and planning
- » **Potential soft benefits:** More proactive control and greater visibility into your IT assets throughout their operational lifecycles

Project management. The CMDB, along with a change and release management process, provides the mechanism to identify, plan, track, update, and monitor the projects that create new CIs, modify CIs, or deploy instances of CIs. Having change and configuration management integrated into the project management lifecycle is critical to ensuring a smooth project-to-production transition and accurate CI status.

- » **Potential hard benefits:** Improved success rate and reduced costs during projects
- » **Potential soft benefits:** Greater staff and customer satisfaction because of smoother implementation of projects

Service performance and quality management. The CMDB provides the ability to link incidents and problems to services and support groups, so you can view service, resource, and organizational performance data. A well-defined CMDB is a cornerstone of the information to support service value reporting, service planning, and scorecards.

- » **Potential hard benefits:** Better understanding of performance to improve efficiency and reduce costs
- » **Potential soft benefits:** Improved customer satisfaction through higher-quality service

Contract management. The CMDB provides necessary information to effectively manage a contract with a service vendor. Such information includes actual performance tracked against contracted targets, invoice and payment details, contract time frames, and renewal information. This information is readily available for incident-by-incident contract management as well as overall performance.

- » **Potential hard benefits:** Reduced costs through more effective management
- » **Potential soft benefits:** Smoother vendor relations

Human resource management (training). The CMDB enables you to track staff requests for information (such as tracking “how-do-I” questions against services and staff) through the incident management process and to identify training requirements.

- » **Potential hard benefits:** Well-trained, well-informed staff
- » **Potential soft benefits:** Greater employee satisfaction

Audit, governance, compliance, and control. The CMDB provides an essential repository of control-related data useful for both internal and external audits. The Control Objectives for Information and related Technology (COBIT) framework, for example, recommends IT controls that can effectively leverage information from a CMDB. COBIT is an industry-recognized control framework that provides a set of high-level control objectives for IT processes and is used to assess the control system of an IT environment. See step 7, “Review and Select Supporting Best Practices,” for a more detailed discussion of this topic.

- » **Potential hard benefits:** Improved ability to implement IT controls required to meet various regulations that impact IT operations
- » **Potential soft benefits:** Higher confidence in audit and compliance capabilities

Task 2. Link Specific Benefits to Project Goals

At this point, the benefit analysis moves from the realm of the generally possible to the realm of the specific and likely. The project team should now work with key service and process owners and stakeholders to pare down the list of potential benefits to those that are related to the specific goals of the project. The team also should quantify the impact of the CMDB on their key process measures.

Stage 1. Assemble the Project Team and Define the Project

You should be specific about the potential hard and soft benefits, so you can create a crisp picture of the future state of the organization if the CMDB project is implemented. You need to figure out what will change in the organization with the CMDB in place, and estimate the impact by identifying improvements to key measures.

Your hard benefit estimates will turn into cost-reduction estimates in the ROI analysis. Your soft benefit estimates will help sell the CMDB project to all affected IT services and build enthusiasm for the project.

Your estimates don't have to be perfect. But they do need to make sense to all key stakeholders, the PEB, and the project sponsor who will approve funding and resources for the project.

To create the logic that ties goals to benefits, and then to key measures, you may need to interview service and process owners for their reaction and support. At the end of this task, you should clearly link their problems to project goals, link the goals to the specific benefits, and link the benefits to the quantifiable impact that you will include in a business case.

For those benefits that you cannot easily quantify the impact, you must include as much relevant information and data as you can, such as:

- » Endorsements from staff and managers
- » Examples from other companies
- » Events reported in the press
- » Examples of the problems or cost of the current state
- » Potential impact of an incident or event if a CMDB does not exist
- » Results of surveys and interviews
- » Forward projections of key trends and rates

For each goal identified in step 3, "Create and Agree on CMDB Goals and Mission Statement," you should complete an impact estimate assessment that identifies the goal, the hard and soft benefits, and the impact. (See figure 4.2.)

| Goal | Benefits | Impact |
|--|--|--|
| Provide accurate information on configurations and their documentation to support all other service management processes | <ul style="list-style-type: none">• Hard benefits: improve process integration, communication, decision making, and compliance• Hard benefits: experience fewer failed changes, increase availability, reduce incident rate, and improve first-fix rate• Soft benefit: improve customer satisfaction | <ul style="list-style-type: none">• 30% cost savings as a result of reduction in failed changes; 20% cost savings as a result of increased availability; 35% reduction in incidents; 40% cost savings realized by incidents solved by first level support staff• IT perceived as supporting and enabling business |

Figure 4.2. Example impact estimate assessment

You should then describe the logic and reasoning behind the link between the goal, benefits, and impact. For example, an explanation might read as follows:

The current first-fix rate is limited by lack of visibility into recent changes. A recent audit showed that 55 percent of incidents were related to changes made within the previous 48 hours. Real-time visibility by service desk personnel to change history tied to the CI through the CMDB will help reduce the time spent asking the user what the state of his or her system is and will help identify potential causes. Other organizations surveyed by a leading analyst saw a 10 percent improvement in the first-fix rate by adding change history to the incident management process. We expect a greater impact and have estimated a 15 percent improvement. Also, a recent customer survey showed that technicians asking the user detailed configuration information they didn't know how to find (such as operating system, memory, etc.) was the third highest driver of dissatisfaction. We anticipate a significant improvement in user satisfaction.

Complete a similar logical trail from problem to benefit and then to an estimated impact for each key goal identified in step 3, "Create and Agree on CMDB Goals and Mission Statement."

Task 3. Review and Refine Benefits with Key Stakeholders

The final task in this step is to review and refine your benefits with key stakeholders. Distribute the list of benefits to key staff for their comments and input. You should have already involved key service and process owners in the development of goals and the specification of benefits and impact. Now is the time to have them review the full set of materials you have developed, and to ask for their support. Your CMDB project will benefit them, so make sure they agree with the goals and logic that lead to benefits and impact.

Key service and process owners need to:

- » Feel ownership of the goals
- » Agree with your logic that links specific benefits and impact
- » Buy into the changes that will enable achieving goals

Keep in mind that a functional owner may agree with all the goals, benefits, and impact, but not have the resource bandwidth to prioritize the changes needed to achieve the results for his or her group.

Include any other staff who will be influenced or affected by the CMDB when it is implemented. Review the comments and input as you receive them. To reduce the chances of conflict later in this project, contact all of the recipients — especially those who did not provide any comments or input.

Stage 1. Assemble the Project Team and Define the Project

This task is significant, because the whole project could sink or swim right here. After the project team reviews these comments, you should update the benefits as required. Then loop back to tie the goals to the revised benefits and impact. You should repeat this cycle until you can clearly articulate how the related functions that rely on the CMDB will change.

Schedule meetings with the key management staff and the project sponsors, and present the facts in a clear and concise manner. Do not try to over-sell your case. Endorse it, without losing sight of your goals, but if the case is strong the facts should be enough. Expect some level of rejection, and also expect to re-estimate some of your benefits.

Summary

Remember that while you promote the benefits of a CMDB project in this step, you will be expected to deliver those benefits by the end of the project. Be careful. Don't exaggerate your case and create nightmares later by promising benefits that are not realistic or likely to be achieved.

Step 5

Build a Business Case

Goals and Objectives

Data collected during the previous steps now must be put into a business case that outlines the problem, solution, and expected benefits. You must present this report to the PEB before presenting it to the sponsor for the budget and resource approval needed to proceed with the project. You may need to circle back and modify the goals, mission statement, and expected benefits as a direct result of getting approval.

Overview

This step is critical for your CMDB project. You are asking for approval for funding and resources to move forward. If your business case is not logical, simple, and compelling, you likely will not get the green light to proceed. Competition for scarce IT resources is intense. In the current IT budget climate, multiple project teams will have compelling proposals with strong ROI, but not all will get approval to move forward.

Although you haven't defined all the requirements, selected a solution, and scoped the exact costs of implementation, you should have enough information to make reasonable estimates to evaluate expected savings and project costs. If you need a more detailed project ROI at this point, turn to step 19, "Calculate Project ROI," for more details on how to construct a full-project cash-flow analysis and a project ROI.

If your work so far is sound and you have aligned current problems with the mission, goals, and benefits, you will be able to put together a strong business case that should result in approval. To avoid surprises, involve both key stakeholders and decision makers in the process of clarifying goals and estimating benefits. If you have involved the right people as you developed consensus on the goals, mission statement, benefits, and impact, you should be able to gain consensus that the CMDB project is a top priority for the organization.

Task-by-Task Guide

Figure 5.1 shows the tasks for developing a business case.

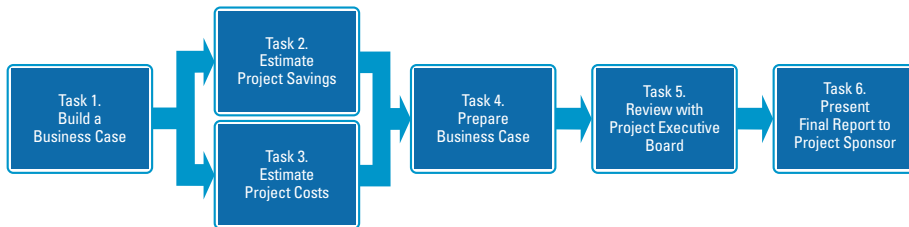


Figure 5.1. Tasks for developing a business case

Task 1. Build a Business Case

To get the green light for a CMDB project, you need a clear alignment of problem, solution, and benefits. If your project does not align with what the organization is trying to accomplish overall, other better-aligned projects will be prioritized over the CMDB project.

A solid business case for a CMDB must include the following elements:

- » Executive summary
- » Problem statement
- » Solution summary
- » Goals and benefits
- » Cost benefit analysis
- » Metrics and measurements

Executive summary. The executive summary provides all of the information required to understand the basics of the CMDB project in a clear, simple, high-level package. It should include, in one or two pages:

- » **Strategic alignment** — Start with the reasons you are proposing a CMDB. Highlight the alignment with corporate strategy or summarize the changes that triggered the consideration of the CMDB project. Showing relevance to the organization is the first step in showing value.
- » **Project overview** — Summarize the problem that the CMDB will solve. Summarize the CMDB solution being proposed. Highlight the processes and services that will benefit from CMDB data. Limit your description primarily to business terms, rather than technical terms.
- » **Impact** — Identify the stakeholders in the project. Summarize the key metrics and goals that will be used to measure success. Outline who the CMDB will affect and how. Identify key risks and outline assumptions. Finally, outline the timeline of the implementation and expected benefits.
- » **Cost benefit analysis** — Summarize your estimated costs and benefits. Highlight key assumptions about project costs.

- » **Metrics and Measurements** — Describe how you will measure whether the CMDB project is successful, including what metrics you will use.

Problem statement. The executive summary is a high-level overview. The problem statement is the introduction to the logic of your CMDB business case. Too often, technology initiatives such as the CMDB are perceived as solutions in search of a problem. This section gives you the opportunity to provide context and relevance. Outline what will be made better by a CMDB and what progress would be limited by doing nothing. The problem description should include:

- » **Current situation** — Describe how the organization currently functions without a CMDB. This section will be as detailed as required and can include process diagrams as well as statements from stakeholders in the business or technology group about the challenges they currently face because various siloed IT functions work without a CMDB. If appropriate, include cost and resource requirements to support the current situation. Remember to mention regulatory requirements that are difficult to support without centralized data access.
- » **Best practices** — Outline ITIL best practices or research about how other organizations perform the same tasks. This information highlights the scale of the potential improvement.
- » **Gap analysis** — Differentiate current practices from best practices to help build consensus. Don't generalize, such as "The CMDB will help centralize information." Be specific. For example, "The CMDB will help the service desk obtain real-time information about recent changes to CIs that will help improve first-fix rate and therefore increase service availability."

Solution summary. Summarize your results from obtaining CMDB knowledge in step 2 and defining the expected benefits in step 4. You haven't yet compiled a complete list of requirements or selected specific technologies, but you should summarize the functional pieces you might need. The solution description might include:

- » **Solution overview** — Provide a concise description of the proposed CMDB solution, describe the IT processes and services that will be improved by the CMDB, and highlight the business process changes that will be required.
- » **Cost estimates** — Describe the estimated costs to implement the solution, including future requirements for new or modified procedures in other departments.
- » **Alternatives considered** — Quickly summarize the other solutions you considered and the key reasons you did not recommend them. You should consider and document a minimum of three scenarios or alternatives to add credibility: 1) Do nothing, maintain current state. Sometimes it is less expensive overall to not solve a problem, especially if the cost of the solution is greater than the expected benefits. 2) The recommended solution. 3) A second alternative, such as expanding the function of an existing solution — an asset management repository, for example. It is important to show that existing systems were considered and if rejected, why.
- » **Project timeline** — The timeline describes the key milestones and stages of the project. Although you will still need to define some aspects of the project plan, showing the expected time frames will set project expectations properly.
- » **Risk analysis** — List your assumptions. Because each estimate of cost or benefits in the business case will be based on a set of assumptions, you should document these assumptions and note their potential impact on cost or benefit estimates. For example, if integration is more difficult than expected, it could increase costs 5 percent.

Stage 1. Assemble the Project Team and Define the Project

Goals and benefits. In this section, you can include a summary of the goals and mission you developed in step 3 and the estimated benefits and impact you created in step 4. If the project justification assumes certain performance improvements, everyone affected by the project should clearly understand the business case and commit to the goals so the project achieves the expected ROI. For each expected benefit and impact, list the following:

- » **Specific metric** — Change success rate, for example
- » **Definition of that metric** — Number of successful changes divided by number of planned changes
- » **Current measure** — For example, 80 percent
- » **Goal after implementation** — For example, 90 percent

Although you can include metrics in both the current problem and proposed solution sections of the business case, consolidating them here ensures agreement on the goals of the project.

Metrics and measurements. Be sure to explain in detail how you plan to measure the success of the project and determine whether the business case was valid. Refer to step 25 for more details on how to establish appropriate metrics.

Cost benefit analysis. Summarize the estimated project costs and the estimated savings associated with the benefits. You have not yet defined all of the requirements or the complete scope of solution, so you can't have exact cost estimates. However, you do need budgetary estimates of the project to weigh against the expected benefits.

If the solution will save \$1 million in improved IT process efficiency, but will cost \$2 million to implement, then you have a problem. However, if you show estimated costs that are reasonable in comparison to the expected benefits, then you can make a strong case for moving forward with the project.

Task 2. Estimate Project Savings

In this task, you will plug in your impact estimate from step 4 and estimate the savings associated with the benefits. The CMDB is typically viewed as an enabler, so the savings of the CMDB project are derived from the improvements to processes and services that will benefit from the CMDB.

Work with someone in your finance organization to convert each benefit and impact from a performance number into a dollar figure. Savings in IT come from reducing costs associated with people, third-party services, hardware, software, and facilities.

For each performance improvement listed in the impact assessment in step 4, you should create a simple equation that links the benefit to a cost reduction. Convert performance improvements to an estimated cost reduction unless higher service levels improve revenue.

For the change example listed in task 1:

- » **Specific metric** — Change success rate, for example
- » **Definition of that metric** — Number of successful changes divided by number of planned changes
- » **Current measure** — For example, 80 percent
- » **Goal after implementation** — For example, 90 percent

Someone in your finance organization will help change that improvement into a dollar figure, such as:

- » Reduce failed changes from 20 percent to 10 percent per year
- » 3,000 total changes per year x 20 percent = 600 current failed changes per year
- » Each failed change requires 40 hours x \$90 per hour = \$3,600
- » Benefit = 300 failed changes avoided x 40 hours x \$90 = \$1.08 million annual savings

A good way to summarize estimated savings is to forecast savings over a period of time, and then itemize based on expected efficiency improvements in various IT functions that will benefit from the CMDB. (See figure 5.2.)

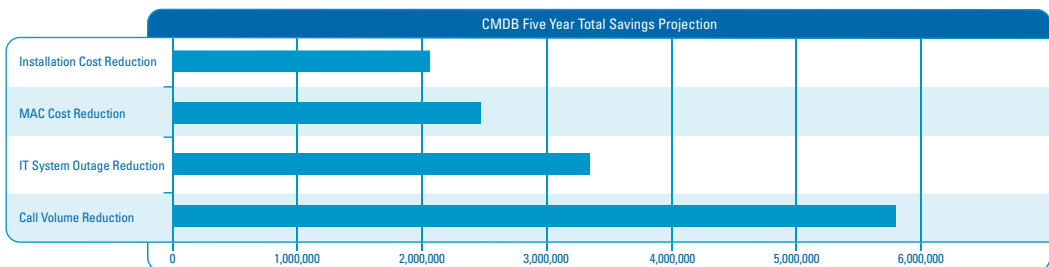


Figure 5.2. Estimated savings projection

If you need a more detailed benefit cash-flow projection for a preliminary project ROI, see the benefit cash-flow projections shown in step 19, “Calculate Project ROI.”

Task 3. Estimate Project Costs

You will also need a total estimated cost for your CMDB project so that you can compare the cost to estimated benefits. Because you have not yet selected a vendor or negotiated a price, use your judgment and past experience to develop the estimates. Talk to your peers at other companies that have implemented a CMDB and get their input on project cost estimates.

Stage 1. Assemble the Project Team and Define the Project

Estimated costs should include not only initial implementation costs, but also the annual cost of running and maintaining the solution. Typical costs you should consider in your estimate include the following:

- » **Project team costs** — Project team costs (which you may have calculated in step 1) directly relate to the cost of staffing and managing the project and do not include costs for technology or tools.
- » **Software licenses** — This includes the cost of licenses for the full stack of technology in the CMDB solution, from the database up through the CMDB applications and other supporting technology.
- » **Hardware** — These costs are related to hardware and facilities. You may need to add the overhead costs of data-center rack space and network bandwidth.
- » **Annual maintenance** — This cost is the total cost of hardware and software maintenance based on a specific contracted period, and may also include the ongoing facilities and network usage costs.
- » **Professional services** — This includes all costs associated with the team performing the work required to implement and integrate the solution.
- » **Training** — The CMDB will affect multiple groups within IT. Budget for and conduct training if you want users to adopt new solutions in their existing process flows.
- » **Administration** — Capture the amount of system-administrator time needed for ongoing support and improvement of the system. Express this amount in terms of FullTime Equivalent (FTE). Someone in your finance organization can help convert people time to dollars.

If you need a more detailed cost cash-flow projection for a preliminary project ROI, refer to the cost cash-flow projections in step 19, “Calculate Project ROI.” You may need to estimate the costs in each year over a period of time.

Task 4. Prepare Business Case

Now it's time to put this all together. Follow the business case outline in task 1 and write a business case. Try to keep the total document under 15 pages — if you have too many pages, no one will read it all. Distribute and share the draft with key stakeholders. Check your assumptions. Show your measures and conversion into hard savings. Also double-check soft benefits. Describe how you will measure whether your business case is valid.

Make sure the service and process owners accept your logic, and ensure that they understand the timeline and expected time to benefits. Get their commitment that they will prioritize the CMDB project if funded. Make sure it's a two-way conversation. Ask the service and process owners for suggested improvements. Incorporate their feedback to cement their alignment.

Task 5. Review with Project Executive Board

After you make the rounds to ensure everyone supports your CMDB project business case, then share it with the PEB. Ask questions to get their input and buy-in. Highlight the “makes sense” approach and simply ask them if it makes sense. Request a sanity check and integrate the PEB’s suggested changes before going for final approval.

Task 6. Present Final Report to Project Sponsor

At this point, present the executive summary. Talk through the contents of the executive summary and highlight the assumptions, risks, resources, and organizational impact.

Remember that many good projects don’t get funded each year because of resource limitations. Someone must compare and evaluate competing initiatives. The CMDB may be an obvious win to you and the project team, but don’t assume it is an obvious choice when you present the report to get final approval.

Be sure to highlight how the project aligns with and supports what are considered top-priority initiatives for the IT organization. Show how the CMDB will improve what matters most to the IT executive team.

Summary

A good business case contains the basic elements of problem, solution, benefits, metrics and measurements, and impact on the organization. It doesn’t have to be an exhaustive effort. The previous steps were building blocks to collect most of the information to develop your business case. By pulling the key information together in a logical flow, with preliminary savings and cost estimates, you have what it takes to present and gain approval from key stakeholders, the PEB, and the CMDB project sponsor.

STAGE 2

Define Requirements and Create IT Service Model Blueprint

Stage 2. Define Requirements and Create IT Service Model Blueprint

During stage 2 of your CMDB project, you will gather and document the various requirements that you will use to select the CMDB solution and guide the implementation. The overall approach is to clearly specify what is required for your CMDB.

In stage 1, you collected information through steps that built up to a business case that was approved by the PEB and project sponsor. In stage 2, you will work with key stakeholders to document and obtain approval on their specific subset of requirements for each step. These stakeholders include service and process owners that will leverage the CMDB, as well as those responsible for compliance and IT asset management. You will also capture CI-level requirements that specify what CI relationships will be maintained, including relationships for CIs that compose the IT service model. Using this information, you will define the specific CI attributes that will be stored in the CMDB. You will then compile all the various approved requirements into a comprehensive requirements document that you will present to the PEB for final approval. Finally, you will translate the requirements into a consolidated IT service model blueprint which you will present to the PEB for approval.

Figure 1 shows the steps for defining your CMDB requirements.

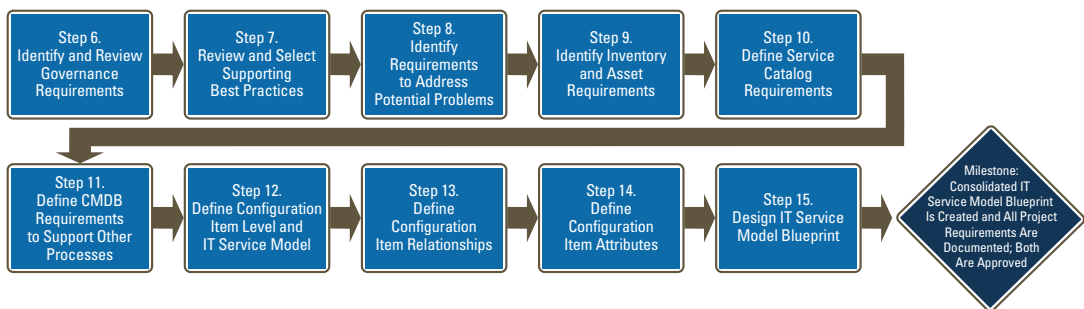


Figure 1. Steps for defining requirements

Each step contains specific goals that you should meet before moving forward. You may want to use the following as a checklist.

Step 6.

Identify and Review Governance Requirements

Goal: A CMDB can enhance a range of compliance activities in IT. Identify corporate and IT governance requirements and regulatory compliance efforts that you want to improve with the CMDB project, and document the CMDB requirements.

Completed: ____

Step 7.

Review and Select Supporting Best Practices

Goal: Start with the IT processes that you have targeted for improvement through access to the CMDB, and identify best practices that these IT processes can use. You can accelerate achievement of the expected benefits related to process improvements by leveraging industry best practices.

Completed: ____

Step 8.

Identify Requirements to Address Potential Problems

Goal: Identify problems encountered on previous CMDB projects and turn lessons learned into project requirements. Many potential implementation and support issues can be avoided with a little effort at this stage.

Completed: ____

Step 9.

Identify Inventory and Asset Requirements

Goal: Pinpoint how inventory and asset management requirements will affect the CMDB project. Asset, inventory, and configuration management are similar, but different enough to call out specific requirements.

Completed: ____

Stage 2. Define Requirements and Create IT Service Model Blueprint

Step 10.

Define Service Catalog Requirements

Goal: Identify requirements that specify how the service catalog will leverage the relationships between services and the underlying CIs. Understanding which CIs relate to a particular service enables you to better meet SLAs and allows you to conduct service-based costing.

Completed: ____

Step 11.

Define CMDB Requirements to Support Other Processes

Goal: Turn IT process touch points with other IT functions into specific requirements. The requirements must reflect how other groups will interact with the CMDB and any special needs these groups have.

Completed: ____

Step 12.

Define Configuration Item Level and IT Service Model

Goal: Define the optimum level for CIs — both service CIs and infrastructure CIs — in your CMDB. This step helps determine the overall breadth and depth of the structure of your CMDB data model.

Completed: ____

Step 13.

Define Configuration Item Relationships

Goal: Identify the specific CI relationship data that you will maintain in the CMDB. Relationship data may be difficult to maintain, but it has significant potential benefit to a range of IT processes and services.

Completed: ____

Step 14.

Define Configuration Item Attributes

Goal: Define CI attributes that you will maintain in the CMDB versus those that will reside in federated data stores.

Completed: ____

Step 15.

Design IT Service Model Blueprint

Goal: Construct the IT service model blueprint using the requirements you have identified for the service catalog, the IT business processes, and the IT service model design. Present to the PEB for approval.

Completed: ____

Milestone: Consolidated IT Service Model Blueprint Is Created and All Project Requirements Are Documented; Both Are Approved

You have constructed the IT service model blueprint based on the requirements you documented for the service catalog, the IT business processes, and the IT service model design. The PEB has approved both the requirements document and the blueprint.

Milestone achieved: ____

Step 6

Identify and Review Governance Requirements

Goals and Objectives

In this step, you will identify governance and regulatory requirements that may affect the overall CMDB project requirements. Corporate and IT governance and regulatory requirements include federal and local regulations as well as industry-related regulations that affect your IT organization. After reviewing all governance requirements with members of your organization that are responsible for compliance, you will specify supporting processes and related CMDB project requirements.

Overview

The CMDB can be an especially powerful tool to help you meet various regulatory and governance requirements that affect IT. Although no regulations require you to have a one, ISO/IEC 20000 sets the standard for all organizations to have a CMDB that is actively managed, reliable, and valid. Requirements of some common regulations can be met in a very simple way if a CMDB is part of the solution.

For example, Sarbanes-Oxley section 404 requires controls for processes that affect financial reporting. IT systems that support financial applications are “in scope” for those control requirements. Organizations may need an auditable change process for changes to those applications and underlying systems, but might choose to follow a more stringent process only for the applications and systems that are in scope. When a change request is accepted, it may be difficult to identify whether the CI is in scope for Sarbanes-Oxley requirements. If the organization has a CMDB and every CI has an attribute field that flags whether the CI is in scope for Sarbanes-Oxley, then the change process can be streamlined and have a different workflow for CIs related to regulations such as Sarbanes-Oxley that have more stringent change process requirements.

Similarly, the Health Insurance Portability and Accountability Act (HIPAA) requires specific control of systems that store personally identifiable patient information. Banking regulations require special control of IT systems.

Stage 2. Define Requirements and Create IT Service Model Blueprint

A CMDB can address a wide range of compliance issues if the system is designed with those requirements in mind. The CMDB will also hold the relationships between CIs, enabling better understanding of which change to a CI may affect another CI. One CI might have regulatory requirements and the other by itself might not, but by association it does. The CMDB highlights this association.

However, it is not the CMDB team's responsibility to identify all the laws your company must comply with. Your company likely has an audit or legal team that tracks changing laws and regulatory requirements. If regulations affect IT in any way, the CIO has probably already assigned someone from IT to be a part of the overall corporate governance and audit functions. That person or team is a key stakeholder that you need to meet with and brief on the CMDB project. This stakeholder might be an IT auditor specifically, an IT representative on the corporate internal audit team, or legal counsel.

It is your responsibility to find the key stakeholder in IT responsible for IT governance and regulatory control, and to explain the CMDB project. Let this stakeholder know you are collecting requirements so that the project can meet his or her needs, and let the stakeholder decide how the CMDB should be used to comply with regulations. You then document those requirements and make sure the solution meets that key stakeholder's needs.

A few words of caution. You are collecting these requirements after goals and expected benefits have been outlined, and after budgetary costs have been approved. If complying with regulatory requirements was not a goal identified in stage 1, you might want to skip this step. But check with these stakeholders first.

Task-by-Task Guide

Figure 6.1 shows the tasks for reviewing and identifying governance requirements.



Figure 6.1. Tasks for reviewing and identifying governance requirements

Task 1. Identify Key Governance and Regulatory Stakeholders

If meeting regulatory requirements is a goal of your CMDB project, then you should specify what the CMDB will do to help the corporate and IT governance and regulatory compliance efforts. There is a broad range of regulations that may or may not apply to your company, and that may or may not affect IT. Understanding all of the regulations that affect your organization is beyond the scope of the CMDB team's responsibility. However, the CMDB can provide a simple and powerful solution to address these requirements as they affect IT.

You should be able to start at the top of the IT organization and find out who has overall responsibility within IT for meeting these requirements. Your CIO will be well aware of many of these regulations, as they are a key point of discussion among the executive team and board of directors.

Find the IT resources that have been given responsibility for meeting these requirements in IT, and brief them on the CMDB project. Details of the governance and regulatory requirements may come from sources at three levels.

Regulations applicable to all companies. At the top level are the legal requirements applicable to all companies. Established by the country in which your company operates, these requirements include federal and state requirements in the United States and community and country requirements in Europe. The Sarbanes-Oxley Act is an example of federal governance; California Senate Bill 1386, applicable to companies conducting business in California, is an example of a state requirement.

Industry governance. The next level you need to consider is industry governance, where certain legislation applies just within your own sector. Examples of industry governance include 21 CFR 11 for the pharmaceutical industry, Basel II for companies in the banking industry operating in Europe, the Gramm-Leach-Bliley Act (GLBA) for financial services in the United States, HIPAA for healthcare organizations operating in the United States, the Federal Information Security Management Act (FISMA) in the United States, and Directive 95/46/EC in Europe.

Internal governance. The third level encompasses the internal, organization-specific governance rules as established by the board of governors or directors, the heads of departments, and so forth.

It's helpful to create a list of internal and external experts that can assist you in understanding governance and compliance requirements. This will facilitate the discussion necessary to develop a list of specific requirements for your CMDB solution.

Task 2. Scope Potential Use of a CMDB

Now you can build on the goals, benefits, and expected impacts you identified in stage 1. In this task, you want to specify how the CMDB will help your organization meet various regulatory requirements. The CMDB can help in several key ways.

Identify CIs in use. The CMDB may be considered the single source of reference for what is currently in use as part of the IT infrastructure. You need to be clear about the scope of what will be loaded as a CI in the initial implementation of the CMDB project, so that you can provide guidance to the IT audit team about the potential of using the CMDB to meet regulatory requirements. For example, if the initial scope includes servers and applications but not networking equipment, then the IT audit team might consider it a partial solution. If including network equipment makes it a full solution, consider adding network equipment CIs as a requirement for regulatory compliance.

USING THE CMDB TO ASSIST IN REGULATORY COMPLIANCE

The CMDB can help you achieve regulatory compliance. Greg Wilson of Wipro provides some insight.

IT executives are now focusing on what matters most to the business. This focus requires greater IT controls, better management of the configuration of infrastructure items, and an understanding of how infrastructure components link together as well as how they link to the critical business functions they support. The CMDB is the underlying solution to address these challenges.

An increasing number of regulations affect IT operations. For example, you probably already know that Sarbanes-Oxley requires IT to demonstrate processes that control all components that can affect corporate financial information. Other regulations require effective control of separation of duties and system access. Well documented, repeatable, and controlled processes reduce the risk of IT infrastructure failures. But then proving it to an auditor adds another layer of complexity.

A CMDB helps address some of the most challenging IT control requirements related to various regulations by providing a central repository of information about both infrastructure components and related users. The CMDB can help ensure that your organization maintains control processes and is ready for an audit. A CMDB contains all the elements of your business service (people, process, technology), which also reduces the ongoing cost of control and compliance.

Excerpt from "CMDB: The Answer to 4 Key Challenges Facing IT," VIEWPOINT, Focus on: CMDB, published by BMC Software.

Identify relationships to applications and business processes. One of the key challenges facing the IT audit team is to identify which equipment supports which processes. If a particular server supports a printer in the marketing department, that server might not be relevant. If a server supports an application that is in scope for an audited business process, or is the database server that contains information covered by a privacy law, then the CMDB can identify those relationships. The CMDB relationship data can be a huge asset for IT audit and compliance teams that need to know what infrastructure supports specific audited processes. CIs can include particular attribute data that flags a business process or application.

Configuration data. The configuration of databases and networks affects security and access rights. Many regulations clearly require the protection of certain kinds of data, such as personal information on patients in a healthcare facility. You must manage individuals' access to protected data, and perhaps even create an audit trail to show whether a breach has occurred. The CMDB can be a simple and powerful way to control and monitor configuration changes that can affect access rights.

Fixed-asset register. The finance organization might track details from purchases, copies of invoices, and purchase order requests, for all equipment that shows a value and depreciation expectations on the balance sheet. However, the detail held in the fixed-asset register (FAR) is normally only a few lines, or one line stating, for example, the purchase of 100 servers at a particular monetary value. During the annual audit, an auditor may request to see the physical, in-use asset. The CMDB can provide an accurate and detailed record of what is currently in use, where the asset is deployed physically, and who is using it.

Your CMDB does not need to contain all the details of all the governance and compliance information, but you must be aware of the elements that affect your ability to provide business services.

Task 3. Collect and Document Requirements

Review the governance and compliance requirements with the key regulatory stakeholders and the CMDB team to document specific requirements. Document the requirements in enough detail so that someone can select a solution and meet the requirement. If the CMDB is going to be the source of record about which CIs are in scope for various regulations, then that needs to be turned into a specific CI attribute data requirement. You will need to ask: How many fields? What is in each field? Is the data all in the CMDB or partially federated with an asset repository?

Together you can determine IT's responsibilities as well as what data should be held in the CMDB and the other IT-related databases (asset management, for example) to create the federated structure. Remember that these connections may extend beyond IT databases and into business databases, such as the FAR.

Task 4. Get PEB and Key Stakeholder Approvals

Create a presentation that highlights how the CMDB can help your company meet governance requirements. This presentation should be short — 30 minutes at the most — to communicate the main points and promote further discussion of the details and next steps. Schedule some time in the next PEB meeting, or call a meeting specifically to address governance requirements. The CMDB project is at an early stage, and governance requirements warrant special attention to ensure the project can proceed.

Be clear that you are asking for approval of the list of regulatory- and governance-specific requirements that will be passed to the team that architects, selects, and implements a solution. If the requirements are clear and thorough, you can avoid the “scope creep” that often plagues CMDB project teams as they move into the implementation stage.

Summary

Meeting regulatory requirements is likely a top priority for your company and IT executive team. The CMDB can provide a simple and powerful solution for many of the key challenges associated with meeting the IT-specific elements of these regulations. If your team has identified regulatory compliance as a project goal and has documented specific benefits in the business case, then meet with the key stakeholders to convert those benefits into specific project requirements.

Step 7

Review and Select Supporting Best Practices

Goals and Objectives

In this step, you will build on the expected benefits identified in step 4, “Review and Define Benefits,” and select the best practices that help service and process owners effectively use the CMDB to achieve the expected benefits that justified the project. You will need to review and assess various best practices that explain how to use a CMDB to maximum advantage. You will work with key stakeholders to identify the specific CMDB-related best practices they plan to follow, so that you can document specific process integration requirements in step 11, “Define CMDB Requirements to Support Other Processes.”

Overview

Why reinvent the wheel? Many organizations in the past have devised, often at great expense, some excellent practices and processes for the delivery of business services within their organization. Often these best practices were developed through necessity; in other cases, organizations developed their own best practices because they were not aware of already existing best practices.

This step will highlight some of the existing best practices and make recommendations for researching others, so you can identify specific changes that will help you achieve your project goals and expected benefits. It is one thing to identify an opportunity for process improvement and say that a particular measure will improve 10 percent, but it is another thing to actually make lasting changes that achieve the expected results. Best practices allow you to leverage what others have learned and accelerate your time to benefit.

Regulatory and standards requirements provide an opportunity for you to define best practices. Identifying IT best practices at this early stage of defining process requirements will help in two key ways. First, service and process owners can start to study best practices with their planned use of the CMDB, so they can prepare their organization for upcoming changes. Second, the CMDB team can document specific project requirements based on how the service and process owners plan to use the CMDB.

Stage 2. Define Requirements and Create IT Service Model Blueprint

For example, say you have identified a goal and benefit related to improving service desk response to incidents by giving technicians access to change history. Now you need to specify how that is going to happen. The obvious question is how do others do that? Do service desk managers look at changes related to a CI that have been made in the last day, week, or month? Do they look at changes just to the CI or also other CIs that have a dependency relationship identified? Does the relationship data automatically populate the service desk application? Or does the service desk technician need to log into the CMDB directly to find the information?

Best practice frameworks may not get to this level of detail, but they should be used to define the new process so that you can create specific requirements that guide both the selection of the solution and the eventual implementation. In the business case you developed in step 5, “Build a Business Case,” you should have identified current and future practices that will leverage the CMDB, and identified expected benefits and impact on specific measures. Reviewing and selecting best practices can help you get specific about what your future process will look like with a CMDB and can help you create your process integration requirements in step 11, “Define CMDB Requirements to Support Other Processes.”

Task-by-Task Guide

Figure 7.1 shows the tasks for reviewing and selecting supporting best practices.

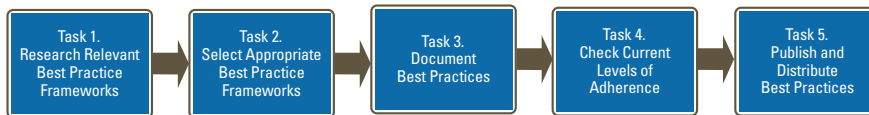


Figure 7.1. Tasks for reviewing and selecting supporting best practices

Task 1. Research Relevant Best Practice Frameworks

The purpose of this task is to review the best practice frameworks that may provide some guidance to your organization’s effective use of the CMDB.

Frameworks that you might leverage for CMDB-related best practices in your organization include ITIL, Application Services Library (ASL), Control Objectives for Information and related Technology (COBIT), New Generation Operations Systems and Software (NGOSS), enhanced Telecom Operations Map (eTOM), and any other frameworks developed by IT practitioners in various industries. You may also find helpful guidance in international IT service management standards, such as BS15000 and ISO/IEC 20000, or quality assurance methodologies, such as Six Sigma and Capability Maturity Model (CMM). Industry organizations, such as the itSMF, often produce guidance and additional best practices to augment the mainstream best practices. For example, itSMF follows ITIL, but for small businesses itSMF also produces guidance that is a simplified version of ITIL processes.

You shouldn’t try to boil the ocean here. In getting your business case approved, you moved past exploring everything a CMDB could do and focused specifically on what the CMDB team and key stakeholders expect the project will do for your organization. As a result, you’ve

identified the specific services and processes that your organization expects will benefit from using a CMDB. That is where you should focus your search for best practices. For example, a CMDB can be a great help in disaster recovery. However, if your team has not identified disaster recovery as a focus for the initial implementation, there is no reason for the CMDB team to research disaster recovery best practices related to a CMDB.

Keep in mind that various best practices may not use the same terminology. They may not call out a CMDB specifically. They may not use the term CI. They may, however, require information about the IT infrastructure to identify IT controls. Or, they may imply or require an accurate source of information so that you can measure and improve processes. Those are practices the CMDB can support.

Task 2. Select Appropriate Best Practice Frameworks

Now that you have a list of all the best practice frameworks available within your geography, country, and industry sector, review them based on the following characteristics:

- » Fit with goals and expected benefits of your project
- » Reference to, or specific guidance for, a CMDB (or equivalent, such as integrated federated metadata)
- » Maturity
- » Existing use in your organization
- » Popularity (number of companies aligned with it)
- » Professional bodies supporting or representing it
- » Breadth and depth of detail (process, roles and responsibilities, integration areas, etc.)
- » Overlap areas (complementary and conflict)
- » Professional training certifications available
- » Consultancies advising on it
- » Applications that automate the integrated solution and have best practices built in

You need to remember three key considerations. First, find out if some of these best practices already have momentum in your organization. Ask around and find out if there are pockets of implementation. If there are, it may make more sense to try to expand the use of a particular best practice that is already partly in use, than to call out the need for another new one. Keep in mind the costs of training and the time needed for the organization to absorb and adopt new frameworks. If your organization has already chosen a best practice such as ITIL, look for CMDB guidance there first.

Second, it is the service or process owner's job to select best practices that relate to his or her function, and implement process change related to the use of the new CMDB. Your role is to identify CMDB-related best practices that might help process owners specify the new way of doing things after the CMDB is in place. The service or process owner should select and drive the best practice that best fits that team's needs. Change is difficult, so remember that the users of the CMDB will make or break the project. The best practices and the new CMDB

THE BUSINESS OF GREAT SERVICE

ITIL is helping one automotive manufacturer refine and redefine service delivery to more than 300,000 company employees worldwide.

A large automotive manufacturing company recently implemented ITIL methodologies to refine its internal service operation processes. The company's IT director for enterprise operations management describes ITIL in plain language: "It's how you organize to operate an IT department — that's all."

After the company moved to a 100-percent IT outsourcing strategy, IT executives realized that a lack of standardization in departmental operations was going to make the job of managing multiple outsourcing vendors complex and inefficient. Realizing the challenge of harnessing and coordinating processes among its four primary technology vendors — not to mention a handful of secondary outsourcers — the company had little difficulty in choosing ITIL.

"We had the need to have a common lexicon across the suppliers, in terms of terminology and process. So we looked around, and ITIL had a lot of push behind it," noted the IT director.

As with any methodology, you need to modify it for your own environment. The company IT team decided to leave the specifics of how to implement the processes to its outsourcing vendors — yet provide clear expectations about the end goals for each area. So if a particular process requires 15 steps in ITIL, the company might only care about four or five. Leaving the behind-the-scenes minutiae up to the vendor partners keeps costs down and simplifies life for the automotive company's partners.

For instance, in the area of incident management, IT dictates only how to classify an incident, whom to notify and when, the type of reports, and alerts. This allows the IT team to focus on the most important aspects of ITIL that relate to business needs.

Today, the company's IT department and the outsourcing partners are all on the same page when it comes to understanding where, when, and why business services have been compromised. This is due in part to the frequent communications that were initially established as part of ITIL. "It's amazing, the mindset change," the IT director observes. "Teamwork has gone up tremendously among the suppliers and within our company. We all can share and learn from what's going on."

Hard results? The number of recurring incidents has declined. On a scale of 1 to 5, end-user satisfaction is cited at 4.69 in 2005 — up from 4.46 in 2003. Also, availability has increased from 99.96 percent in 2003 to 99.98 percent uptime across 375 applications tracked globally. "We know intuitively and anecdotally that it's made a huge difference," the IT director notes, while admitting that will need to be quantified over time.

As with any overhaul to the IT organization, the IT director advises making it "evolutionary not revolutionary." Otherwise, you could face a revolt from your workforce and/or suppliers.

The company has kept the program somewhat under wraps outside of its IT staff. Senior management knows very little if anything about the new operational focus and processes. The IT director wants to keep it that way. "From a customer perspective, it's all about the quality of service that we deliver. I don't really want [ITIL] visible." In other words, the business doesn't need to know how IT gets to the end goal; IT just needs to keep the business satisfied with the results.

Fortunately, the ITIL methodology fits in well with the automotive manufacturing culture, which centers on process and control. Standardized processes also mean better efficiency and coordination among the outsourcing partners, which will cut costs for the company's IT department. "ITIL has been the rallying flag to dramatically increase our operations focus and our operations accountability," the IT director concludes.

TIPS FOR MAKING ITIL WORK FOR YOU

The IT director of enterprise operations management for a large automotive company's IT department offers a few pieces of cogent advice on how to bring ITIL to play in your IT organization:

- > Have a good baseline understanding of your operations today, so you can show the progress later with statistics and detailed analysis.
- > Engage your suppliers and/or your workforce. They need to be part of the process to both adapt and adopt ITIL.
- > Invest in education. Senior executives with 20 to 30 years experience, in particular, will benefit from learning about the ITIL terminology, which can be very different from typical IT language.
- > Don't go out and buy software tools before you have your processes hammered out. If you have to start by using Microsoft Excel spreadsheets and Microsoft Word documents to record information, do it.
- > Don't try to adopt ITIL verbatim. Instead, choose the concepts and processes that make sense for your business and IT environment.

should make people's jobs easier, not harder. Use best practices you've identified to help process owners see and feel how their process will improve with the CMDB. Remember the "what's in it for me?" perspective, and ensure you appease it.

Third, feel free to mix and match what makes sense from these different frameworks. You can select IT control requirements from COBIT, and combine and implement with some best practices from ITIL. If your organization will be certified to a particular standard, then following any CMDB-related practices in that standard should be a top consideration. For example, ISO/IEC 20000 may be a goal for obtaining industry recognition that your services are at a known level of maturity and quality. Using ITIL will help you to achieve ISO/IEC 20000. ISO/IEC 20000 tells you what to do to manage IT, and ITIL tells you how to do it.

You want to ensure that the best practice framework you select is proven and has enough buy-in that it will continue to exist in the future. Implementation is easier when training classes are available and suppliers have the knowledge, skills, and technology for the selected best practice framework. Best practices available outside of the mainstream may be applicable to your organization and may complement the mainstream best practices; just be aware that fewer applications may support them and the longevity may be limited.

You will not need all of the best practices that you've discovered to successfully implement your CMDB. It should be obvious which ones you should utilize fully, which ones you will use parts of, and which ones you should use as a reference. If you find best practices that are relevant, but not a good fit for the first stages of implementation, note them and set them aside for possible future use, in particular for your continual service improvement program.

Task 3. Document Best Practices

Now that you have identified the best practices applicable to your organization and reviewed them, you should document your findings and recommendations. Remember that you may recommend implementing elements of several of the best practices to create your own solution. This is acceptable. However, obtaining application suites that automate and integrate these elements may be difficult, or even impossible. You might need to make compromises to avoid high costs in time, budgets, and integrations.

Task 4. Check Current Levels of Adherence

You have discovered, reviewed, and documented each key service or process owner's choice of CMDB-related best practice frameworks that support the specific goals of your project. Now you need to present your findings to the PEB.

The PEB needs to understand the current state of the IT operation so that future process changes can be scoped as part of the CMDB implementation. If a key process requires a major adjustment to use the CMDB, the project team may want to note this as a significant risk that should be managed.

The PEB may want to conduct a review or audit with key service or process owners to create a baseline. Sometimes an organization can use ITIL as the baseline and then determine which elements it has implemented and to what level of alignment, maturity, and integration (if any).

Stage 2. Define Requirements and Create IT Service Model Blueprint

An organization might engage analysts or consultants to check against maturity models, such as the Gartner Infrastructure Utility Model. Or an organization might engage IT auditors to review against COBIT. Whatever method you choose, at the end you will have an understanding of the current state of operations and the gaps that need to be addressed.

Task 5. Publish and Distribute Best Practices

Once you have developed a plan based on the current maturity review, and have received PEB approval, publish the plan and make it available to the entire CMDB team. Prepare a brief presentation, so the team can get an understanding quickly. This presentation, together with the full documentation, can then be made available for reference throughout the CMDB implementation project.

Summary

The use of the CMDB is the whole purpose of this project. Why reinvent the wheel when figuring out what will change after the CMDB is in place? Time spent identifying CMDB-related best practices accelerates the time to benefit that justifies the project. Work with key service or process owners to review and align with those best practices that are best suited to your organization. Document the selected practices, and ensure it is clear what is being implemented. Key process owners can now begin planning specific changes to their work procedures, and the CMDB team can start documenting specific process integration requirements for the CMDB.

Step 8

Identify Requirements to Address Potential Problems

Goals and Objectives

In this step, you will identify and think through potential problems that might occur when building and maintaining a CMDB. You will identify the root cause of those problems and specify requirements that should prevent the problems from occurring.

Overview

Like every project team, you want your CMDB project to run as smoothly as possible and be an overwhelming success. In many project implementations, the team reacts to problems as they arise — confirming the saying, “Experience is something that you get just after you needed it.”

Fortunately for you, this adage is not as true in a CMDB project, because other organizations have already encountered most of the problems that might occur when building a CMDB. The key is to identify as many of these potential problems as possible and then take steps to either avoid them or reduce their impact.

Problems will fall into two broad categories: the problems that can occur when building a CMDB, and the problems that can occur after the CMDB is implemented. You will need to identify and address both.

Project team members should always keep in mind the idea of reducing problems and the impact of problems, particularly when attending educational events, conferences, and other networking opportunities. The project team members should pay special attention whenever they hear a problem with a CMDB implementation being discussed, and then report that problem to the other project team members.

Remember: A smooth implementation includes not just building a CMDB but also ensuring that once implemented, the CMDB is effective and easy to manage.

Task-by-Task Guide

Figure 8.1 shows the tasks for identifying and addressing potential problems in a CMDB project.

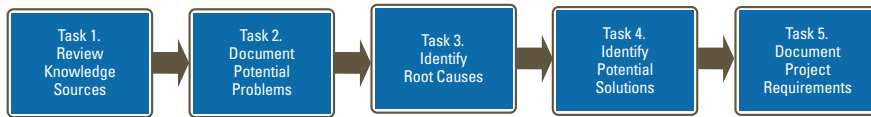


Figure 8.1. Tasks for identifying and addressing potential problems in a CMDB project

Task 1. Review Knowledge Sources

In step 2, “Obtain CMDB Knowledge,” you identified the information the project team requires to build a successful CMDB. Your work in step 2 will help you pinpoint information sources to help identify and reduce problems.

You’ve already defined your CMDB goals and expected benefits. Now you need to go back through relevant CMDB information you found to look for potential issues and lessons learned. You can find much useful information in books and other publications. Review all potential knowledge sources and evaluate them for useful data.

In addition to those sources, use your network of peers and build on their successes and lessons learned. Networking is a rich source of information, because first-hand experience is priceless — especially when the person providing the information can also suggest a solution or workaround.

Task 2. Document Potential Problems

In step 2 you were encouraged to record in a central repository any potential problems that could affect the CMDB. Now, review the potential problems you have already identified, and look for additional potential problems that may arise.

Here are some common problems that may occur:

- » **Lack of management commitment** — No project can succeed without management commitment and drive. You can achieve isolated wins implementing a CMDB without management commitment, but these wins will be few and far between. Commitment itself is not enough; senior managers must show their commitment to the CMDB project by their presence, involvement, and sponsorship.
- » **Spending too much time on complicated process diagrams** — When you start to approach a CMDB project, you might be tempted to produce complex and detailed process maps. This project is about building a database, not process engineering, so do not get pulled into creating complicated process diagrams.
- » **Not creating work instructions** — Too often, work instructions are not written to describe supporting, managing, and maintaining the final CMDB. It is very important to have clearly stated work instructions that are in writing, published, and continually reviewed. Service management process models can assist customers by providing work instructions out of the box.

- » **Not assigning data owners** — Many IT organizations are silo-based and not process-oriented. The CMDB will use data from different sources, which may cause confusion, especially if the data owners are not clearly defined. For example, when reorganizing the CMDB, which data can be removed? Without data ownership, who will decide?
- » **Not focusing on the right balance between performance and quality** — Most IT monitoring activities concentrate on performance, while ignoring quality and processes. A CMDB with high performance and poor content is useless. Likewise, great content and very poor performance is also a hindrance. Focus on balancing performance and quality.
- » **Being too ambitious** — Building a CMDB is a major undertaking that requires a great deal of planning and skill. Many organizations attempt to include too many goals for their CMDB in the initial deployment. Keep the project achievable, and look to attain early wins that show progress and quickly deliver value.
- » **Not focusing on the right level of granularity** — A major obstacle to building a CMDB is not focusing on the appropriate level of granularity of the data. A high level of granularity may result in a CMDB that is highly used, but is difficult and expensive to maintain. A low level of granularity will be easy to maintain, but very few people will use it. This is another balancing act.
- » **Allowing organizational demarcation** — Multiple organizations in addition to IT will use the CMDB and benefit from it. This often causes conflict, especially in cases where organization boundaries are rigid and ownership is important. All organizations need to understand that the CMDB is a joint venture and that success comes from everyone working together, not from ownership of a process.

MAINTAINING ACCURATE DATA IN THE CMDB

Information management consultant Tim Mason addresses the challenge of ensuring your CMDB data is consistently correct.

Clear accountability for data goes hand in hand with the process element. We found that everyone wanted information from the CMDB, but no one wanted to pay to maintain it. People pointed to the configuration management team and said, “It’s their responsibility.” Configuration managers, however, cannot own the data because they have no way of knowing if the data is accurate. They can own the processes related to checking accuracy, but not the data itself.

With this in mind, we built a model of data ownership with clear measures that put the onus of maintaining CMDB data on the people who know the data best. This involved implementing a network of data owners and data managers. The data managers are members of the support group for each CI. We aligned their roles to the data they need to support on a CI — for example, operating system support was aligned to operating system information. Data managers are accountable to a data owner for data quality.

We incorporated data quality targets into contracts for third parties and established personal objectives for internal staff. The data owner acts as an escalation point to which configuration management can raise issues. We modeled all this in the CMDB. Now, when users click on a field, they can immediately see who owns the data.

Excerpt from “6 Tips for Effectively Managing the ‘People’ Factor,” VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Stage 2. Define Requirements and Create IT Service Model Blueprint

- » **Ignoring best practices frameworks other than ITIL; not considering corporate control requirements** — Although ITIL is regarded as the industry best practice for IT service management, many other best practices and frameworks exist, such as COBIT, Six Sigma, and CMM. Also keep in mind corporate control requirements, such as Sarbanes-Oxley. Ignoring these other components can delay implementation. Even worse, if you don't focus on these components, then you are not maximizing the CMDB's potential.
- » **Not preparing the CMDB management team** — Once you have completed the CMDB project, the fun really begins. This is a challenging database to maintain; it includes data from many sources and potentially high levels of federation. Make sure you prepare your CMDB management team so that your CMDB maintains reliability and accuracy.

Use the preceding list as a starting point. Identify all of the potential problems and record them in a central repository.

Task 3. Identify Root Causes

For each potential problem you have identified, you should specify the root cause of the potential problem in your organization. First, a warning: Do not let this task become an obsession. If you have discovered a large number of potential problems, you will need to prioritize them based on risk and impact. You may need to involve people who are not members of the project team — database experts, for example.

After you have prioritized your problems, you can begin to identify the root causes. As you identify each root cause, you can continue to the next task. It is not necessary to wait until all of the root causes have been identified.

Task 4. Identify Potential Solutions

Specify a solution that addresses the root cause of each problem. Remember to be realistic. You will not be able to eliminate all of the potential problems. In some cases, you may need to create a workaround to bypass the problem when and if it ever occurs.

When identifying your solutions or workarounds, you may find that some problems cannot be eliminated or that workarounds cannot be created. Reducing risks is sometimes the only action that you can take. For example, you can minimize the problem of identifying the right level of granularity by planning and you can reduce the risk by appointing official data owners. Unfortunately, not everyone sticks to their commitments, so you also must conduct regular audit checks of the CMDB to reduce the risk. Take as many precautions as possible to reduce the impact should a known problem occur. Every project will encounter its own set of problems. Whatever risk-reduction actions your project requires, make sure that you use a change management process and fully document the actions.

Keep in mind that there may be more than one way to solve a potential problem. Therefore, identify all possible solutions or workarounds before progressing to the next task.

Task 5. Document Project Requirements

Now document the project requirements that will help you achieve the solutions you have identified for the problems. You can avoid some of the problems, such as “not preparing the CMDB management team” through effective project management. Other problems might require a process solution or a technical solution that you should call out in the CMDB requirements. As part of documenting the requirements, you might want to identify the project stage where you would actually implement the workaround or fix.

Summary

This step is important, because the quality of the end product relies on astute observation to identify and reduce the impact of problems. The key here is to learn from those who have gone before you. Care must be taken, however, not to allow this step to take too much time. Keep the risk factor in mind at all times to manage this step. And be specific in defining requirements that address the root cause of the potential problems as they apply to your project, goals, and unique organization.

Step 9

Identify Inventory and Asset Requirements

Goals and Objectives

In this step, you will identify all of the IT asset management-related requirements for your CMDB, if improved IT asset management is one of your CMDB project goals. You will work with the IT asset manager to define the current scope of asset management and what it covers, such as contracts and finance. You also will specify CMDB requirements that will meet the needs of expanded asset management that leverages the CMDB.

Overview

Your CMDB will contain information about a broad range of IT infrastructure components. According to ITIL, a configuration item (CI) can be defined as any component that you need to manage. IT inventory management, IT asset management, and configuration management functions all use a common subset of attribute data for each CI. However, each function has different additional requirements for attribute and relationship data that you should sort out before you complete the detailed CI and data model requirements in steps 12, 13, and 14.

Inventory and asset management applications manage the same CIs with a different purpose than configuration management. Each of these functions most likely has a purpose-built application and supporting database.

The inventory layer of CI attribute data is usually a list of items that describes the unique physical properties and location of a CI — for example, model number, serial number, location, owner. A copy of Microsoft Office in use on a desktop is a CI that can be tracked as inventory. The concept of *physical* might seem confusing when describing software, but the concept can still apply if you think of *physical* as an entity. This unique physical attribute data is used to track the quantity and location of CIs. An inventory management application that uses this data tracks ownership, location, condition, lifecycle, and stock levels. It might keep track of the number of copies of Microsoft Office in use and compare to the number of licenses purchased.

Stage 2. Define Requirements and Create IT Service Model Blueprint

The asset management layer of CI attribute data focuses on financial and governance information. The office suite in the earlier example has monetary value that can be added to a CI. Asset management can provide the answers to questions such as :

- » How much did the asset cost?
- » What is its value at any time?
- » How much does it cost to maintain?
- » What are the terms of the contract?
- » What is the level of depreciation?
- » When should it be replaced?
- » What governance controls are required?

This data is vital information for the financial management aspect of managing the IT infrastructure. Those responsible for IT financial management can use this data to understand the costs of services that IT provides to its clients.

IT typically focuses on IT assets, but remember that IT assets are just one type of capital assets. For example, a large production assembly line requires far more assets than just the IT assets. Similarly, a retail logistics system requires at least trucks, stores, shelving, and storage in addition to its IT assets. These physical assets also contribute to other organizational assets, such as human, information, financial, and intangible assets. Asset management applications may manage these other types of assets as well as IT assets.

Configuration management requires an additional layer of data about a CI, including current patches and specific functional settings, and includes relationship data. An entry in the CMDB is a CI, which can contain inventory and asset data. Establishing relationships between these CIs creates a powerful tool. For example, if a server is a CI and all of the workstations are also CIs that have data describing their relationships, then you can quickly see which workstations are attached to which server. As a result, you can prioritize more accurately, resolve incidents faster, and authorize change requests more reliably. CIs turn a list into a management tool. (Upcoming steps will discuss CIs and their contents in more detail.)

While there is usually significant overlap in what is tracked as a CI and what is tracked as an asset, the distinction is often a point of contention between operational support groups and IT asset management and finance groups. Figure 9.1 details the differences.

Inventory applications, IT asset management applications, and configuration management applications can all leverage a single CMDB. For the implementation to be successful, however, the requirements for these functions need to be defined up front and the CMDB must be designed to accommodate the requirements of these functions.

| Asset Management | Configuration Management |
|---|---|
| Goals: Manage asset costs, contracts, and usage/ownership throughout lifecycles | Goals: Provide logical model of IT environment as basis for ITIL processes |
| Value: Lower asset TCO/acquisition costs, reduced purchasing, more efficient allocation, more accurate budgeting/planning | Value: Greater business service stability, availability, quality (via related ITIL processes) |
| Asset: Physical IT component tracked based on value, contractual compliance | CI: Physical or logical IT component managed for its operational impact |
| Type: A CI can be an asset if it is worth tracking for cost, contract, and usage | Type: An asset can be a CI if it is worth managing for operational stability |
| Differences: Assets not likely to be managed as CIs include monitors, printers, servers in inventory | Differences: CIs not likely to be managed as assets include a custom Java component, a process, a service model |
| Relationships: Basic relationships (peer, parent, child) between assets are maintained for retirement process, ownership, license matching | Relationships: Sophisticated relationships between CIs are maintained to assess change risk, analyze root cause, assess service impact |

Figure 9.1. Differences between asset management and configuration management

Source: "Facilitating Effective Asset Management with the CMDB: A Nine-Step Approach," *VIEWPOINT, Focus on: CMDB Leadership*, published by BMC Software.

The CMDB can be a significant improvement even for existing IT asset management programs. Many organizations gain this perspective from painful experience. When Y2K required organizations to check every IT asset and component to ensure that it was Y2K compliant, many IT organizations did not have an accurate list of assets and their attributes. Huge amounts of money were spent to inventory those assets. Most organizations missed this golden opportunity to build an accurate asset database or CMDB. Now legislation such as Sarbanes-Oxley is again making IT review its assets, but this time the asset database must remain accurate. If an asset database is a requirement, then it makes sense to tie into other activities that need the same information.

In this step, you will identify and define the inventory- and IT asset-related information to be stored in your CMDB. Some of this data might already reside in other databases; if so, you'll need to decide whether to replicate that data into the CMDB or federate that data — that is, link to the original data and access it on behalf of the CMDB when it is required. You'll also need to determine whether any new data should be included in the CMDB. You will use the output of this step when you create the CIs and populate the CMDB in later steps.

Task-by-Task Guide

Figure 9.2 shows the tasks for identifying assets and inventory components.

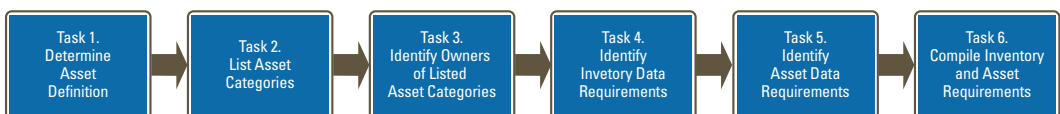


Figure 9.2. Tasks for identifying assets and inventory components

Task 1. Determine Asset Definition

The first task is to work with the key stakeholder who is responsible for IT asset management in your organization to determine the established definition of an IT asset. Some organizations expense IT capital purchases that fall below a certain dollar value or that have a short expected life, so they don't have to track depreciation. By knowing how your organization defines assets, you can better identify the CIs that will have additional asset-related attribute data in your CMDB.

If your organization already has a documented definition, then use that. If not, then consult with your senior IT managers, IT auditors, and IT accountants to obtain a definition. Don't write your own definition unless a senior manager requests you to do so. You want to ensure that you are following the organizational rules, not creating business definitions.

Task 2. List Asset Categories

Next you must identify the categories of CIs that are considered assets according to your organization's definition. Find out what types of assets are currently being tracked in the asset management application. Also find out what the key stakeholders would want to track if they had an accurate CMDB. The list might be large, but stick to your organization's definition of an asset.

IT asset categories might include:

- | | |
|------------------------|------------------------|
| » Workstation hardware | » Application software |
| » Workstation software | » Printers |
| » Local servers | » Laptops |
| » Remote servers | » Controllers |
| » Operating software | » Processors |
| » Telecommunications | » Scanners |
| » Business software | |

The point is to identify CI types that will require additional asset-related data to be associated with them. You are not populating the CMDB. You are ensuring that when you do fill it with CI data, you have identified the CIs that require additional asset-related data collection. After you have created your list, discuss the contents with the same people that provided the definition, just to double-check that you have not misunderstood or missed any categories.

Task 3. Identify Owners of Listed Asset Categories

All asset categories should have owners; that is, someone must be responsible for the lifecycle of the assets. Not only will ownership improve the management and content of the CMDB, but it also will help you to meet current, and potentially future, governance requirements.

As you identify the owner of each asset category, you might discover new asset categories and amend some categories that were identified in the previous task. This is good, because the better the list, the higher the quality of your CMDB. However, keep in mind that identifying the asset category owners may be more difficult than it sounds, and some asset categories might not have owners.

CONFIGURATION ITEM OR ASSET?

You can tell whether something is a configuration item (CI) by determining whether you have to manage it in order to deliver an IT service. If it requires management, then it is a CI. Another way to identify whether an item is a CI is to use the USMC test:

- > Is it unique? (U)
- > Is it required to deliver a service? (S)
- > Can you manage it? (M)
- > Does it have at least some characteristics that can change? (C)

If the answers to all questions are yes, then the item can be considered a CI.

ITIL considers all resources related to IT to be assets. For the purposes of this chapter, however, the discussion will be limited to financial assets. To determine whether to track an item as a financial asset, look for affirmative answers to questions such as the following:

- > Is this asset being depreciated as a fixed asset?
- > Does the asset have any annual maintenance agreements?
- > Do I need to track costs for this item over its complete lifecycle?
- > Is this asset covered under a software license agreement and does it need to be tracked for compliance?
- > Could we negotiate lower costs if we knew how many of this type of asset we would purchase next year?

If the CMDB will also be used for IT asset management, the IT asset management group will have a much broader definition of either what goes in the CMDB or what information needs to be accessible via federated data stores. In addition to configuration data, IT asset management would also want to record asset data such as contract, location, and other information that enable the cost tracking and compliance controls.

Task 4. Identify Inventory Data Requirements

Now get specific about what type of information you need for inventory purposes. You might have a bar-code system that associates an externally identifiable number with a manufacturer's serial number. You might specify location, owner, purchase date, or other information that is relevant to inventory managers.

Keep in mind that each field of data you collect and maintain comes at a cost. If the data must be updated when it changes, that costs money. If the data is not updated and becomes stale, the reliability and adoption of the CMDB as a whole comes into question.

Task 5. Identify Asset Data Requirements

Now get specific about what type of asset-related information you need for asset management functions. You might track purchase price, stage of asset lifecycle, or other information that is needed for the asset management application.

Also identify asset-related data that is specific to certain types or classes of assets. Be careful to keep the discussion focused on the asset categories to be included in the CMDB.

Task 6. Compile Inventory and Asset Requirements

Compile the final list of inventory and asset data that needs to be added to each CI and incorporate the requirements into your master requirements document. Distribute the list to senior IT and business managers. The data to be included for each of these IT assets will be discussed in step 14, “Define Configuration Item Attributes.” At this point, you are creating the list of IT assets whose data will be contained or federated into the CMDB.

Summary

In ITIL V3, the definition of asset is expanded to include anything that is a resource or a capability, including such things as the organization, processes, knowledge, people, and information. Yet, not every asset will be placed in the CMDB. In ITIL V3, asset management focuses on financial assets.

Whether you include inventory and asset attributes in your CMDB involves many considerations: You may or may not already have an IT asset database or numerous IT asset databases, a clear IT asset definition, and asset category owners. Perhaps your organization never had the need to manage assets, so nothing exists at the moment. Governance requirements also will influence your decisions. You must consider these factors and decide to what degree assets will be represented in your CMDB.

If in doubt, then include, or federate, assets into your CMDB. Even if you do not include the financial attributes for each asset, you at least can be confident that your CMDB encompasses CIs that are considered assets. It is easier to build a CMDB that includes assets than to retrofit assets into a CMDB later.

The less effectively that assets are now managed in your organization, the bigger the benefit of adding a CMDB. The accurate data in the CMDB can help drive initial savings from removing duplicate software, identifying software that is no longer used but is still being paid for, eliminating redundant hardware, and reducing the number of suppliers you must manage.

Step 10

Define Service Catalog Requirements

Goals and Objectives

In this step, you will identify requirements that specify how the service catalog will leverage the CMDB. When you understand which CIs relate to a particular service, you can better meet SLAs and can more effectively calculate service-based costing. To make the connection between infrastructure or business services and underlying CIs, your CMDB should include relationship data that references services defined in your service catalog.

Overview

The CMDB can significantly improve the management and delivery of an organization's service offerings. If a goal for your CMDB project is to enable an improved service, then the CMDB requirements must include service details and relationships. Including the service details and relationships in your CMDB will enable you to analyze data on a service basis to perform important service management activities. Examples of these activities include:

- » Timely, accurate service impact assessments for proposed changes
- » Appropriate incident response and escalation from the perspective of the service offering
- » Relevant resource focus and attention to fixing the problems that present the greatest urgency and impact to the business
- » Intuitive and effective service level monitoring and reporting of both customer service commitments (service level agreements [SLAs]) and related internal or external supplier partner agreements (operating level agreements [OLAs] and underpinning contracts [UCs])
- » Improved risk analysis, which will reduce potential critical failures and problems — especially when assessing the impact of changes

You can accomplish these activities only if you have multiple views through your configuration structures (service configuration structures and infrastructure configuration structures) that link your services to your underlying infrastructure. The CMDB is a great repository to hold the break

Stage 2. Define Requirements and Create IT Service Model Blueprint

down of the service offerings and show the links to the underlying infrastructure as well as related internal or external supplier chain. The practice of Business Service Management (BSM) creates a bridge between service assets and business services provided at a higher level.

Equipped with this knowledge, you can identify the type of relevant management information you will require to meet the service commitments you have made across the service types. Use this information to determine appropriate CMDB structuring, levels, relationships, and attributes.

Your organization may offer different types of services that will demand very different measurement and management characteristics. For example, your organization may provide services that are categorized as business service offerings. Ready for direct consumption, these services might include an end-to-end service, such as a desktop that comes complete with network connectivity, office productivity software, and access to approved business solutions. Alternately, your organization may offer services to integrators who combine your IT services with other IT services (an application service, data center service, infrastructure service, or network service, for example) to create a business service offering.

The differences between these two types of service offerings and the consumer expectations related to them can create very different information needs in your CMDB. Figure 10.1 shows a generic model of how service and infrastructure configurations interrelate and the hierarchy that might exist between and within the configuration structures. Step 12, “Define Configuration Item Level and IT Service Model,” discusses this hierarchy in greater detail.

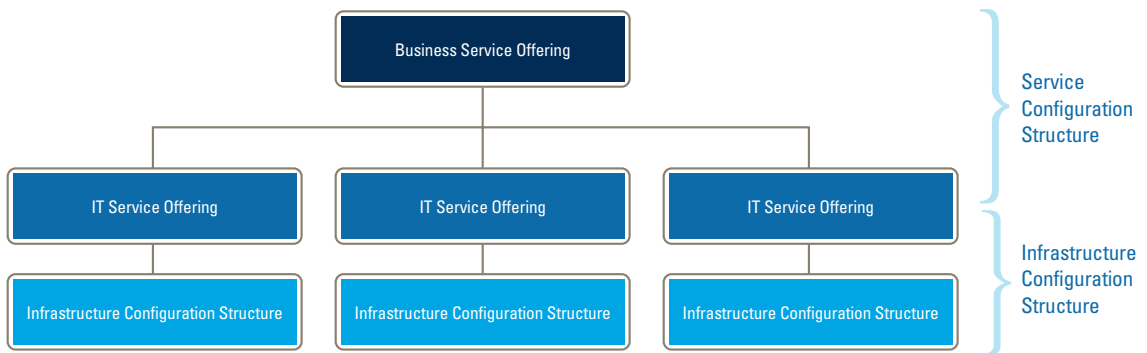


Figure 10.1 Interrelationship between the service configuration structure and the infrastructure configuration structure

Business customers increasingly demand service levels and service management from an end-to-end perspective. They are becoming much less patient with service definitions and agreements that focus on technology and infrastructure components. Instead, customers insist on service agreements that reflect their business services.

The closer the service is to the end user, the more end-to-end encompassing the service commitments will need to be, and the more integrated the management of all the service components will need to be. Several components and provider partners might make up the service your

customer is measuring you against. The information needed to manage the service commitments and the supplier chain can vary greatly between infrastructure services (such as data center services) and end-user-consumable business services (such as an e-mail service or a business solution).

By capturing the services in your service catalog as service CIs in your CMDB, understanding the hierarchy of those services, and identifying the key information requirements to effectively manage the service commitments, you can ensure you have a window into your services that underpins successful delivery management. Keep in mind that linking service assets to business services is a key element of BSM.

Task-by-Task Guide

Figure 10.2 shows the tasks for defining CMDB requirements that relate to the service catalog.

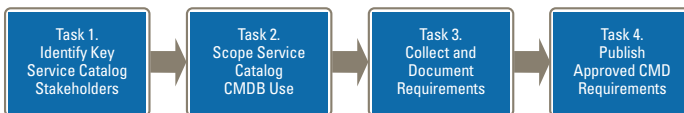


Figure 10.2. The tasks for defining CMDB requirements that relate to the service catalog

Task 1. Identify Key Service Catalog Stakeholders

Work with the key stakeholders you identified when you created your project goals to refine impact estimates to create specific requirements.

You should work with the people who have ownership of the service catalog. These individuals should be very knowledgeable about existing and planned services. They also should be able to identify the service monitoring and reporting information needed to improve customer service level and support management. Stakeholders could include planning managers, service planners, portfolio managers, program or service owners, customer account representatives, and service level managers.

Task 2. Scope Service Catalog CMDB Use

You will need to determine how the IT service catalog will use CMDB data to improve its function. The service catalog may be a simple table, matrix, or spreadsheet, or it may be a much more detailed document describing each service, its characteristics, and its service level commitments.

The general approach for using the CMDB to support a service catalog is to create service CIs in the CMDB that correspond to services in your service catalog. Service CIs can be divided into subservices, and can be linked to underlying infrastructure CIs, creating a service hierarchy.

Consider this example. Your catalog includes a specific business service. To be available for use by the customer, the business service depends on a network service, an application service, and so forth. By defining this business service as a CI in the CMDB and relating it to the network service and the application service, you now have the ability to identify the impact of incidents and problems that are recorded against the network CI or the application CI to the business service — not just the technology service. This ability may also help you prioritize problem management activities.

WHAT IS A SERVICE?

Brady Orand, *IT service management practice manager at Column Technologies, Inc.*, discusses how to begin defining business services.

When I help companies to create service definitions, first I gather all the IT stakeholders together and ask them what services they provide to the business. They usually start saying things like, “We do change management” or “We do patch management.”

I turn the conversation into a business discussion, and ask them to define the services that the business consumes. For example, a business does not buy change management or incident management. A business buys the ability to provide outstanding customer service or the ability to efficiently fulfill orders or to keep its production line operating smoothly.

Defining services is an exercise in getting different minds to meet. IT needs to realize that the business doesn’t care about the details of patch management, change management, intrusion detection, or virus detection. Even though these are vital activities performed by IT, the business is not really concerned with them. Rather, the business is focused on completing its critical business processing. But if those critical IT activities weren’t performed, then the business would most definitely care — because their critical business processes would not be operating effectively.

Excerpt from “Start with Service Definitions and Reap Success,” VIEWPOINT, Focus on CMDB Leadership, published by BMC Software.

Further, you have a view into your technology, allowing you to consider the potential risk and impact of a change request to an infrastructure component on your customer’s business service. This integrated information, combined with an integrated tool, forms the basis for timely, effective business and/or IT service monitoring.

To begin this task, use your service catalog and related agreements (SLAs, OLAs, and UCs) as a primary source for determining requirements. At the end of this task, you will be able to answer these questions:

1. **What types of services does my organization offer?**
A good understanding of the types of services your organization offers will help you establish CI structuring and relationships.
2. **Who are the customers?**
Knowing the customers for the services you provide will enable you to define CI relationships or attributes, or both.
3. **What other services and infrastructure do the service offerings rely on?**
Understanding the services and infrastructure that the service offerings rely on will help you determine CI structuring and relationships.
4. **Who are the suppliers of these other services and infrastructure?**
The answer to this question will assist you in establishing CI relationships or attributes, or both.

5. What service commitments have been made?

Understanding service commitments enables you to define CI attributes.

6. What is the critical management and reporting information required for effective delivery management?

Use the answer to this question to help you define CI attributes.

First, identify the types of service offerings you provide. These types might include business services that are ready to be consumed, and IT services that provide service elements that must be combined with other service elements to form a business service. Select at least one service from each type of offering. Break down the service to the underlying infrastructure level, identifying the answers to questions 1–4 (see above). You must break down and identify the required relationships for all your service offerings prior to populating your CMDB. For now, it is only necessary to repeat this step until you understand the types of relationship information needed to manage the service. This information will be used to determine the CMDB service structure and level design, and it will help you determine important types of relationships. Step 13, “Define Configuration Item Relationships,” discusses relationship types in more detail.

The last activity in this task is to understand the commitments that you have made in your service level agreements with your customers. Continue to analyze the services that you’ve broken down, focusing now on understanding and documenting answers for questions 5 and 6. This analysis will enable you to determine the types of attributes that you will require in the CMDB. Remember, many CMDB designs identify adequate attribute information to manage the technology and the infrastructure; however, the CMDB designs often overlook important service management attributes. Sample service management attributes could include:

- » Service level targets and priority
- » Service entitlement information (hours of service, required approvals, etc.)
- » Service-related notification, communication information, etc.
- » Service owners (by their job titles, for example)
- » Service maintainers
- » Service managers (by their job titles, for example)

Keep in mind that the service level measures and management information may also vary according to the types of service offerings you identified earlier. For example, figure 10.3 shows an IT service called Network WAN Service. The customer of the Network WAN Service is looking for measures related to the performance and cost of that service (latency, circuit availability, throughput, circuit cost, etc.). This IT service has been bundled with other IT services (Application Service and Data Center Service) to create a business service called Point of Sale Service. The customer of this business service would look for measures that focus on end-to-end solution performance and cost. The measures may also include solution performance, transaction processing, and solution availability across all the combined services. In both cases, the CMDB will need to house the attributes and relationships of the combined services and components to enable both perspectives.

Stage 2. Define Requirements and Create IT Service Model Blueprint

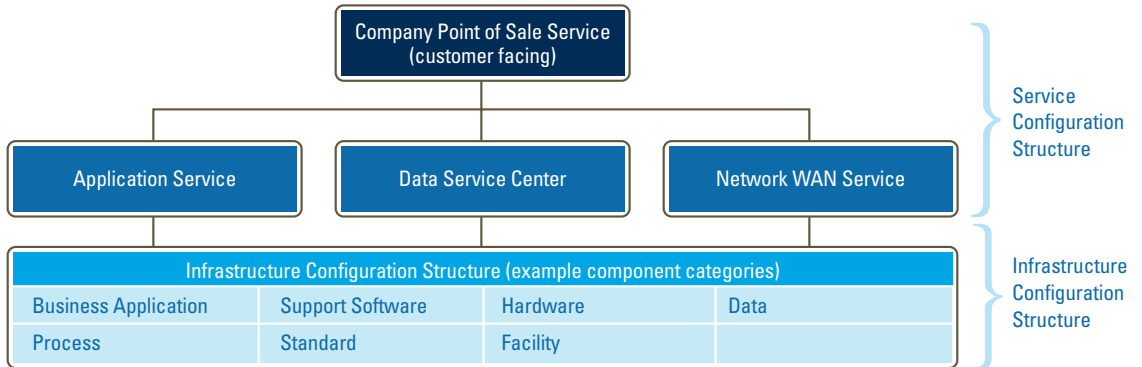


Figure 10.3. Types of services that help determine required attribute information

Figure 10.4 provides sample measures for different types of service offerings.

| Service Type | Example Measurement Attributes and Measures |
|--|--|
| Business Service Offering (e.g., Point of Sale Service) | <p>End-to-end availability of the Point of Sale Service Service is available 24x7, 99% of the time, except Saturdays from noon to 3:00 p.m. for preventive maintenance calculated quarterly</p> <p>Responsiveness of the Point of Sale Service Maximum time to complete a sale is 15 seconds, 99% of the time</p> <p>Cost of Point of Sale Service described in business transaction terms (price per sale) Each sale costs \$0.10</p> <p>Target Point of Sale Service restoration time and % If an outage occurs, the service will be restored in 2 hours, 98% of the time, calculated quarterly</p> <p>Target Point of Sale Service response time and % If a high-priority (high-impact, high-urgency) outage occurs, the customer will be notified in 15 minutes, and every 4 hours thereafter</p> |
| IT Service Offering (e.g., Network WAN Service) | <p>Availability of the WAN Service WAN Service is available 24x7, 99.95% of the time, except for preventive maintenance from noon to 1:00 p.m. on Saturdays, calculated monthly</p> <p>Responsiveness of the WAN Service (latency/wait time) WAN Service latency is <12ms, 95% of the time, calculated daily</p> <p>WAN Service circuit cost per month WAN Service circuit cost is \$1,200 per month for 3 Mbps symmetrical throughput (no limit of bandwidth)</p> <p>Target WAN Service circuit restoration time and % If the WAN Service experiences an outage, service will be restored in 1 hour, 99% of the time, calculated monthly</p> <p>Target WAN Service response time and % If a high-priority (high-impact, high-urgency) outage occurs, the integrator will be notified in 5 minutes, and every 2 hours thereafter</p> |

Figure 10.4. Sample measures for different types of service offerings

By now, you will have answered the six questions listed in this task and have a good basis to articulate your service catalog-driven requirements.

Task 3. Collect and Document Requirements

Next, distribute your list of CMDB requirements to the key stakeholders you identified earlier. Review all comments and suggestions, and revise the CMDB requirements document to ensure that current and future requirements, agreed upon by the key stakeholders, are reflected in the final document.

Task 4. Publish Approved CMDB Requirements

Finally, incorporate the CMDB requirements that relate to your service catalog into your master requirements document. Also, provide the requirements to the CMDB design team for consideration in the CMDB information model.

Summary

The CMDB can provide a powerful service view of the technologies and products it offers. The CMDB is a very effective tool, allowing you to view information from both the service and infrastructure perspective. If you take the time to understand the types of services you offer, the related service hierarchy, and critical service level management information requirements (both today and emerging), you can ensure that your CMDB supports effective delivery of business services.

Step 11

Define CMDB Requirements to Support Other Processes

Goals and Objectives

In step 10, “Define Service Catalog Requirements,” you defined the CMDB requirements for the processes that support the elements of your service catalog. The goal of this step is to ensure that your final CMDB will contain the data and information the other IT service management processes require so they can be as efficient as possible.

Overview

The CMDB can provide views of data to give you the information you need to make wise IT business decisions and efficiently perform IT service management processes. To achieve these results, you’ll need to determine which processes your CMDB will support. Refer to the goals you developed in step 3, “Create and Agree on CMDB Goals and Mission Statement.” Also look at the expected benefits you defined for your CMDB project in step 4, “Review and Define Benefits.” Even if you are not implementing ITIL best practices, you can use the ITIL processes as a basis for defining the service management processes you will support. You should also consider non-ITIL processes. Together, the goals, benefits, and processes will help you define your CMDB requirements.

ITIL Processes

ITIL provides a set of best practices for IT service management and related processes. The underlying enabler to most (if not all) of the ITIL processes is the CMDB. Figure 11.1, “Sample Configuration Management System,” illustrates the interaction between IT processes and the CMDB.

A CMDB enables you to create a service dependency map so that you can relate events, incidents, problems, and requests for change to service impact. Refer to step 4, “Review and Define Benefits,” for a detailed description of the value that the CMDB brings to the various ITIL processes.

Stage 2. Define Requirements and Create IT Service Model Blueprint

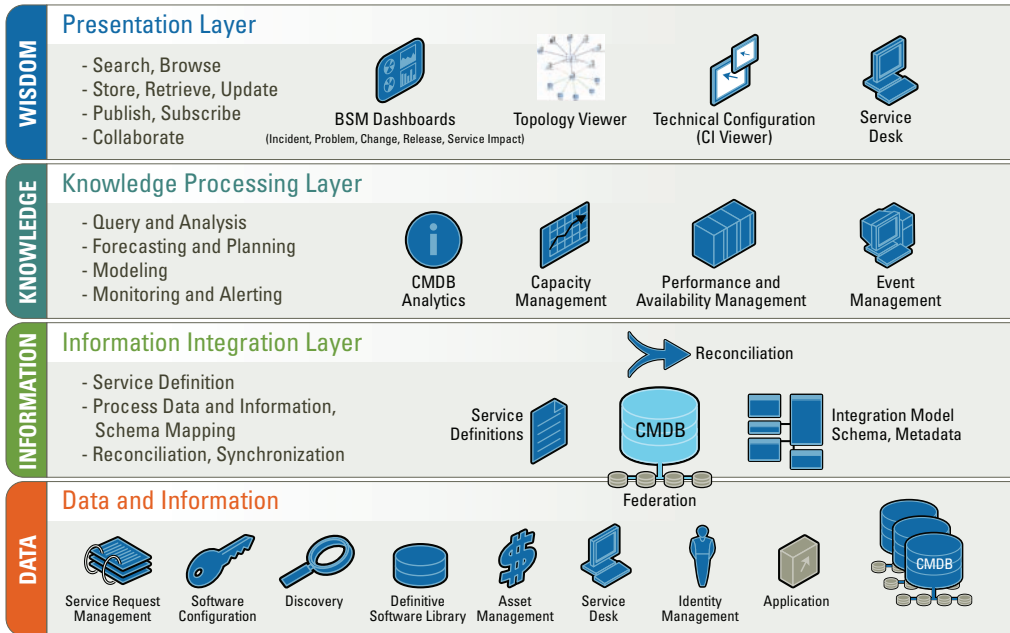


Figure 11.1 Sample Configuration Management System

Non-ITIL Processes

In addition to the ITIL processes, organizations perform many other IT processes that rely on accurate IT service and component data. Refer to step 4, “Review and Define Benefits,” for a detailed discussion of the value the CMDB brings to some potentially important non-ITIL processes.

You should understand how these processes might need IT service and component data from the CMDB. You also should know which processes are most critical and therefore worth your effort to collect and maintain the data, including the correct links and relationships.

Task-by-Task Guide

Figure 11.2 shows the tasks for defining CMDB requirements to support other processes.

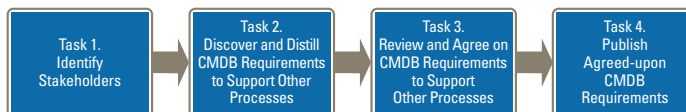


Figure 11.2. Tasks for defining CMDB requirements to support other processes

Task 1. Identify Stakeholders

You need to involve the right stakeholders so you can ensure accurate identification of all the process requirements and can build organizational buy-in. You will need stakeholders representing each of the areas included in the goals you defined in step 3, “Create and Agree on CMDB Goals and Mission Statement,” as well as each area you specified as benefiting from the CMDB in step 4, “Review and Define Benefits.” When identifying your stakeholders, consider these factors:

- » **Seniority** — Are these individuals senior enough to have the authority to speak on behalf of their areas? This may not be a factor in all organizations.
- » **Content knowledge** — Can these individuals actively participate and speak to today’s needs and gaps as well as identify emerging requirements?
- » **Appropriate representation** — Is the representation in the group broad enough to help you distill the data and identify the processes to consider now and in the future?

At a minimum, the stakeholder group should include process owners or managers and those essential to the individual processes. The group also should include those responsible for overall service quality.

This stakeholder group must understand the mission, goals, expected benefits, scope, and overall purpose of the CMDB that you defined in stage 1. The group will help to ensure you have appropriately considered the CMDB needs of all key processes — ITIL and non-ITIL — that you expect to support with the CMDB. Furthermore, this group will help you obtain organizational acceptance of the CMDB road map.

Task 2. Discover and Distill CMDB Requirements to Support Other Processes

The stakeholder group will help shape the overall CMDB design by providing CMDB requirements from their perspectives. You will need to inventory (or at least understand and document) the key processes in your organization that will rely on the CMDB, as determined in step 3, “Create and Agree on CMDB Goals and Mission Statement,” and each area you specified as benefiting from the CMDB in step 4, “Review and Define Benefits.” This is a good time to review and confirm that these are, indeed, the key processes.

THE IMPORTANCE OF IDENTIFYING STAKEHOLDERS

Don’t stop with simply *identifying* stakeholders. Rather, engage them in the process of defining requirements. All too often, the CMDB project team has already decided what they want to build and the idea of “identifying and engaging stakeholders” is merely an exercise in figuring out who should receive copies of the design document. Don’t make this mistake. Meet with your stakeholders on a regular basis; ask them what they need/want — and really listen. Always remember that, ultimately, the CMDB is a repository of information to be leveraged by the IT service management processes and the process owners. It exists for their benefit. Actively involve those process owners and other stakeholders in the requirements definition activity.

Stage 2. Define Requirements and Create IT Service Model Blueprint

Then assess the processes you identified in steps 3 and 4, one at a time, to identify their CMDB needs. You can accomplish this assessment in a number of ways. You might simply document a list of abilities or needs for each process. Some examples include:

- » The incident management process:
 - The CMDB needs to enable access to important support information (support team information, CI end-to-end service mapping, escalation information) for quick and accurate problem resolution.
 - The CMDB needs to allow access to mean-time-to-restore targets and business impact information so IT can properly assign priority and apply appropriate escalations.
- » The change management process:
 - The CMDB needs to show the CIs related to a request for change so IT can perform an accurate change impact assessment.
 - The CMDB needs to enable the ability to understand the potential impact of a change on each service.
- » The performance and quality process:
 - The CMDB needs to provide a view of performance information by individual service, service portfolio, and customer group.

You should also have a general understanding of your current process maturity levels and whether your organization has any goals or plans to increase maturity. Some sources of insight might be service improvement programs, ITIL maturity assessments, process audit reports, and so forth. Understanding where your organization is focusing its process improvement efforts will help you prioritize the processes that will influence CMDB requirements.

You might conduct a prioritization exercise to create a sequenced list of requirements and a timeline that may influence the CMDB road map. The CMDB project could address the high-priority requirements now and consider the others in the future.

Task 3. Review and Agree on CMDB Requirements to Support Other Processes

When you have defined your CMDB requirements to support other processes, distribute the list to the key stakeholders identified earlier. If you included some of your key stakeholders in the requirements definition exercise during task 2, you may need to circulate this report only to the rest of the group.

Consider any feedback from the stakeholders and incorporate it into the final requirements document.

Task 4. Publish Agreed-upon CMDB Requirements

Now, distribute the final agreed-upon requirements to the key stakeholders, and incorporate these requirements into your master requirements document. These requirements, along with the requirements defined in step 9, “Identify Inventory and Asset Requirements,” and step 10, “Define Service Catalog Requirements,” will serve as the basis for the IT service model design as well as the definition of relationships and attributes.

Summary

Do not underestimate the importance of this step. You need to have a clear picture of the information the CMDB must provide to your IT business processes. Spending a little time to draw out these requirements now will ensure your CMDB is designed for your business needs today and tomorrow. If your organization is seeking to improve its process maturity, a well-designed enabling CMDB is half your battle. The CMDB is the intersection point for all your processes. Without it, you will have difficulty achieving process integration and you will limit the maturity of your processes.

Step 12

Define Configuration Item Level and IT Service Model

Goals and Objectives

In this step, you will define and document the optimum level for CIs in your CMDB. Levels refer to the depth of CIs you choose to include in your CMDB, whereas the structure refers to both the depth (level) and breadth (overall scope).

Configuration structures should describe the relationship and position of CIs in each structure. Be sure to identify both the infrastructure configuration structures (which include application components) and service configuration structures (which identify all the components in a particular service).

A CI can be a single component or a complete system. For example, a CI could be a workstation or each workstation component — such as the keyboard, processor, mouse, and screen. You must specify the level of CI for each CI category, or type, as it relates to the CMDB scope. If your plan for populating your CMDB is to add workstation hardware, workstation software, local servers, remote servers, operating software, telecommunications, business software, etc., until you achieve the overall scope of your CMDB, then you need to provide the CI levels for each of these categories.

Overview

Gone are the days of isolated technology solutions that have dedicated infrastructures. The demands for IT organizations to deliver services faster, to reduce costs, and to reuse and share infrastructure components present myriad challenges.

You need to support IT services at the required levels while efficiently introducing changes that combine critical infrastructure components into new service offerings. You must understand the potential service impacts of proposed changes. And, you need to assign the right priority and escalation to incidents and problems. To meet these requirements, you need an IT service model that provides a decision-support view of infrastructure and service configurations,

Stage 2. Define Requirements and Create IT Service Model Blueprint

including the key relationships that underpin your services. Step 10, “Define Service Catalog Requirements,” outlined how defining and understanding your service offerings provide the basis for creating an IT service model. Now you should extend that knowledge of the service configuration structure into the infrastructure — the infrastructure configuration structure — that underpins those services to enable effective end-to-end service management.

Previous steps described the ability of the CMDB to support governance, asset and inventory, and service catalog requirements. The CMDB must also support the operational needs of integrated ITIL and non-ITIL processes. You need to consolidate these requirements into a unified and prioritized service model that specifically addresses *what* your organization needs to get out of the CMDB to support your IT business services. Figure 12.1 shows how requirements determine CI level, relationships, and attributes, and how an IT service model blueprint ties all of these factors together.

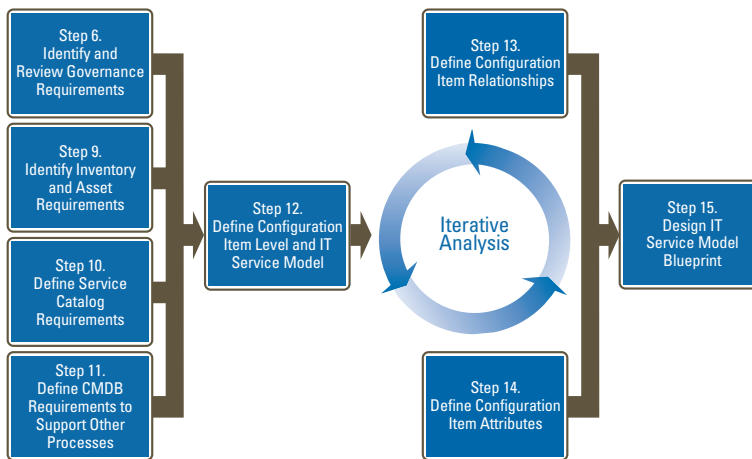


Figure 12.1. Creating the IT service model blueprint

Determining the appropriate CI level, attributes, and relationships is an iterative process. You need a strong understanding of these three dimensions of CMDB development before you finalize your IT service model. Step 15, “Design IT Service Model Blueprint,” provides additional guidelines to help you determine when you should address requirements through level, attributes, or relationships. To minimize unnecessary iterations, familiarize yourself with this step as well as the attribute and relationship steps that follow before you begin your IT service model design in step 15.

The CI level is important to the structural design of your CMDB. Defining too many CI levels will consume significant effort with little reward. Too few CI levels can leave you without the information you need to effectively control, analyze, and manage IT services. Even worse, you could face serious legal ramifications if you fail to manage information adequately. Consider an activity such

as creating your personal budget. Too many categories and subcategories (and sub-subcategories, etc.) of expense types can create an overly complicated budget, and you may never use some categories. This same logic applies to configuration management information models. The desire to *capture it all* can cause you to store information that provides little or no business value and to incur unnecessary costs. Understanding *how much* to capture is equally as important as deciding *what* to capture, and is one of the biggest challenges in your CMDB design.

You'll need to answer questions such as: Should a workstation be a CI, or should there be a CI for each workstation component, such as one for the screen, one for the processor, one for the keyboard, one for the mouse, etc.? Should the productivity software suite be one CI or should there be one CI for each component, such as a word processor, spreadsheet, etc.? Should a patch to the productivity software suite be a CI? Typical criteria to consider include:

- » **Cost or value** — Is the component sufficiently expensive or valuable that you need to track it individually? An expensive graphics card could be a CI even if it resides in a workstation.
- » **Change considerations** — Do you frequently make changes to the component, especially location changes? Does the CI level support adequate control, recording, and reporting to the level of "independent change?"
- » **Traceability** — Do you need to trace the component, perhaps for audit reasons (change-related audit trails and accountability over material CIs, for example)?
- » **Governance and compliance requirements** — Do you need to demonstrate Sarbanes-Oxley compliance? Do you need one source of information about all infrastructure elements so you have a checkpoint for contract and usage compliance?
- » **Management of service commitments** — Is the component a critical part of a service offering that you would need to consider when analyzing the impact of changes on service commitments?
- » **Maintainability** — How will CI levels affect your database and information maintenance abilities? For example, how do you update thousands of PC CIs to show that a new software release has been installed on each one?
- » **Delivery cost and quality** — What level of CI is sufficient so you can record incidents and changes to support effective problem management activities?
- » **Interrogation capability** — Would the inclusion of the CI or categories of CIs allow the required search or analysis capability within the infrastructure configuration?

After you have answered these questions, you can define the boundaries of your service model. Later steps will define attributes and relationships that, coupled with your level design, provide the basis for your organization's IT service model blueprint.

CAN PEOPLE BE CIs?

Troy McAlpin, CEO of AlarmPoint, makes the case for including people as CIs in your CMDB

Can people be CIs too? The answer is an overwhelming “yes.” People can — and should — be CIs because they represent an organization’s human assets. Therefore, the CMDB should include data about the relationship of an event to the business customers who rely on a service, as well as the relationship of an event to the personnel required to resolve it in the incident management process. This relationship information relies upon data about People CIs.

People CIs are an obvious extension to the data you already include in the CMDB. Personnel, after all, are some of the most expensive resources in IT. Information about people can improve your organization’s ability to resolve incidents and provide services to the business.

What information about people should you maintain in your CMDB? You’ll want to include their duties, skills, certifications, and interests, as well as the services they use and the services they support. Relevant contact information, such as phone or pager number, e-mail address, instant messaging handle, location, language, time zone, and schedule, should also be included. In other words, People CIs should include the following information about the people who consume and provide services: who they are, what they do, how you find them, and the events and services that are pertinent to them. People CIs have attributes that help you understand who the best person is to help resolve an incident. The attributes also can indicate that a person is an end user of IT, is associated with a particular service, and wants to know about an impact or a potential service level agreement (SLA) violation, for example.

Sometimes, a person will play different roles depending on the event. Someone may be a direct impact person for one type of event because of a dependence on a particular asset or service. But, for another type of event with a different asset, that person may be responsible for resolving incidents. For example, if a network component fails, the network support or operations team is primarily responsible. However, the network security group may also need to be informed about the failure. You have to understand the person’s relationship to the service and the different components that make up the service, as well as the role the person will play.

Because a person’s functions can change often, all of this information can get complicated. For example, travel plans or schedule changes affect someone’s availability. If someone gains additional technical certifications, that person can now assist on different types of events. Maintaining People CIs in your CMDB can help keep all the information straight.

Excerpt from “Can People Be Configuration Items?” VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Task-by-Task Guide

Figure 12.2 shows the tasks for defining CI levels.

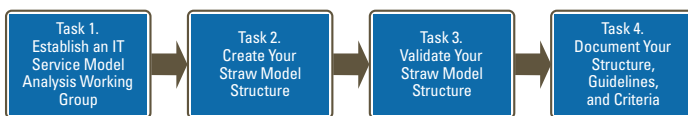


Figure 12.2. Defining CI levels

Task 1. Establish an IT Service Model Analysis Working Group

In designing your IT service model, you'll need a working group that includes people who have a focus on service management as well as infrastructure management. The stakeholders you identified in step 10, "Define Service Catalog Requirements," and step 11, "Define CMDB Requirements to Support Other Processes," can provide expertise for your service configuration structure. This structure will require relationships and attributes that help interrogate and analyze information from the perspective of your organization's business service and IT service offerings. Stakeholders who have technical and operational expertise can help you to define your infrastructure configuration structures. These stakeholders may include application development or support teams, IT infrastructure architecture and support groups, security and privacy practitioners, and process owners.

Representatives from the technical and operational perspectives will be critical to ensuring that your IT service model includes the appropriate linkages between service and infrastructure configuration structures. The IT service model analysis working group must use a combination of viewpoints from these two configuration structures to rationalize and reconcile the leveling, relationship, and attribute requirements simultaneously to meet business needs.

Task 2. Create Your Straw Model Structure

To successfully distill and rationalize your IT service model requirements, the working group needs to create and agree on a straw model structure (a working model that serves as a framework, and evolves over time) and levels that link and enable views of the service and infrastructure configuration. Your straw model structure and levels should include:

- » Definitions of configuration structure terms
- » Common CI categories and levels (including guidelines and criteria) for the infrastructure configuration structure

Establishing a straw model structure can help you to align and communicate the concept of an IT service model and consistently address requirements. Figure 12.3 illustrates a straw model structure.

Stage 2. Define Requirements and Create IT Service Model Blueprint

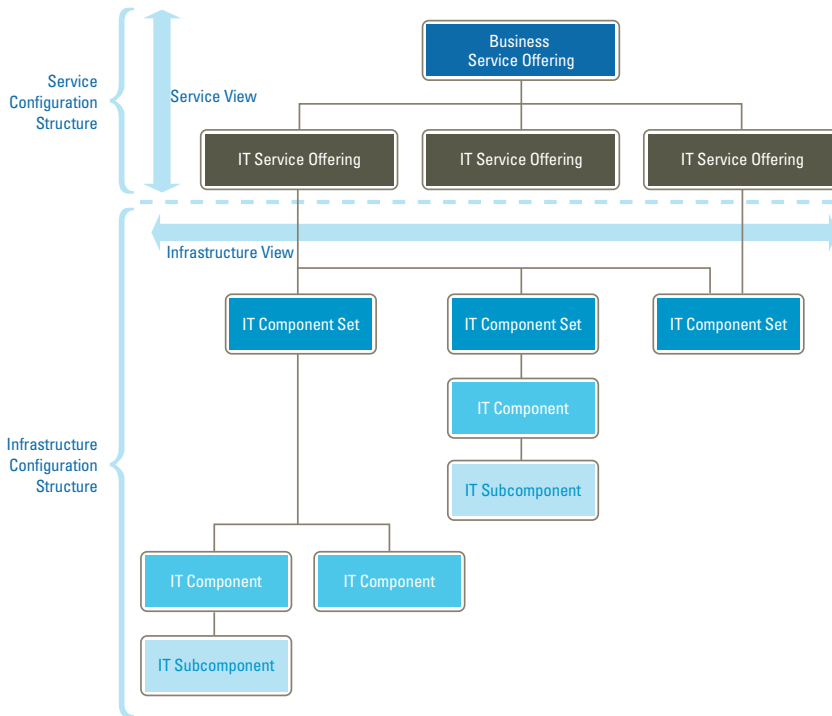


Figure 12.3. Example of a straw model structure

Figure 12.4 shows a linkage between the service configuration and infrastructure configuration structures (initially presented in step 10, “Define Service Catalog Requirements”). Defining services as CIs allows your CMDB to relate individual technology assets (IT components) to end-to-end services (service offerings). The infrastructure configuration structure is where most organizations begin (and often end) when defining and developing their CMDB structure.

Step 12. Define Configuration Item Level and IT Service Model

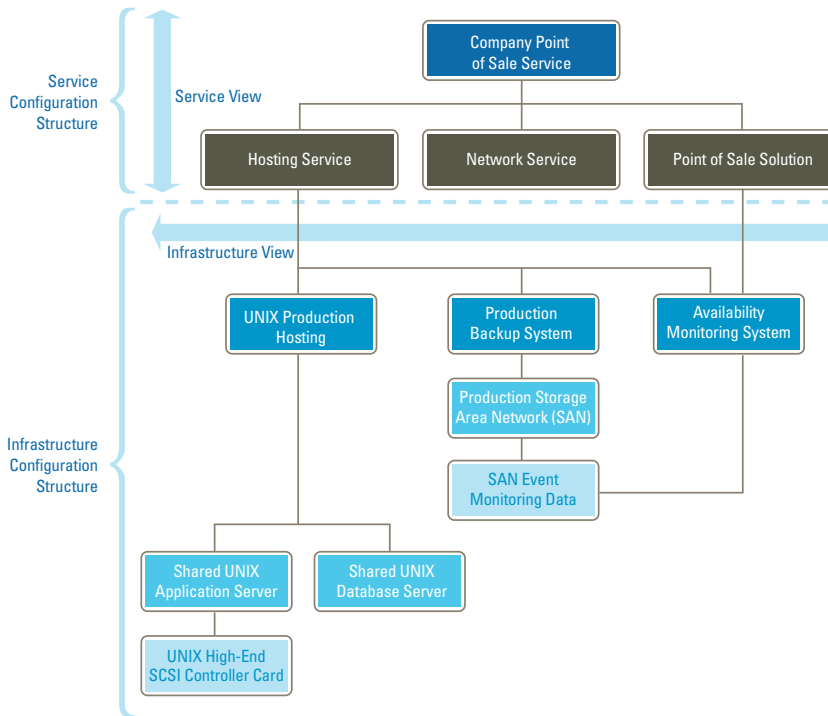


Figure 12.4. Linkage between the service configuration and infrastructure configuration

Defining configuration structure terms. Figure 12.3 also introduces some terms that define different levels of an IT service (discussed in step 10, "Define Service Catalog Requirements"). The generic term *service* is often used to describe both business service offerings and IT service offerings. However, without structure, communicating what is being delivered becomes difficult, if not impossible. Categorizing offerings and components is necessary so you can consistently represent and enable analysis and interrogation from an end-to-end perspective and by unique infrastructure views.

The names of the levels are less important than their meanings. You can select different terminology to express what is relevant to your organization. Figure 12.5 lists infrastructure configuration structure terms and includes the service configuration structure terms that were introduced in step 10, "Define Service Catalog Requirements."

Stage 2. Define Requirements and Create IT Service Model Blueprint

| Level | Offering |
|-------------------------------------|---|
| Service Configuration Levels | <p>Business Service Offering – A business service is an IT service that supports a business process, or a service that is delivered to business customers. Business service offerings typically are bound by service level agreements.</p> <p>IT Service Offering – An IT service supports the customer's business processes, and is comprised of people, processes, and technology. IT service offerings typically are bound by service level agreements.</p> |
| Infrastructure Configuration Levels | <p>IT Component Set – An IT service offering consists of one or more sets of IT components. An IT component set allows you to establish groupings or bundles of IT components with groupings of other IT components (component sets) to define new service offerings and simplify the management of relationships. Outsourcing of service functions typically occurs at this level, and outsourced sets of IT components and their performance objectives are often bound by underpinning contracts. Component sets provide the linkage between infrastructure and service CIs. Grouping component sets creates a valued service offering.</p> <p>IT Components and Subcomponents – Component sets are bundles of IT components, or traditional CIs. IT components can have subcomponents, and those components can have subcomponents, etc. This parent-child relationship reflects the CI level used to store detail, and the relationship is structured by categorizing CIs. The best practice of managing changes at the level of the independent change often determines the CI level.</p> |

Figure 12.5. Configuration structure terms

Defining categories. IT components should fall into predefined, standardized categories, each one containing *like-CIs*. In the infrastructure configuration structure, categories encompass similar CIs that are then further detailed through parent-child relationships, where children are specializations of their parent. Categories can help you understand how much detail is required in each group of similar CIs. For example, consider the category of hardware. The next level down could contain generic categories of hardware, such as servers, workstations, and routers. The further down this hierarchy, the more specialized the components become. (See figure 12.6.)

Categories allow for ease of interrogation and analysis of the CMDB as well as simplified attribute management. Consider the following scenario. An emergency patch to a server operating system is being proposed to respond to an identified vulnerability. As the manager of servers, you want to query the CMDB to identify all the servers that will be affected, so you can understand the size and scope of the change. If you have categories in your CMDB, the query needs to look only within the server category. A service management representative will want to understand what services may be affected by the proposed change, so that service management can adequately inform its customers. The correlation between the infrastructure (those particular servers) and the service structure (service offerings) allows this information to be accessed and analyzed quickly and effectively.

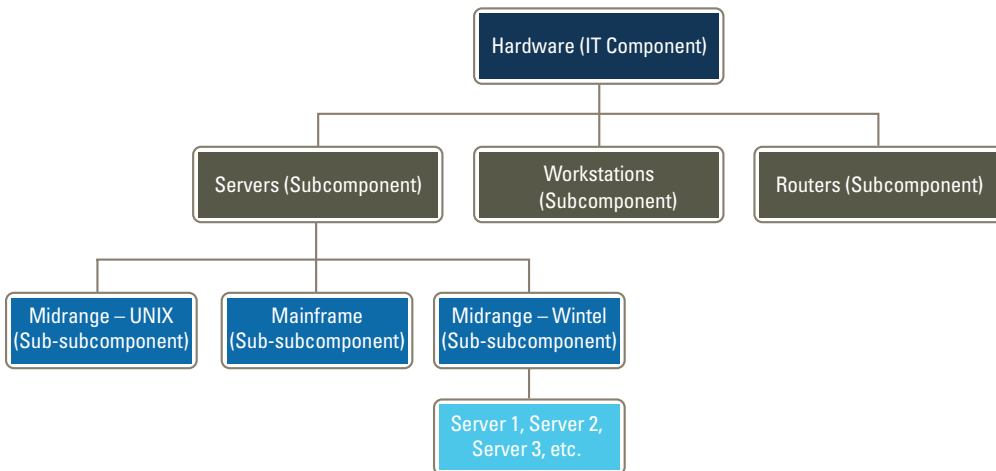


Figure 12.6. The category of hardware and the IT subcomponents

When you manage your CIs, remember that different categories have different lifecycles and inherent attributes. Most CMDB implementations recognize types of categories in their structure, allowing you to group CIs accordingly. See step 14, “Define Configuration Item Attributes,” for further discussion.

Use the following guidelines to begin defining your IT component categories:

- » Use categories to refine searches and maximize query performance through optimized filtering. IT components may be *physical* or *logical*. *Physical* components are assets that have a specific location, take up space, and can be “seen.” Examples include servers, documents, operating system instances, desktops, etc. *Logical* components are assets that may or may not have a fixed location. They might not take up physical space and can be conceptualized, but not “seen.” *Logical* components often complement physical components. Examples include processes and standards, business and IT services, Internet, corporate intranet, software releases, etc. Some stakeholders may be interested only in the physical CIs and therefore would benefit from categories that represent only physical assets, for example.
- » The more top-level categories of components your CMDB has, the more challenging it may be to search and find CIs. Limiting the number of categories at the highest level of your CMDB will simplify the search criteria.
- » Use autodiscovery tools to populate your CMDB with physical component CIs. With well-defined component categories, you can simplify the autodiscovery linkages. Most likely, you will need to enter logical components manually. The good news is that logical components tend to change far less frequently than hardware components and their specific attributes.
- » Standardize the highest level of component categories to ensure that all components conform to a consistent taxonomy. Without consistency, the value of the IT service model diminishes. When correlating CIs becomes more complex, then the CIs lose information structure and meaning, making interrogation and analysis less effective.

Figure 12.7 provides an example starting point for defining IT component categories.

| IT Component Category | Category Criteria |
|--------------------------------------|--|
| Application/Business Software | Is the IT component a piece of software — either purchased, developed, or composed (reuse of other software components) — that provides business functionality of a given service element or system? <i>Examples: Developed or service-specific commercial off-the-shelf (COTS) application</i> |
| Support Software | Is the IT component a piece of software that indirectly supports a system's function, but does not provide the core functionality of a given system? <i>Examples: Operating system, backup software, antivirus software</i> |
| Hardware | Is the IT component a physical technological device that supports one or more functions within a system? <i>Examples: Server, workstation, router</i> |
| Data | Does the IT component represent a specific type of system, user, location, or organizational data that, when represented in a structured format, represents various types of information utilized by a system? <i>Examples: Clients, locations, application records, case files</i> |
| Network | Is the component a logical representation of a network component? While the devices for a network are defined as specialized hardware components, you should reference the interfaces on network components, such as ports, segments, virtual LANs (VLANs), etc. <i>Examples: Port 21, switch port 9/24, mission-critical DMZ, intranet zone, router T1 interface</i> |
| Process | Does the IT component represent a series of steps, procedures, or work instructions? Customer-facing service requests typically are represented by a unique procedure that must be followed to complete the request. <i>Examples: Add e-mail account, change domain password, back up system</i> |
| Standard | Is the IT component a set of enforced guidelines, policies, and/or restrictions for a particular aspect of IT? <i>Examples: E-mail retention policy, network access security protocol</i> |
| Document | Is the IT component a physical document, authored and owned, that relates to a specific system? <i>Examples: Build kit, service agreement, change management process guide</i> |
| Facility | Is the IT component a physical location that supports some form of IT operations or houses other IT components? <i>Examples: Remote office, data center, call center</i> |

Figure 12.7. Example starting point for defining IT component categories

Determining how much data to capture. Now that you have defined *what* information needs to be captured (breadth), you need to determine *how much* information should be captured (depth). What is the appropriate level of CIs within your infrastructure configuration structure? To answer that question, consider what level of detail you need to effectively control, record, and report CIs in order to meet business requirements.

Best practice suggests that changes should be *made* at the level *recorded* in the CMDB. For example, if an organization has decided to record software CIs down to the detailed level of each module (even though populating to this level will require more effort and cost), the change would be *made* and *recorded* at that level. However, if the software CI leveling extends only to the level of the program, then the change at the single module level would require the whole program to be recompiled and the change recorded at that level.

Use your requirements balanced against common sense to rationalize leveling decisions. Factors may include maintainability, availability, quality of information, and the possibility of more appropriately using attributes and relationships to capture the necessary data.

When you break down categories of IT components, you do not need to capture *everything* that represents a component. For example, in the hardware category, you could break each server into subcomponents (chassis, motherboard, disk array, power supply, etc.) and further subcomponents (memory, processors, SCSI controller card, etc.). However, when you consider each level of detail, ask yourself four fundamental questions before proceeding:

1. Is this level of CI necessary to meet service catalog or IT business process requirements?
2. What is the value in capturing this component at a more detailed CI level?
3. What is the overhead to manage and maintain this component compared with the value of doing so?
4. Is this level required to manage the control of changes related to this CI?

If the value of adding a subcomponent is comparatively less than the overhead to create and manage it, then don't add it. Instead, represent the additional detail as one or more *attributes* of the parent CI. Furthermore, you can choose to *relate* CIs rather than add attributes. Leveling, attributes, and relationships are interrelated and steps 13 and 14 will cover these concepts in more detail.

Routinely review your CI level to ensure that information is at the right level to be useful and accurate. At the beginning of your CMDB project, however, try to identify the lowest level of CI you require and create a CI level structure to enable its collection — even if you do not plan to collect this level right away. This approach will minimize CMDB redesign, which can be very costly and time consuming.

Task 3. Validate Your Straw Model Structure

The service catalog requirements affect the service configuration structure and will further inform attribute and relationship design. The IT business process requirements will strongly influence the infrastructure configuration structure levels.

For these reasons, the logical next step is to take the services you have already broken down to the underlying infrastructure level (step 10) and break them down further into the infrastructure layers of your newly defined straw model structure.

This break down gives you an opportunity to validate that your structure, categories, and levels meet your IT business process and service catalog requirements and can be practically applied. As you go through this exercise, you will further refine the guidelines and criteria that you have created.

Task 4. Document Your Structure, Guidelines, and Criteria

Now that you have validated the straw model structure (configuration structures, categories, and levels), you should ensure that you have documented the iterative decisions through this point of the design. These guidelines and criteria may require further refinement when coupled with CI relationships and attribute design (in step 15, “Design IT Service Model Blueprint”). The straw model structure and your design of attributes and relationships will form the IT service model blueprint for your organization.

Summary

One of the most important and challenging areas of CMDB design is choosing the right CI levels and related attributes and relationships. You must determine the right balance among the needed control, information availability, and the cost and effort required to maintain the CMDB data. You’ll achieve that right balance when you meet the IT business process and service catalog requirements for your organization. Too much information will create unnecessary cost and work for your resources and will possibly undermine the acceptance and use of the CMDB. Too little information will leave your organization with an inadequate level of control and an inability to optimize your services and performance. The CMDB sweet spot allows maximum control with minimum information. You need a comprehensive and consistently applied configuration structure. Otherwise, you won’t be able to provide the enterprisewide decision-support information necessary to run your IT business.

Step 13

Define Configuration Item Relationships

Goals and Objectives

In this step, you will define and document the relationships for all of the types of CIs that you defined in the previous steps. You will map the types of relationships that you define onto your service and infrastructure configuration structures. This map provides the important links and views of your services and underlying infrastructure. You will then review and agree on the draft relationship design. Finally, you will document the design, which will form part of your IT service model blueprint.

Overview

You have identified CMDB requirements to support your organization's service catalog and IT business processes, and you have started mapping these to an IT service model structure. In step 12, "Define Configuration Item Level and IT Service Model," you established levels within this model and linked a service configuration structure with an infrastructure configuration structure. This *linking* of structures is a critical step in providing an end-to-end view of the services that IT provides to the business.

The concept of *relating* CIs to one another, whether they are between or within service configuration structures or infrastructure configuration structures, is at the very heart of the CMDB. Without relationships between items in the CMDB, you have nothing more than a collection of asset identifiers. Asset management systems typically stop at this point, focusing on the information stored within database items versus the relationships between items. Without relationships, it is impossible to understand the configurations of assets or how combined assets deliver and support a valued end-to-end service.

ESTABLISHING CI FUNCTIONAL RELATIONSHIPS

David Chiu of BMO Financial Group, on determining interdependencies of CIs

A major issue that you'll need to address is controlling the number and complexity of functional relationships to be defined and managed. The situation is similar to the complex relationships among people. Some are familial or social, while others are health-related or service oriented. The list is long and varied. When people discuss relationships, however, they typically do so in a particular context that narrows the relationships to be considered. For example, in the context of patient admission to a hospital, only health-related and familial relationships are relevant.

Likewise, in the world of IT, the context determines the type of relationships that are of interest to you. Consider an example in which a server failure occurs. If you're the technician troubleshooting the problem, you may be interested in the physical and logical relationships of the server to other technology assets. If you're Service Desk personnel, on the other hand, you probably are interested in the relationships between the failed server and the business services it supports.

Categorization enables you to deal with the volume and complexity of relationships and present data and relationships in a straightforward, understandable manner. You only look at the relationship types that are of interest to you. To categorize relationships by type, you need an IT service structure that provides a formal method to define and name relationship types. For example, the term *runs on* may be established to define a type of relationship between an application and a server. Conversely, the term *run* defines the relationship between a server and an application.

Excerpt from "The IT Service Structure: Insight Gained through Implementing a CMDB," VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Relationships essentially provide an ability to understand dependencies between two CIs. Many dependencies have a direction. For example, if CI A *depends upon* CI B, then you know that CI A requires some output from CI B. Relationships in a CMDB are a two-way model. In the preceding example, you could say CI B *is a dependent of* CI A. (See figure 13.1.) This two-way interaction introduces a unique opportunity when you create your service model blueprint and enable it through technology. Be sure to consider the direction of relationships, because it provides additional and useful insight into your configuration data.

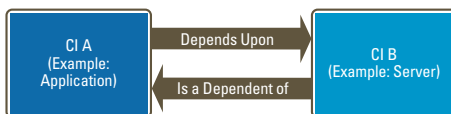


Figure 13.1. Two-way model of relationships in a CMDB

Creating a relationship between two CIs often occurs from a high-level perspective. However, if you manage one of the two CIs, you would want to know what CIs your CI is a *dependent of*, as well as the CIs that your CI *depends upon*. When assessing a change, for example, you would want to navigate from CI to CI using this two-way relationship model. Interrogating the CMDB using relationships will tell you what might be affected by a change, both directly and indirectly, as you branch out from the CI being modified. Another way to look at this is to consider a river. At any point in the river, events upstream might affect the flow of water at that point as well as points downstream. This view into your organization's IT structure provides real insight into your operational, tactical, and strategic capabilities and objectives.

One way to support this two-way model is to identify the inverse dependency for every relationship. The generic relationship type *depends upon* has an inverse relationship *a dependent of*. Figure 13.2 outlines additional examples of this mapping.

| Relationship Name | Inverse Relationship | Direction of Dependency | Example |
|---------------------------|---------------------------------|---|--|
| A depends upon B | B is a dependent of A | A depends upon B | <i>You can represent most relationships by this base relationship type. Try this with the other examples in this table.</i> |
| A is a copy of B | B is the original of A | A depends upon B <i>Since B is the original, any changes to B might affect the copy.</i> | <i>This disk image is a copy of that disk image. If the source image changes, then it is no longer a copy (unless it represents a point-in-time copy).</i> |
| A applies to B | B is an application of A | B depends upon A <i>B is affected by what A applies.</i> | <i>This physical security policy applies to that data-center access process. Changes to the security policy might affect the data-center access policy.</i> |
| A is used for B | B provides use for A | B depends upon A <i>B relies on the usage of A.</i> | <i>This Web camera is used for that identification card process. A non-functioning Web camera might affect the process.</i> |
| A pays for B | B is paid for by A | B depends upon A <i>B may need to be paid for, so a service or a component can remain active.</i> | <i>This cost center pays for that business application. The business application may require payments within a certain time frame to meet contractual obligations.</i> |
| A supports B | B is supported by A | B depends upon A <i>B may require support activities to occur to ensure availability targets are met or performance levels are maintained.</i> | <i>This server is supported by the mid-range server support group. The support group may perform support activities that are necessary for the server to maintain operational targets.</i> |
| A is a parent of B | B is a child of A | B depends upon A <i>B should represent some specialization of A.</i> | <i>A motherboard is a parent of a processor on the motherboard. The same processor cannot be a child of two motherboards at the same time.</i> |

Figure 13.2. Inverse dependency of relationships

Figure 13.2 identifies a special relationship, the parent-child relationship. The concept of this relationship type is that children represent some detailed aspect of the parent. Think of this as a *specialization* of a parent. A server (parent) has system components (children to the server but parents on their own), which then have subcomponents (children of the children), and so on. Your CMDB should contain this foundational relationship type to express levels within your infrastructure configuration structure. Using the parent-child relationship will help ensure that you capture your CMDB infrastructure CIs and represent them consistently. Parent-child relationships provide an opportunity for simplified attribute management.

Carefully consider the relationships you'll need in your CMDB, as relationships provide a unique opportunity for your organization. They allow you to group and associate CIs based on certain criteria, expressed by the name of the relationship you have defined. For example, you may need a view of all the CIs that *use* other CIs, or you may want to know where CIs are *copied* within

Stage 2. Define Requirements and Create IT Service Model Blueprint

your organization. As a general rule, do not define relationships unless you are addressing a specific view that meets a specific need. Err on the side of fewer relationships rather than more, because removing poorly defined or seldom-used relationships is more challenging than adding new relationships later as requirements become clearer. Finding information is more complex if you have too many relationships.

Task-by-Task Guide

Figure 13.3 shows the tasks for defining CI relationships.

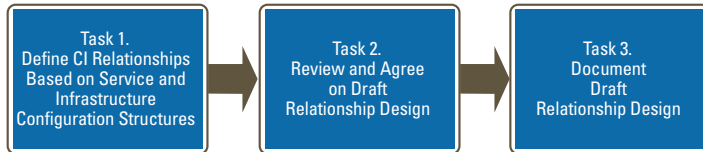


Figure 13.3. Tasks for defining CI relationships

Task 1. Define CI Relationships Based on Service and Infrastructure Configuration Structures

This section outlines many example relationship types that will enable your service and infrastructure configuration structures. The same analysis working group you established to review your CI level requirements should help define the relationship requirements. By the end of this section, the working group will be able to answer the following questions:

1. What relationships and guidelines are needed to support your service configuration structure needs?
2. What relationships and guidelines are needed to support your infrastructure configuration structure needs?

By reviewing and expanding or modifying the example structures that follow, you will be ready to address your specific organizational requirements for relationships.

Service configuration structure. Your working group will need to focus on those relationships that directly support the associations between your IT business service offerings and IT service offerings. Figure 13.4 shows an example relationship type that can be used for this association.

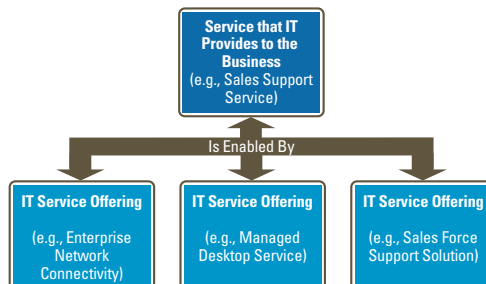


Figure 13.4. Example relationship to support the association between an IT business offering and infrastructure service offerings

The *is enabled by* relationship allows CIs that support the services that IT provides to the business to be associated with multiple IT service offering CIs that *enable* the end-to-end business service. Each IT service offering CI would have an inverse relationship, such as *enables*, representing the dependency.

Your IT service offerings require relationships as well. You should now identify the relationship that associates IT service offerings with IT component sets. Figure 13.5 shows IT service offerings associated with both IT component sets and other IT service offerings. You can accomplish this association through an *includes* relationship. The inverse of this relationship would be the *is included in* dependency. The ability for one IT service offering to *include* another IT service offering provides flexibility so that you can recombine an IT service offering numerous times, both for internal services and as a service to other integrators.

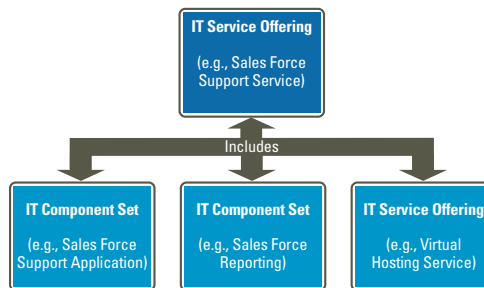


Figure 13.5. IT service offerings associated with IT component sets and other IT service offerings

Infrastructure configuration structure. Both internal and external providers to your organization need the ability to bundle IT components into sets. Sets allow greater flexibility for managing components that fulfill a well-defined function, and sets also simplify the number of relationships that require management. You could think of sets as a system, defined by Merriam-Webster as “a regularly interacting or interdependent group of items forming a unified whole.” Figure 13.6 represents a grouping of IT components.

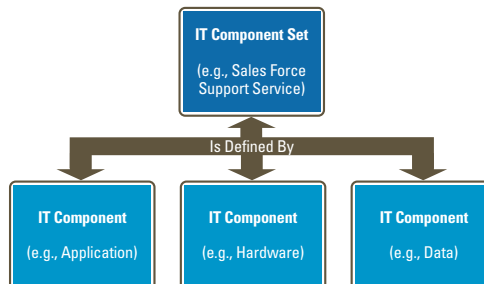


Figure 13.6. A grouping of IT components

An *is defined by* relationship associates IT components with a set. The inverse relationship could be *defines*. An IT component set needs only *defines* relationships to associate parent component CIs. Child CIs would be indirectly related and therefore would not need specific relationships on

Stage 2. Define Requirements and Create IT Service Model Blueprint

their own. This does not imply that you would relate the highest-level parents in your infrastructure configuration structure, which were expressed as category CIs in step 12, “Define Configuration Item Level and IT Service Model.” Instead, you would relate component CIs that represent those components and all of their children that should belong to the set. If some children don’t belong to your set, then you have chosen a component that is too generic.

Unlike the *includes* relationship defined earlier, the *is defined by* relationship would not normally relate IT component sets to other completed IT component sets. Relating IT component sets to other IT component sets adds a lot of complexity to your model. Even if you have two systems that share all the components except for one, you should create all the IT component relationships again. This approach makes performance measurement and monitoring easier, and it simplifies some measures. For example, if a cost measure for a component is normally shared between two component sets, the cost would be easier to manage through individual relationships to the components rather than by determining the percentage of a related component set’s cost. Be sure to consider any implications before expanding the scope of the *is defined by* relationship.

After you establish how components are related to sets of components, you must define how components themselves are related to one another. As discussed earlier, the concept of a parent-child relationship is very important. For discussion purposes, assume that each IT component falls into a category, as proposed in step 12, “Define Configuration Item Level and IT Service Model.” IT components within the same category have CIs that further break down into more specialized CIs, emphasizing the need for parent-child relationships.

Figure 13.7 is an example of IT components in the hardware category. Each child has a unique parent. However, a parent may have many children at each level. This type of relationship requires additional consideration, because CIs should not be children of more than one parent. If such a relationship is required for other reasons, then enable it through a relationship type other than a parent-child relationship.

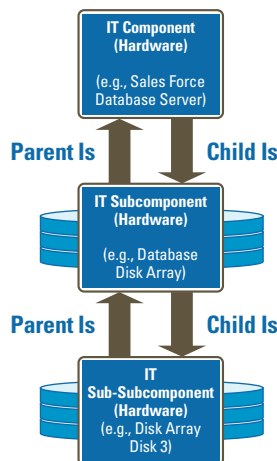


Figure 13.7. IT components in the hardware category

So far, this task has discussed how to group CIs in sets (link between service structure and underlying infrastructure structure) and how to further break down components within a set through parent-child relationships. Now you need to explore how to define other relationships within the infrastructure structure.

You may want to relate IT component CIs to one another, so you can understand how dependencies contribute to the successful management of your organization's IT assets. Your infrastructure configuration structure working group should focus on defining unique relationship types that enable this view.

One approach is to start with the generic *depends upon* relationship in your stakeholder sessions. This approach allows you to define a basic dependency map. It will also ensure that you identify relationships that support the interdependent view of your infrastructure, so that "nice to have" relationships are prioritized accordingly. When two CIs require different types of dependencies, consider creating a new relationship type to differentiate the relationships. For example, assume you have a server A that *depends upon* another server B. Then assume that server A also is synchronized with server B, where server B is the master, because they are configured to share a capacity load. You might need to reference this additional dependency separately from the generic *depends upon* relationship. Hence, you might need a separate *is synchronized with* relationship, for example.

Reporting requirements often drive relationships. Your working group should ensure adequate understanding and translation of your performance and reporting requirements, identified for IT business processes. These requirements may reveal a need for different relationship types to provide information where the simplistic *depends upon* relationship is insufficient.

Task 2. Review and Agree on Draft Relationship Design

Once your working group has explored your relationship types, then you must review and agree on the draft relationship design. The key to defining relationships, for either service or infrastructure configuration structures, is simplicity. The simpler the relationship model, the more likely it will be maintained and used. A CMDB that has 30 relationship types is more likely to have seldom-used relationships, and when they are used, the value of those relationships is questionable. Keep this in mind, and communicate to your stakeholder groups the need for simplicity.

If the working group used assumptions in its design of relationships, then the group may need to check with the broader service catalog and IT business process requirements stakeholders for further confirmation or clarification. If the group did not use assumptions, then you can delay the consultation with stakeholders until all considerations of structure, level, relationships, and attributes have been integrated into a proposed CMDB blueprint.

You also should capture any emerging tool-driven requirements noted during the design process that will help you select the CMDB solution. Future steps will discuss selecting and updating a CMDB tool to support the creation of your newly defined CMDB. Presumably, if your organization has already procured a CMDB, you will have involved the subject matter expert in some

Stage 2. Define Requirements and Create IT Service Model Blueprint

of the designing workshops. Although letting technology define business requirements is not a wise practice, sometimes technology options are limited (or forced, due to existing procurements and/or procurement limitations), and you should consider this when reviewing your requirements. Some CMDB solutions may force a particular set of relationship types.

Task 3. Document Draft Relationship Design

Now that you have created a list of draft relationships for your IT service model, ensure that these relationships are well documented and form part of the emerging blueprint for your IT service model. You can further refine these draft relationships when you couple them with the CI level and attribute requirements, which will be brought together and finalized as part of step 15, “Design IT Service Model Blueprint.”

Summary

Understanding relationships between CIs is, in essence, what differentiates a CMDB from an asset database. The ability to define relationships is critical to meet the CMDB requirements identified in earlier steps.

You might choose to implement only a generic relationship (*depends upon*) at first and then introduce more as you need them. The key is to have the right number of relationship types. Too many may become unmanageable and not easily understood by users. Too few may not give you the detailed view you need to interrogate your CMDB and analyze the information.

Also remember that you should not determine relationships in isolation. You must consider relationships along with attributes, infrastructure and service configuration structures, and levels when you create your IT service model blueprint.

Step 14

Define Configuration Item Attributes

Goals and Objectives

In this step, you will identify and document attributes for each type of CI, keeping in mind the importance of establishing the correct level of granularity. If you have too many attributes, you will have a large amount of information that could be difficult to maintain. If you don't have enough attributes, maintenance will be easier but use of the CMDB will be low because of lack of content.

To help you achieve the correct level of granularity in your CMDB, this step discusses the benefits of employing a federated CMDB that links core data from many data sources used by other relevant service management processes. A federated approach avoids duplication and management issues and allows functions such as viewing a service from another application.

Overview

Attributes are data elements that describe CIs, much like adjectives that describe nouns. Attributes help to identify and detail the important characteristics of what is in use, the status of the items, and their location. Samples of hardware CI attributes could include make, model, serial number, location, version, license number, and so forth.

In the previous steps, you established CI structure and levels and added relationships between those CIs, keeping in mind the need to limit the number of CIs and relationships to derive the greatest benefit with the least amount of complexity. Attribute definition follows a similar — yet easier — path of consideration. The idea of using your service catalog requirements and IT business process requirements continues in this step. Resist the temptation to capture all types of interesting and informative data related to the CI to store it in the CMDB; if this attribute data is not required and adds little or no value for managing your IT business offerings or IT service offerings, then this would be a costly and useless exercise. Also consider whether the information you need already exists in other systems and whether you need to replicate it into your CMDB.

Stage 2. Define Requirements and Create IT Service Model Blueprint

For example, think about the IT business process requirements to manage problems. You need to be able to analyze incidents against CIs to identify trends and conduct root-cause analysis. If your organization has an incident management system, then that system likely stores incident data. However, you will also need to be able to analyze the CI information from a change and release management perspective. Again, your organization may have separate systems to collect the discrete information. The constant in all the systems is that the information is related to a single CI (even if it is not consistently called the same thing). Figure 14.1 provides a sampling of systems and possible attribute information.

| Attribute Sources | Examples |
|--|--|
| Asset Management System | Depreciation amounts for hardware, desktop lease details, buy vs. build vs. compose decisions |
| Incident/Problem/Change Management Systems | Instances where a CI outage resulted in high-severity/priority incidents, root-cause diagnosis details, failed change impacts |
| Event Monitoring Systems | Trigger thresholds for bandwidth consumption, CPU performance trends, event correlation details |
| Financial System | Cost center specifics, chargeback formulas, profit margins, financial trends, fiscal targets, employee salaries |
| HR System | Employee payment schedules, position details, reporting relationships, skills, certifications, learning plans |
| Directory Systems (e.g., Microsoft Active directory or LDAP) | User details, location definitions, association of hardware location to user consumption, authentication criteria, passwords |
| Definitive Media Library | Physical license store, archive of all in-house developed code, license compliance reports, master copy of commercial off-the-shelf (COTS) software packages Physical repository of hardware components, replacement parts, hardware inventory records. |
| Document Management System | Physical store of documents, document search and usage statistics, differential document changes between versions, contributing authors |

Figure 14.1. Systems and possible attribute information

Replicating all of this information would be of little value. When you define attribute details, you need to determine what data elements are absolutely important for managing and operating your service offerings and infrastructure.

Increasingly, IT organizations are considering the concept of federation to address the need for a single way to access data being held in many different locations. Federation allows users to obtain the data they need without understanding exactly where the data resides or accessing multiple data stores. The required linkages to these various internal (and increasingly external) data stores reveal a federated data model. When you determine what to store (discover and maintain) in your CMDB and what to link (federate) to other data stores you will need to consider data maintenance and access needs. Consider the following:

- » **Frequency of information change** — If the information changes often, and real-time replication and synchronization cannot occur, a link from the CMDB to the data store may be more suitable to ensure the most up-to-date information is accessed in the best possible way.

- » **Frequency of information access** — If the information will not be accessed frequently, a link may be a better choice as it would minimize overhead of synchronization and still ensure timely access to the data when required.

Think of the CMDB as the source for the most important information you need to manage IT. Like a marketing brochure that highlights the key services of a company, the CMDB highlights the key attributes of CIs that support your strategic, tactical, and operations goals for your service and infrastructure.

The term “metadata repository” is often used by other disciplines to describe a repository that contains data about data. In a way, your CMDB contains data about the actual physical and logical IT asset data in your organization. Federating this information greatly simplifies the overhead of tracking and maintaining attribute information — a task often better suited to the federated systems. Your organization likely has made investments in a wide range of systems. Don’t try to duplicate this functionality by creating a super database. Aside from the technical challenges in creating and maintaining such a monster, you will undoubtedly face significant political issues in activating and managing this CMDB.

The tasks in this step will help you capture the most important CI information in a way that is simple to manage and easy to conceptualize.

Task-by-Task Guide

Figure 14.2 shows the tasks for defining CI attributes.

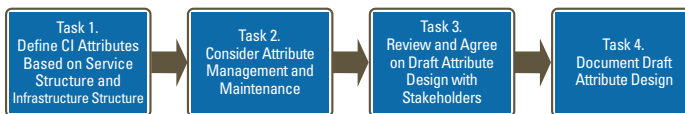


Figure 14.2. Tasks for defining CI attributes

Task 1. Define CI Attributes Based on Service Structure and Infrastructure Structure

For this step, you should use the same working group as used for leveling and relationship design, with the addition of individuals who have a data source perspective. These representatives will be aware of earlier decisions about required CIs and where CI information is currently collected and managed. Combine relationship and attribute design sessions when possible and practical. You may want to define relationships or attributes as a focus for each session, but expect both criteria to evolve iteratively.

The working group must discover and distill attribute requirements for both the service configuration structure and infrastructure configuration structure defined in your straw model structure. The group should systematically identify the additional information that you need to collect and design into your straw model to meet your requirements. Remember, the intent of attributes is to provide necessary characteristics: what CIs are in use, their status, and their location. The concept of CI lifecycles, discussed in step 21, “Create CI Lifecycle Management Processes,” introduces the idea of attributes that define the state of the CI. States are a useful attribute to consider for meeting your IT business process requirements.

Stage 2. Define Requirements and Create IT Service Model Blueprint

You will need broad stakeholder involvement to answer the fundamental question: “What is the value proposition for storing specific data in the CMDB?” For every attribute, ensure that it will actually be used. If you are uncertain, leave it out. It is far easier to add an attribute later than it will be to retire and hide this information in the future.

Start by defining attributes for your infrastructure configuration structure using the categories you defined in step 12, “Define Configuration Item Level and IT Service Model.” Categories, which are a special class of CI, have specific attributes that can be inherited from their parent CI or passed on to their child CI. Your service configuration structure is unlikely to need categories, so in most cases, service CI classes will not be candidates for inheritance. Figure 14.3 provides some examples of service configuration CI classes and their corresponding attributes.

| Service CI | Examples Attributes |
|------------------------------|--|
| IT Business Service Offering | <ul style="list-style-type: none">• Service support hours• Business user catalog entry ID• Cost per incident• Service owner/sponsor (could be a relationship to a role class)• Availability target (% for combination of all IT service offerings)• Incident restoration target (SLA end-to-end service)• Incident response time (SLA customer response time)• Customer representative (could be a relationship to a role class)• Service portfolio (could be a relationship to a service portfolio class) |
| IT Service Offering | <ul style="list-style-type: none">• Service support hours• Enterprise catalog entry ID• Cost per incident• Service owner/sponsor (could be a relationship to a role class)• Availability target (% for combination of all IT service component sets)• Incident restoration target (SLA or OLA)• Incident response time (OLA provider response time)• Integrator representative (could be a relationship to a role class)• Provider portfolio (could be a relationship to a provider portfolio class) |
| IT Service Component Set | <ul style="list-style-type: none">• Component support hours• Cost per incident• Provider (could be a relationship to a role class)• Component set availability target (% combination of all CIs in the set)• Capacity threshold |

Figure 14.3. Service CI classes and their corresponding attributes

Figure 14.4 provides some examples of infrastructure configuration CI classes and their corresponding attributes, along with subcategory examples. You may or may not require these additional levels of infrastructure configuration structure categorization, depending on the complexity and volume of noncategory CIs you need to manage.

| CI Class | Examples | Attributes |
|-----------------------------------|---|--|
| Infrastructure CIs and Categories | CI Category Example Attributes | Other CI Class (Noncategory) Example Attributes (Note: These could be category attributes if you want them to be inherited through a parent-child relationship and they were unique to that category.) |
| Application Software | <ul style="list-style-type: none"> • Code version • Code language • Build • Compiler version • DSL ID (could be a relationship) | <ul style="list-style-type: none"> • Presentation layer module • Presentation logic layer module • Business logic layer module • Data access layer module • Data layer module |
| Support Software | <ul style="list-style-type: none"> • Operating system version • Build • Image version • Virus definition file | <ul style="list-style-type: none"> • Wintel operating system • UNIX operating system • Virtual server software • Antivirus/spam/phishing software • Backup software • Server/workstation base image |
| Hardware | <ul style="list-style-type: none"> • Manufacturer • Make • Model • DHL ID • Serial number • MAC address • IP address (fixed) • Firmware version | <ul style="list-style-type: none"> • Wintel server • UNIX server • Desktop • Laptop • Tablet • Peripheral <ul style="list-style-type: none"> • Network printer • Network attached storage • Network devices (router, etc.) |
| Data | <ul style="list-style-type: none"> • Client ID • Location ID • Address • Postal/ZIP code • Organizational identifier | <ul style="list-style-type: none"> • Client data • Location data • Organization data |
| Service Unit | <ul style="list-style-type: none"> • Bandwidth threshold • Cost per month • Options | <ul style="list-style-type: none"> • Network unit • Desktop unit |
| Process | <ul style="list-style-type: none"> • Master process name (could be a relationship) • Process owner (could be a relationship) • Contingency description | <ul style="list-style-type: none"> • Service request • Work instruction • Procedure |
| Standard | <ul style="list-style-type: none"> • Effective date • Exemption details | <ul style="list-style-type: none"> • Security standard • Policy • Procurement standard |
| Documentation | <ul style="list-style-type: none"> • Version number • Author • Editor • Review process | <ul style="list-style-type: none"> • Support documentation <ul style="list-style-type: none"> • Build kit • Contingency plan • Service blueprint • Service agreement |
| Facilities | <ul style="list-style-type: none"> • Physical location • Facilities contact (could be a relationship) • Geographic region | <ul style="list-style-type: none"> • Data center • Remote office • Consolidated office |

Figure 14.4. Infrastructure CI classes and their corresponding attributes

Task 2. Consider Attribute Management and Maintenance

After you have determined the additional attributes you need, you should also consider the following questions to help you identify federation requirements, population approach, and management and maintenance of CMDB data.

What are the data sources that will compose your federated data model?

The answer to this question will help you establish the possible system owners or representatives who will need to understand the data to be replicated as well as the frequency. If you need to extract data from sources that are held to certain privacy or security standards, you may need to involve a policy decision-maker. In many cases, you can leverage these systems' data exchange protocols and application programming interfaces (APIs). In a perfect world, every system would include a Web services interface, allowing structured information to be exchanged through predefined requests. More likely, you will need to integrate some legacy data into your CMDB. Consider this requirement when you select your CMDB enabling technologies.

Where and how will the CMDB data be used in other systems?

CMDB information is a critical enabler for integrating ITIL and other processes. Access to CMDB information will be necessary to process transactional information in some of these systems; therefore, the CMDB and other systems may require tight integration. The federated CMDB data store may need to feed into some of these systems. For example, incident management records include CI data, and the service desk might require direct CMDB access to assess impact and risk for incidents.

How will you populate the CMDB with attribute information?

When automated population is not possible, you might need resources to manually add attributes.

How can you simplify management of attributes through inheritance?

Attribute management is an important part of your CMDB design and use. Attribute types provide a mechanism for supporting and simplifying attribute management. To explore types of attributes, consider three basic types: core, class, and custom. (See Figure 14.5.)

| Attribute Type | Definition | Example |
|----------------|--|---|
| Core | Mandatory attributes that apply to every CI in the CMDB. | <ul style="list-style-type: none"> • Name • Owner • Label • Description • Created date • Last modified • Version • Lifecycle group identifier |
| Class | Supporting attributes that are unique to a class of CIs. Categories are a special class of CI that pass on attributes based on inheritance, through parent-child relationships. | <ul style="list-style-type: none"> • Server <ul style="list-style-type: none"> • Make • Model • Manufacturer • Serial number • Document <ul style="list-style-type: none"> • Version • Author • Editor |
| Custom | Complementary attributes unique to a specific CI (instance) to provide detailed information not expressed in core- or category-based attributes. | <ul style="list-style-type: none"> • Server ABC123 <ul style="list-style-type: none"> • Last security audit date/time • Operator minimum security clearance level |

Figure 14.5. Basic types of attributes

When you look at the class attribute type, you see attributes that further support the mandatory core attributes, but are unique to that class of CI. Some useful opportunities emerge when you apply the concept of inheritance to attributes.

For example, assume that CIs have been grouped within a category, as outlined in step 12, “Define Configuration Item Level and IT Service Model,” and are broken down using the parent-child relationship outlined in step 13, “Define Configuration Item Relationships.” You could establish the highest level CI in the hardware category as “hardware.” The hardware CI category could take on some category-specific attributes, such as installation location, accessibility, dimensions, etc. All child CIs of this hardware category could inherit these parent attributes and add their own to the mix.

Assume a server is one of the child CIs of the hardware category CI. You could define a new category of CI called “server,” and you could add additional attributes that apply to all servers, such as make, model, storage capacity, etc. All children of the server category could be the individual servers, and they would inherit attributes from both the hardware category and the server category. By adopting a category hierarchy model such as this, your CI information becomes self-organizing, to a certain degree. Figure 14.6 illustrates this example.

Stage 2. Define Requirements and Create IT Service Model Blueprint

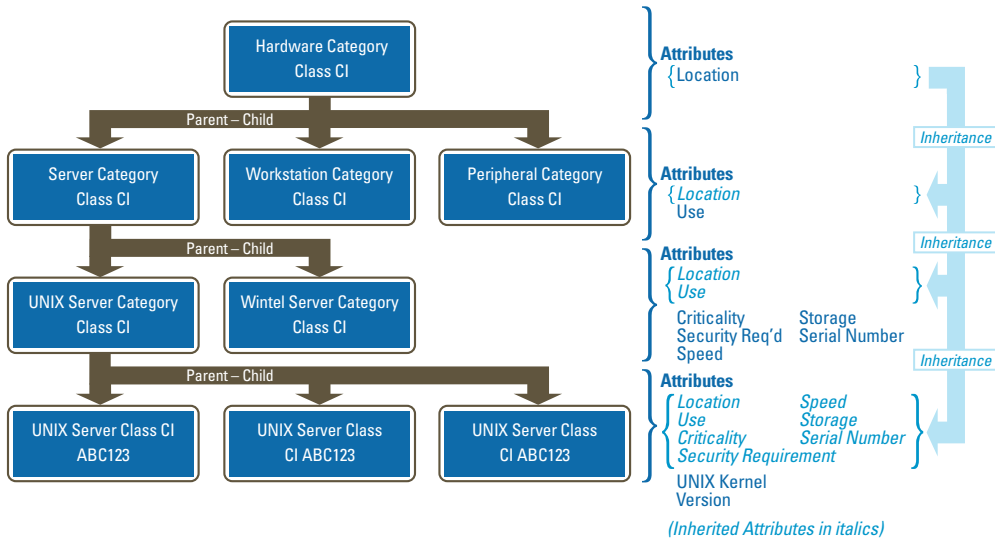


Figure 14.6. A category hierarchy model

Figure 14.7 shows a slightly different perspective of inheritance: categories in which a class CI (not a category CI) is used to manage attributes of the Wintel and UNIX server CIs.

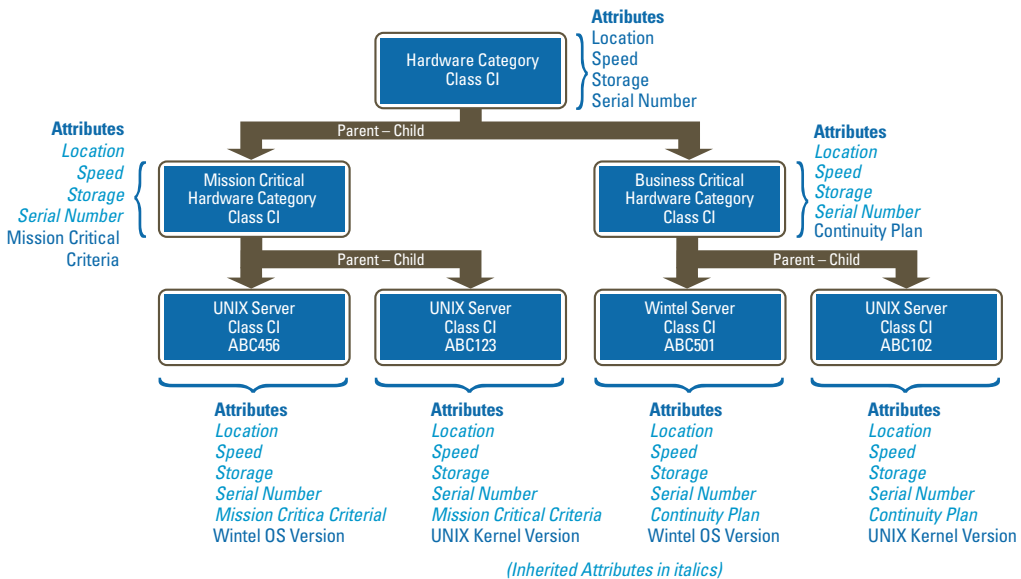


Figure 14.7. Using a class CI to manage attributes

The attributes outlined in Figure 14.3 do not include any data until you have created and populated the CMDB with the specific CIs. As such, inheritance relates only to the data types and not to the populated values within these types. Child CIs will not assume the parent attribute values by default; you will need to populate them accordingly. Classifying CIs using the concept of inheritance will help you consistently define your CMDB data model and simplify attribute management.

You should strive to use the core and class attribute types for the majority of your attributes. Custom attributes should not become the norm and should reflect the need for specific, detailed CI information.

Task 3. Review and Agree on Draft Attribute Design with Stakeholders

In many cases, it is unnecessary to review the draft attribute design with the broader service catalog and IT business process stakeholders at this point. However, if there are requirements assumptions underlying the design of specific attributes, it would be wise to discuss these with the stakeholders that identified the information need. This ensures that the attribute design considerations are valid and concise as you move into the final design step to consolidate the levelling, relationship, and attribute information. Broad stakeholder validation is reserved for step 15, where the final proposed model is reviewed and validated. Be sure to also document any emerging technology requirements or limitations that you have identified during your discussions.

Task 4. Document Draft Attribute Design

Now that you have created a list of draft attributes for your IT service model, ensure that the attribute fields are documented adequately to capture your attribute fields, rationale, and any key design considerations. This information, combined with your design notes from steps 12 and 13, will be compiled into reference material in the next step as part of the IT Service Model Blueprint. Remember, you can further refine these draft attributes and create your final attribute design when you couple them with the CI level and relationship designs, which will be brought together and finalized as part of step 15, “Design IT Service Model Blueprint.”

Summary

Attributes are an important enabler in your CMDB. They are the data elements that characterize your IT assets, and they give you the final elements of decision-support information necessary to manage your services and IT business processes effectively. Although you can expand your attributes later, give adequate consideration to what attributes are essential to characterize the CIs in use, their status, and their location. You also must consider federation and simplified attribute management.

Technical, organizational, and even political or cultural factors will influence your decisions. There is no one-size-fits-all answer. Each organization must determine its preference, and you should seriously discuss and debate these questions to reach informed decisions.

Step 15

Design IT Service Model Blueprint

Goals and Objectives

In this step, you will construct the IT service model blueprint by using the requirements you have identified for the service catalog, the IT business processes, and the IT service model design. The documented blueprint should include the guidelines and practices you used to determine the required CI structure, relationships, and attributes. The blueprint should also detail the governance and mechanisms you will use to continually improve and refine the model in response to evolving business requirements. At the end of this step, you will have a documented and approved IT service model blueprint that meets your current and projected information needs; facilitates communication and awareness of the CMDB; and provides the details you need to select a CMDB tool, as well as to construct and populate your CMDB.

Overview

Let's compare the process of building your CMDB with building a new home. You wouldn't even consider constructing your new home without a well-documented and agreed-upon blueprint that provides details and clarity about factors that influence construction details and coordinated activities. The straw model structure that you created in the preceding steps is analogous to a document where you have defined the desired total square footage of your new home, the number of levels, the number of rooms, each room's square footage and layout, the electrical requirements, size and placement of windows, etc.

Stage 2. Define Requirements and Create IT Service Model Blueprint

Just as a builder or architect would translate these requirements into an architectural diagram before building your home, you should follow the same approach and represent your requirements in a consolidated IT service model blueprint before building your CMDB. Once completed, the blueprint will clearly show the service configuration structure, the infrastructure configuration structure, and the levels required to store your CI information. Your blueprint also will detail the relationships between CIs to ensure that dependencies are understood and enable both interrogation and analysis from the perspectives of services and infrastructure. The blueprint will further define what critical attributes you need to store and where.

In the preceding steps, you iteratively completed most of the design, distillation, and rationalization of your IT service model. Now you must ensure that you integrate all of the considerations and details into a final blueprint and validate them against the entire set of identified requirements. Validating the blueprint with your stakeholders will help to guide any final refinement, identify and inform CMDB construction and population planning, and ensure broad stakeholder understanding of the integrated IT service model blueprint. Finally, the tasks in this step will result in a blueprint that has been agreed to by the PEB, and which you can publish. After completing this step, you can proceed with building your CMDB, confident that you are constructing a CMDB that directly supports your organization's needs.

Task-by-Task Guide

Figure 15.1 shows the tasks for designing the IT service model blueprint.

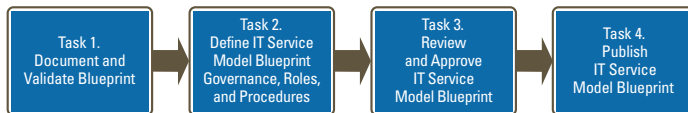


Figure 15.1. Tasks for designing the IT service model blueprint

Task 1. Document and Validate Blueprint

Your blueprint is the mechanism for documenting design details and should include any information that can provide guidance in the future when you consider new CI levels, categories, relationships, and attributes. Figure 15.2 illustrates the core areas of the blueprint that you need to document and suggests key content that you should consider.

The blueprint diagram contains three columns of information: attributes, CMDB level and structure, and relationships. Keep in mind that the blueprint should not be a document showing all CI instances across your organization, but rather a template of the agreed upon standard structures, relationships, and attributes for your organization.

SERVICE MODELS: ENABLING A TOP-DOWN APPROACH TO MANAGING IT OPERATIONS

Bob Balassi, CTO of Maryville Technologies, discusses how service models can help you improve the business relevance of IT services and better deliver business value.

Most organizations are moving away from a bottom-up or infrastructure management approach to IT operations so they can better align IT with business strategy. The service model offers a simple yet powerful approach that ties the operational management of systems to the needs of the business.

A top-down approach ties strategic business processes to IT services to service level objectives to operational plans, and finally, to infrastructure and system level performance. The top-down service model approach can then be expanded across three key areas of IT management focus:

- > Availability and performance management
- > Cost management
- > Compliance and controls

After the appropriate service level objectives have been defined, the service model becomes a blueprint — guiding the instrumentation of your infrastructure. The key is to instrument the infrastructure to capture the appropriate data in the CMDB related to availability, performance, and service costs, and to monitor the IT controls or compliance aspect of that service.

By performing a decomposition of these three areas, you can identify the data that is required in your CMDB and the depth you should track within a service model. A spectrum of choices must be evaluated to balance costs and benefits.

At one extreme, you could take a logical perspective and create a very high-level service definition. Perhaps you deliver an IT service to four geographies. A simplistic service model would take all of the underlying technology infrastructure associated with each geography and roll up all activity to that lowest level. That service model provides an information flow, but it is not very detailed or functional from an IT operations perspective.

At the other extreme, your lowest level of the service model could be every physical component within the infrastructure that is in any way related to service level objectives. You could also create logical components within that service model, if you need to track data against that logical component or if you want to initiate particular actions when the logical component has an impact. The highest level might start at the application and continue down to all the physical devices in the infrastructure.

Whether you take a logical or physical approach to service modeling, the first step in creating a service model is to identify the physical and logical components that populate the CMDB. Capture every physical component, including server and network hardware, operating systems, and applications. Additionally, capture all of the organizational or functional components that relate to the service; for example, the various administrative teams such as the network administrators. These two data groups create the total domain of components.

Remember to constrain yourself to the service topology. Don't try to capture everything. Focus on the components that are required to deliver and manage the services that are important to the business.

Establishing a service model helps IT organizations communicate to the business in terms of the services that they're consuming, instead of technology details that often have little meaning to them.

Excerpt from "Service Models and the CMDB," VIEWPOINT, Focus on: CMDB, published by BMC Software.

Stage 2. Define Requirements and Create IT Service Model Blueprint

The center area of Figure 15.2 represents the CMDB level and structure decisions that you made in step 12, “Define Configuration Item Level and IT Service Model.” The numbered circles correspond to the following:

1. The top part of the CMDB level and structure blueprint defines your IT service configuration structure.
2. The service configuration structure consists of end-to-end services that IT provides to the business.
3. IT service offerings represent the elements of an end-to-end service and are combined and reused to deliver business value.
4. The bottom part of the CMDB level and structure blueprint represents the infrastructure configuration structure of your CMDB.
5. The IT component sets link your IT service configuration structure CIs (that is, your IT service offerings) to groupings of specific IT components.
6. The IT components are maintained in predefined category and subcategory structures.
7. The specific IT components that represent the actual infrastructure IT assets are children of your parent category CIs.

The left side of Figure 15.2 represents the CMDB attributes and attribute classes that you defined in step 14, “Define Configuration Item Attributes.” The numbered circles correspond to the following:

8. The top part of the CMDB attributes blueprint defines the service CI classes and attributes. Service attributes likely will represent service-specific information, such as service targets, capacity thresholds, and availability objectives.
9. The bottom part of the CMDB attributes blueprint defines infrastructure configuration structure category and component classes and attributes. The infrastructure CI class shows a component set class that is not likely to have had any attributes assigned. You should document the category and subcategory classes to name all of the categories that you have selected for your organization. This list also should name any of the inheritable attribute fields related to that category. Document any other classes that you designed in step 14, showing the class name and attribute field names.

The right side of Figure 15.2 represents the CMDB relationships that you defined in step 13, “Define Configuration Item Relationships.” The numbered circles correspond to the following:

10. The top part of the CMDB relationships blueprint defines the service CI relationships required to establish and manage your service configuration structure.
11. The bottom part of the CMDB relationships blueprint defines the component set relationships that have specific IT components, the specialized parent-child relationship between categories and components, and any other relationship types between components that your organization requires. These additional relationships allow you to define dependencies that cross component sets and/or categories, so that you can define other views of your infrastructure configuration structure CIs.

CMDB Level and Structure

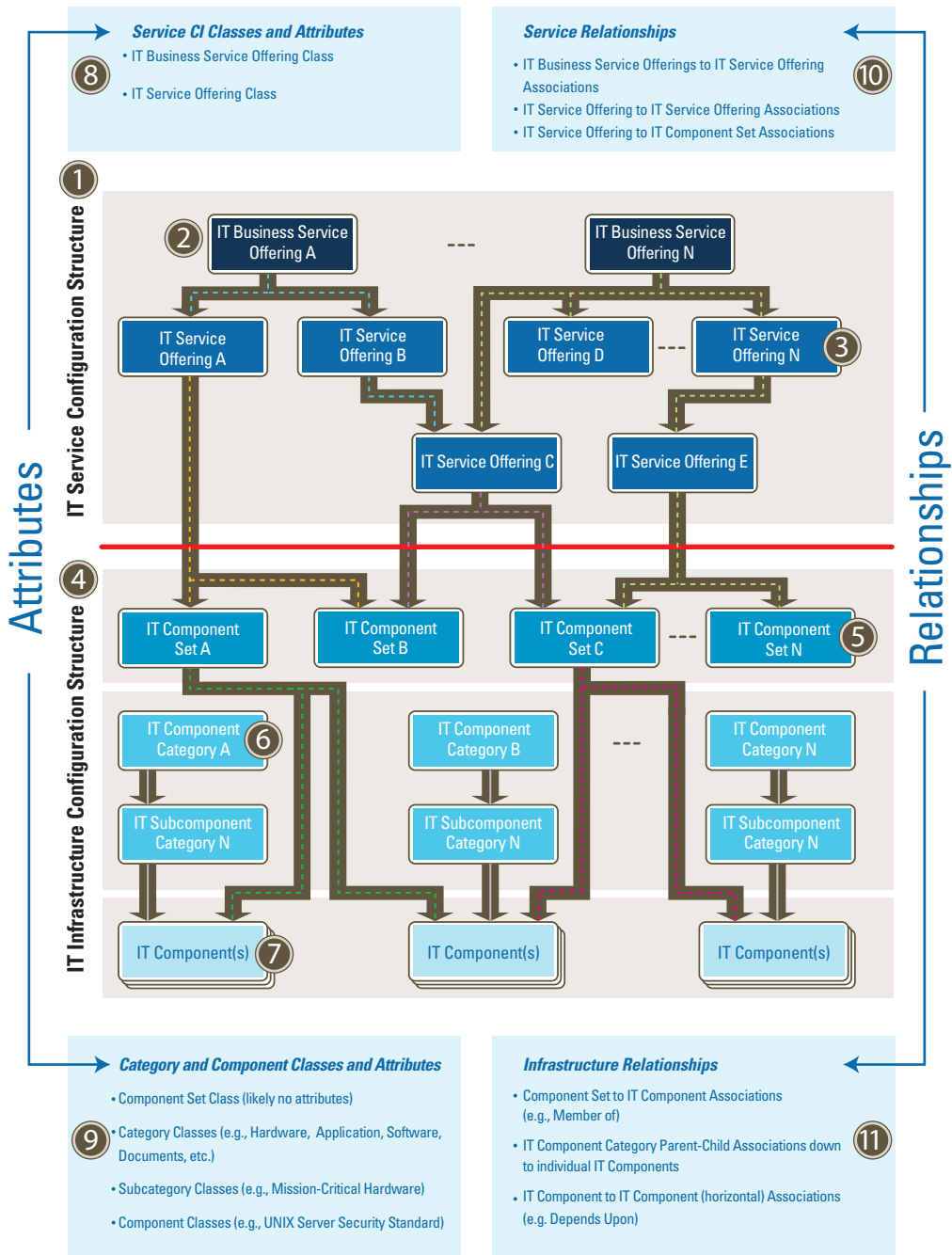


Figure 15.2. IT service model blueprint

Stage 2. Define Requirements and Create IT Service Model Blueprint

Figure 15.3 provides an example of a blueprint for a specific business service, sales support. The figure shows the IT service configuration structure and the IT service offerings — enterprise service support and point of sale solution — that enable the business service offering. The diagram also shows the IT service configuration structure, breaking down the IT service offerings to the component level.

Once your IT service model blueprint shows all of your design specifications, you need to test and validate the blueprint. Run through a series of real-world validation scenarios related to your service and infrastructure CIs, so you can be confident that the structure, levels, categories, relationships, and other attributes are sufficient to meet both the service catalog and IT business process requirements. This validation scenario approach, which you can undertake by creating specific use cases, will provide useful information for validation sessions and communication material. Although you will ultimately document your blueprint on paper, it might be helpful to use a multidimensional modeling tool that facilitates validation and simulation exercises.

Also include in your blueprint proposed guidelines for the CMDB level, attribute, and relationship design. You may wish to translate these guidelines into formal policies following final approval of the IT service model blueprint. Policies can help ensure that the data added to the CMDB adheres to your design specifications. Whether you use formal policies or stick with guidelines, these should be owned and managed by your configuration management process owner.

Step 11, “Define CMDB Requirements to Support Other Processes,” introduced the concept of ability statements as a means of understanding requirements. Now, using the ability statements and your blueprint, hold validation sessions with the requirements stakeholders to:

- » Ensure the blueprint adequately reflects requirements
- » Communicate any trade-offs that were made to meet enterprise needs
- » Promote awareness of the blueprint and the driving principles
- » Collect any priorities for population that may affect your plans for CMDB construction and population

When you have completed the validation, record the final refinements in the blueprint.

Task 2. Define IT Service Model Blueprint Governance, Roles, and Procedures

At this point, you have put a lot of thought and effort into designing an IT service model blueprint that meets the current and emerging needs of your IT business. However, you will need to revisit the model regularly to ensure that it continues to evolve to meet the ever-changing needs. At some point you might need to create additional relationships and attributes to support new service catalog and business process needs. Regular review will also ensure that you identify and retire information that is not used or is no longer required, so that you can maximize productivity in collecting data that is valued and useful.

CMDB Level and Structure

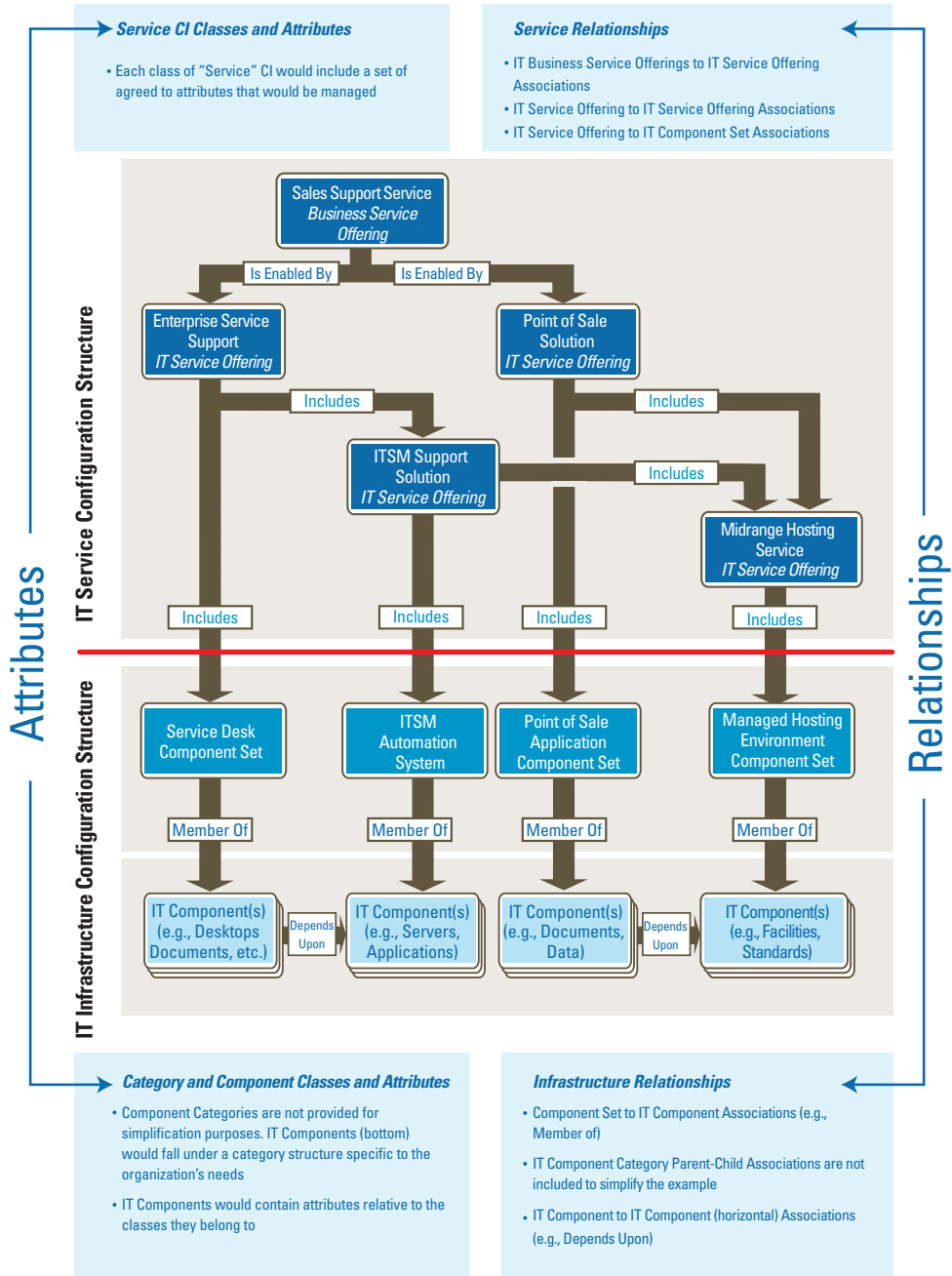


Figure 15.3. Example Blueprint

Stage 2. Define Requirements and Create IT Service Model Blueprint

Be sure to consider how you will identify, analyze, and implement new and changing requirements. You also will need to establish governance mechanisms and integrate them with your configuration management processes. Clearly defined and documented roles and streamlined procedures will help you maintain the integrity of your model while modifying it to enable use and evolution.

Task 3. Review and Approve IT Service Model Blueprint

After you have a consolidated blueprint of the CI structure, attributes, and relationships that will make up your CMDB, ask the PEB to review the blueprint for final approval prior to publication. The final document should identify any rationalizations for decisions (trade-offs) regarding CI levels, attributes or relationships versus CI, CI groupings, etc. Also include any requirements that are not being met at this time and your plans to address those requirements in the future. In addition, detail your plans for continuous improvement, the frequency of revisions, and the mechanisms and process for formal modifications.

Task 4. Publish IT Service Model Blueprint

Once the PEB has approved your IT service model blueprint document, you can officially publish it. As part of your integrated CMDB project communications, the CMDB IT service model blueprint and the governance and guidelines for modification should be broadly distributed to the individuals who will be active contributors and consumers of the IT service model information. You will also use the blueprint as a guide for tool selection, CMDB construction, and CMDB population.

Summary

You have achieved a major accomplishment in your CMDB project: an IT service model blueprint tailored to your organization. You have ensured a comprehensive and extendible design for your IT service model through engaging key stakeholders and utilizing clearly articulated service and IT business process requirements. This well-designed IT service model blueprint will be the authoritative source for CMDB construction and population planning. Further, when coupled with effective control mechanisms, this blueprint will ensure your CMDB houses the critical information you need to manage your services and infrastructure today and well into the future.

STAGE 3

Select CMDB Solution and Tools

Stage 3. Select CMDB Solution and Tools

During stage 3 of your CMDB project, you will complete the planning process and technology selection so that you can move ahead with building, populating, and deploying your CMDB.

In stage 1, “Assemble the Project Team and Define the Project,” you collected information step by step to build a business case that was approved by the PEB and project sponsor. In stage 2, “Define Requirements and Create IT Service Model Blueprint,” you worked with key stakeholders to document and obtain approval on a broad range of solution requirements.

In this stage, you will select your CMDB solution after a careful analysis of the requirements and solution alternatives. You will plan the population of the CMDB and map CIs to data sources, determine your federation strategy, and set business rules for automating the management of multiple data sources. That planning will guide your selection of additional tools to automate the initial population of your CMDB and the ongoing update of key CI information. Finally, if required, you will complete a full project ROI calculation.

Figure 1 shows the steps for selecting your CMDB solution and tools. Each step contains specific goals that you should meet before moving forward.

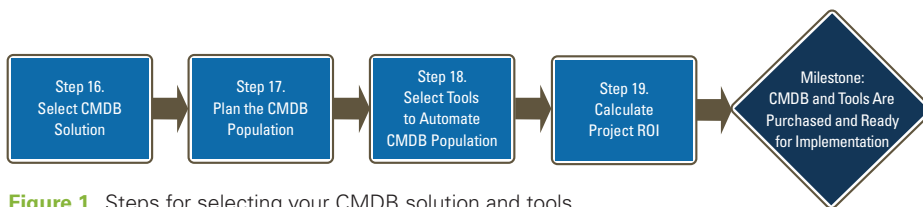


Figure 1. Steps for selecting your CMDB solution and tools

You may want to use the following as a checklist.

Step 16.

Select CMDB Solution

Goal: Select the CMDB and supporting technologies, and procure the components of the solution. Careful consideration of requirements and available solution options is key to procuring a solution that will ensure project success.

Completed: ____

Step 17.

Plan the CMDB Population

Goal: Carefully plan the population of the CMDB, including federation strategy and mapping CIs to data sources. Documenting business rules that enable automated handling of precedence, reconciliation, and normalization is key to successful initial population of the CMDB and ongoing data accuracy.

Completed: ____

Step 18.

Select Tools to Automate CMDB Population

Goal: Identify and purchase additional discovery and automation tools to aid CMDB population. A broad range of existing and new IT tools can help automate CMDB data management.

Completed: ____

Step 19.

Calculate Project ROI

Goal: Use data collected from the previous steps to create a financial analysis of the expected costs, benefits, and ROI for the project. This information will help guide the project implementation team as you move into stage 4, "Construct and Maintain Your CMDB."

Completed: ____

Milestone: CMDB and Tools Are Purchased and Ready for Implementation

You have purchased all the components of your CMDB solution and planned the CMDB population, and you have completed a full financial analysis of the cost and expected benefits of the solution.

Milestone achieved: ____

Step 16

Select CMDB Solution

Goals and Objectives

In this step, you will form the team that will identify and select the CMDB and supporting technologies, and procure the components of the solution. This activity includes documenting your requirements for the technologies, based on the project scope and objectives. Then, the team can review the tools currently in use to identify whether they can meet these requirements. If not, then the team should create and distribute a request for proposal (RFP) so that you can procure solutions that do meet the requirements. Finally, you will present a solution review to the PEB.

Overview

You will need to select a tool for the CMDB itself as well as the supporting technologies that are part of the overall CMDB solution. The CMDB you choose may be an extension of an existing in-house solution, or it may include a new set of tools as part of a suite. The key to success is to first select the best possible CMDB to meet your needs within the budget that was approved when you presented your business case. Then add any additional tools (discovery or visualization, for example) to meet requirements, as your budget permits.

The previous steps provided a clear guide to the structure and content of your CMDB, as well as various functional requirements. In this step, you'll form a cross-functional team to review the previously documented functional requirements and your current solution capabilities to create a detailed requirements list that you will use to create an RFP. Be sure to review what is available, and consider extending and using what is currently in place before you conclude anything. Remember, selecting your CMDB software is a single-shot opportunity, so be careful. Do not rush into it, and be sure to consider all options before deciding.

Task-by-Task Guide

Figure 16.1 shows the tasks for selecting technologies.

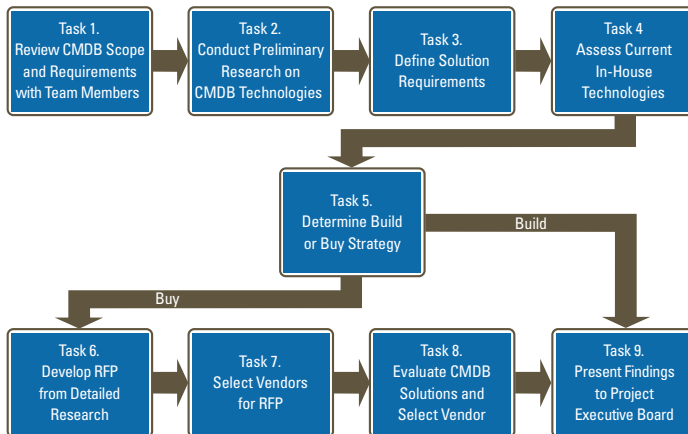


Figure 16.1. Tasks for selecting technologies

Task 1. Review CMDB Scope and Requirements with Team Members

First, you'll need to create a solution review and procurement team. The selection of a CMDB solution will most likely involve specialists who have not been involved in the project yet. You should include an overall solution architect as well as technical specialists from various technology domains, including security, network, open systems, applications, and perhaps mainframe.

The first order of business with this team is to review the project objectives. Make sure the solution review team fully understands the objective, scope, and requirements so the team can develop a solution that fits. Therefore, each team member should read the business case written in step 5, "Build a Business Case," which outlines the goals, objectives, problem, solution, and cost benefit analysis. This team will also need to review the complete set of materials created in stage 2, "Define Requirements and Create IT Service Model Blueprint." If you don't already have one, consider setting up a shared directory, including all materials collected while documenting requirements for the solution review team.

Task 2. Conduct Preliminary Research on CMDB Technologies

This task uses the same framework to build on the exploratory research conducted by the team in step 2, "Obtain CMDB Knowledge." Now that you have documented the detailed requirements for your project in stage 2, the team, including the new members responsible for selecting the technology, should conduct more detailed research on what is available. The selection team needs to understand the broad range of products in the marketplace and, more importantly, the high-level functionalities of these products. Gaining this high-level understanding will prepare the team to better define the capabilities required for the CMDB technologies.

Following are some suggested sources of information:

- » Vendor product publications
- » Vendor Webinars
- » White papers and technical articles found on the Internet
- » IT service management conferences and vendor booths
- » Research papers from IT analyst firms

It is important to develop a functional framework of solution components. Some vendors offer suite solutions, while others offer specialized products. As you review capabilities, look for relationships, integrations, and similarities among the tools. If you find any, then you may want to group these tools into a set or suite. Think along the lines of an office suite where all of the components — such as a word processor, calendar, spreadsheet, and presentation manager — are automatically integrated and work as one multifunctional resource. This integration may influence your choice of suppliers. If your requirements call for a broad range of capabilities, a supplier that provides a suite of integrated tools and technologies could be a better choice than a group of “best of breed” suppliers whose tools need integration.

The CMDB solution market is maturing, but vendors may not all share a common vocabulary. To prevent terminology confusion, categorize solution elements by basic function. You may not need all the capabilities that are offered. Figure 16.2 and the following discussion highlight the core functions that are part of a CMDB solution.

User interface. These capabilities are designed for the configuration manager and additional IT personnel, who need to define, manage, and use the CI attributes and relationship data in the course of their IT operations or process management functions. These capabilities include:

- » **CI administration** — These capabilities are used by the configuration manager and CI owners who are responsible for creating CIs and maintaining accurate attributes and relationship information for business services, IT systems, and related CIs in the CMDB. Some of the functions include definition and maintenance of business rules to automate tasks such as tracking CI changes or CI data reconciliation, creating batch processes to automate bulk import of CIs into the CMDB, and interactive editing of CI properties and relationships.
- » **Reporting and querying** — Search and reporting capabilities enable the analysis of individual CIs or a collection of CIs (a business service), such as detecting CI changes and comparing different CI versions, or trending of CI characteristics based on CI service records or analysis of CI dependencies for change and incident management. Intelligent querying capabilities should allow users to search CIs and related CIs in the data model as objects and not as data elements in database tables.
- » **Visualization** — These capabilities provide graphical representation of dependencies and relationships between CIs as infrastructure components, and IT business service offerings. The solution should include the ability to graphically map CIs and view them as IT systems and as business services. The visualization tools should facilitate service impact modeling as well as cause-and-effect analysis. The visualization capability should include

Stage 3. Select CMDB Solution and Tools

the ability to draw and print the CIs in a hierarchy structure (e.g., from top to bottom or from left to right) or in a star cluster format. Users should also be able to filter data out, as well as pan and zoom.

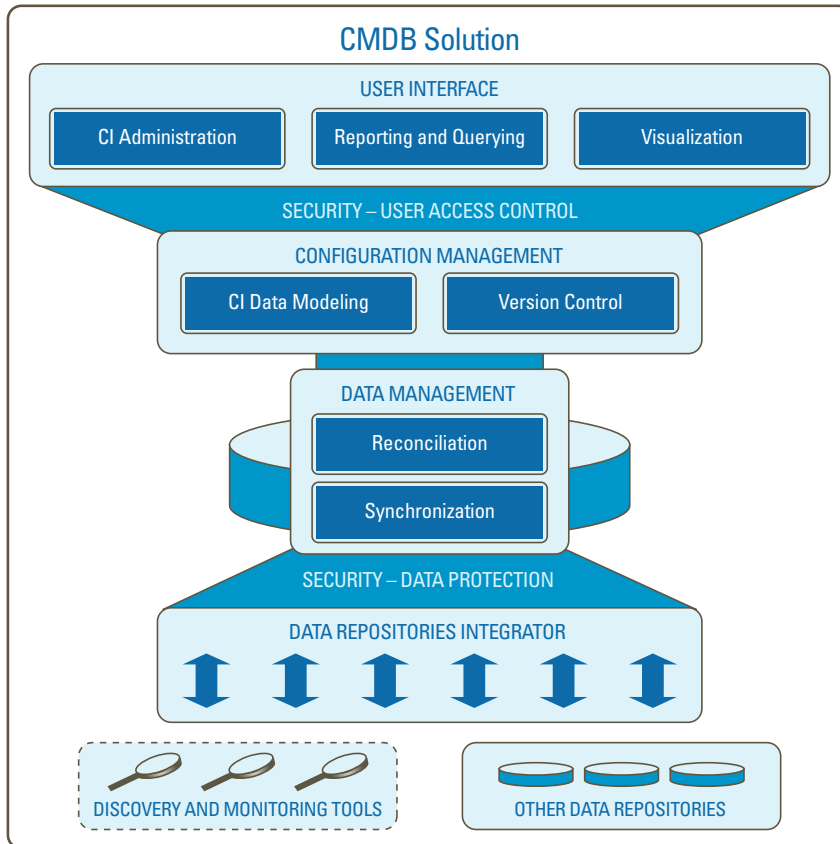


Figure 16.2. CMDB core functions for consideration

Security. Security includes both user access control and data protection. The solution should include system-level security related to the application, network, database, and support security standards. IT should support the definition of user profiles for different configuration management functions: CI owner, CI user, and CI supporter. IT also should define access rights related to specific process roles, such as change manager or service desk technician.

Configuration management. In a CMDB, configuration management includes CI data modeling and version control:

- » **CI data modeling** — A predefined object model based on industry standards provides a default set of CI properties and relationships and includes properties such as hierarchy and inheritance. Some solutions treat the CMDB as a database full of data elements and are not object oriented. However, more extensive CMDBs are very difficult to manage without an object schema that relates CIs in the context of their environment and use.

You can apply data modeling standards to a CMDB, and your CMDB data model should include out-of-the-box relationships and hierarchies that can be easily modified. The solution should include best practice templates for CIs of varying complexity, such as entire systems, releases, single hardware items, software modules, or hierarchic and networked relationships between CIs.

- » **Version control** — This function provides a unique identifier to reference the changes of a specific CI or a collection of CIs that support a service that IT provides to the business. Version control includes capabilities to support automated or manual snapshots and rollback to a previous known baseline configuration, if necessary. It should support the automatic updating and recording of the version number of a parent CI if the version number of any child component CI changes. Versioning should enable the viewing of a history of CIs, should allow the comparison of the current state to a previous known good state at a very detailed level, and should provide tools to reduce the manual work of comparison. Related software release and distribution tools should be able to leverage CMDB versioning capabilities to roll back configurations to a previous known good state.

Data management. Data management functions in a CMDB should include, at a minimum, reconciliation and synchronization:

- » **Reconciliation** — The purpose of reconciliation is to take various representations of the same data and to bring it together. Reconciliation is used in three ways: 1.) In the identity mode, you take an instance from different sources and show how they are the same. 2.) In a merge, you take data from two different sources and decide which data wins (either attribute by attribute or an entire data set). 3.) In compare, you take data from different sources and say how they are different.

In reconciliation, there is no interpretation, no conversion, and no changing. You put together rules. Setting up reconciliation rules to define precedence and automatically reconcile data variance is critical to keeping data accurate and useful.

- » **Synchronization** — Replication and synchronization capabilities are used to push and pull data between the CMDB and different data repositories.

Additionally, data management functions should support bulk CI creation, modification, and deletion.

Data repositories integrator. The data repositories integrator consists of adaptors to integrate the CMDB to various data repositories through APIs, standard data exchange formats such as XML, or standard Web services or direct custom integration methods. These data repositories should include the ability to integrate data from the processes identified in step 11.

Discovery and monitoring tools. These tools may or may not be part of the CMDB solution (depending on the vendor offering). They are used to automatically collect CI attribute data, relationship data, and usage data for population into the CMDB. Discovery tools collect network, application, and hardware configuration and topology information. Business services, however, cannot be collected automatically. You must define them manually using the collected IT infrastructure and application configuration information. Monitoring tools collect usage and status information for applications and infrastructure CIs.

Stage 3. Select CMDB Solution and Tools

Other data repositories. In addition to the data repositories from discovery and monitoring tools, other sources store information related to people, processes, and documents in their own data repositories. These may include:

- » People information from HR systems
- » User ID and password information from directories such as LDAP
- » Incident, problem, change, and service-request ticket information from IT service management tools
- » CI contracts and policies from a document management system
- » Business process information from computer-aided software engineering (CASE) tools

By integrating with other data repositories, you are, by default, integrating with other processes. A final major consideration for the technology selection team is to determine the overall integration strategy used with these other data repositories. There are three general approaches to establishing a CMDB:

- » **Integrated multiple data repositories** — Multiple databases from various IT systems are directly linked to each other to facilitate point-to-point CI data exchange. In this scenario, there is no central repository. Instead, depending on the information required, the user must know which database holds the required CI information. (See figure 16.3.)

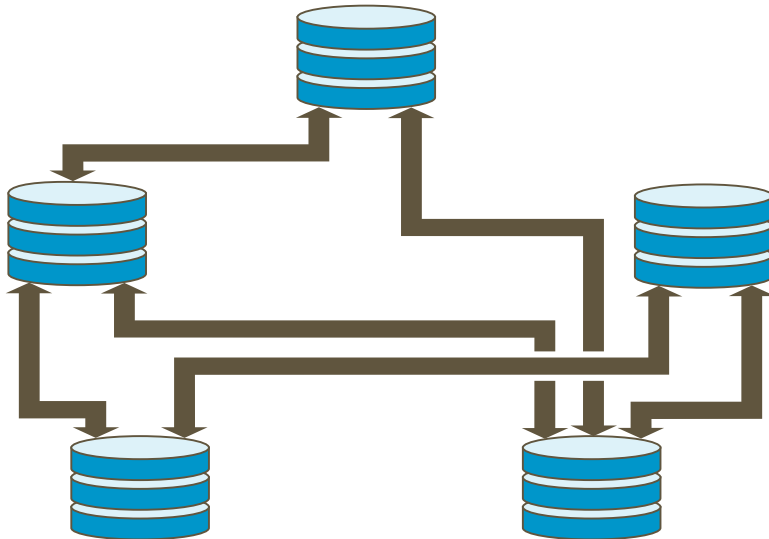


Figure 16.3. Integrated multiple data repositories

- » **One single centralized data repository** — All CIs and related data are stored in one database, similar to building a data warehouse.
- » **Federated data repositories with one central data repository** — One central database stores key data for all CIs and links to other databases from various IT systems, providing more detailed attribute data and relationship information. Access to the CI data is obtained through the central database. (See figure 16.4.)

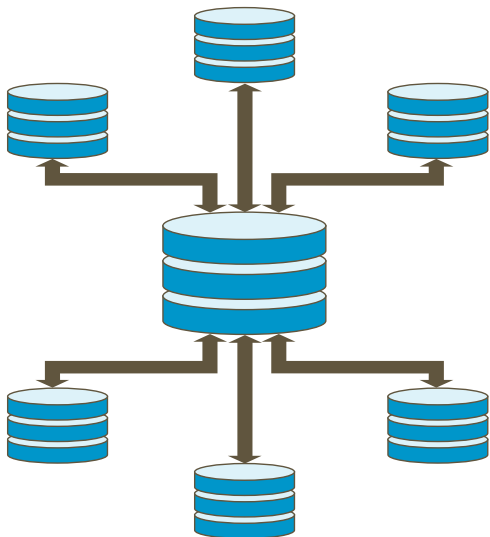


Figure 16.4. Federated data repositories with one central data repository

Figure 16.5 highlights the pros and cons of each approach.

| CMDB Type | Pros | Cons |
|---|---|--|
| Integrated multiple data repositories | <ul style="list-style-type: none"> • Low initial investment • A relatively easy and fast method to establish data exchange between a few data repositories | <ul style="list-style-type: none"> • Reconciliation and synchronization of data would be a challenge when the number of data repositories grows beyond a few databases • “Book of record” for CI data is stored in multiple data repositories, potentially making it difficult for users to easily find trusted CI information |
| One single centralized data repository | <ul style="list-style-type: none"> • One place to find all CI data • Centralized control of changes to data model • One point of entry to find all CI data | <ul style="list-style-type: none"> • Technical complexity of implementation requires long lead time • Performance of data repository will become a challenge as the volume of data increases over time |
| Federated data repositories with one central data repository | <ul style="list-style-type: none"> • Solution is scalable • Impact of integration to other data repositories is minimized • Increased control of CI data for definition of business services | <ul style="list-style-type: none"> • Many reporting tools cannot easily access the data stored in the federated data repositories through the central CMDB • Many federated CMDB products have not reached full maturity in the industry |

Figure 16.5. The pros and cons of each approach to establishing a CMDB

Although the federated approach is generally recommended, be sure to match the solution to your specific requirements. If your organization requires a partial CMDB solution, and not a full CMDB, you might be able to extend the functionality of an existing asset management or service management solution with additional functionality and discovery information. However, if you require integration of the CMDB data into existing IT and non-IT systems, a federated approach is preferred.

FEDERATED CMDB — MYTH, REALITY, REQUIREMENT, OR PLOY?

Jonathan Markworth, *managing consultant with CompuCom Systems, Inc., discusses the benefits of a federated CMDB.*

Wouldn't it be great to have a single, all-knowing, all-powerful, and self-maintaining tool to manage every facet of information about the IT infrastructure? Wouldn't it be great to start over fresh — to replace your legacy enterprise management infrastructure with tools that do everything within a fully integrated platform? The reality is that most organizations have inherited (or deployed) dozens of applications, tools, utilities, data stores, hardware platforms, and management frameworks that perform one or more of the service management functions. Each of these has a data repository that provides information to support some critical function within your current environment. In the context of the CMDB, the data repositories related to these tools contain most of the critical CI attributes that can be used to identify the relationships between the CIs. The key question is then how do you leverage the investments you've already made to build an integrated data repository such as a CMDB?

One method is to export the CI identification, attribute, and relationship data from each of these data sources and combine it into a single database. Over an extended period of time, this method can be very difficult to maintain and only becomes more complicated as the number of data sources grows.

An alternative approach is to build a CMDB that consolidates identification, attribute, and relationship data, making it readily available to all the IT processes that need it and without requiring all of the data to be copied centrally. In this federated model, the CMDB holds CIs and a critical subset of data, but then links to other sources of related data such as service desk tickets, service level agreement definitions, or even management consoles. Deployed correctly, the federated model can facilitate the introduction of an enterprise CMDB that can span the whole IT organization, allow for a phased transition plan from legacy systems — if that is even necessary — and preserve the ability of the IT organization to continue day-to-day operations with fewer interruptions.

Excerpt from “The Potential of the CMDB — Driving a New IT Reality,” VIEWPOINT, Focus on: CMDB, published by BMC Software.

Most out-of-the-box CMDB solutions are built as the foundation to support the vendor's own IT environment autodiscovery solutions or IT service management solutions. The IT environment in most companies will already have a collection of heterogeneous discovery and/or service management tools. Therefore, do not limit your research to the vendors whose tools are already installed at your company.

Task 3. Define Solution Requirements

Now you need to refine and reduce the superset of functions you identified in your CMDB technology research to meet the specific functional requirements you documented in stage 2. Use the list of CMDB core functions as a framework to document the specific requirements for this project. You will use this list of solution requirements to evaluate existing in-house solutions and as the foundation of your vendor RFP.

Task 4. Assess Current In-House Technologies

Using technology you already have in house is always a consideration. Now that you've reviewed what is available and determined your specific requirements, the next step is to do

a gap analysis against what technology you have, so you can determine whether extending a current solution is an option.

How big is the gap between a modified in-house solution and the required functionalities? Have heavy investments already been made in the existing in-house-developed solution that can be extended to meet requirements? The answer largely depends on how extensive your CMDB requirements are. If your requirements don't include use of CI relationship data, you might be able to extend an existing asset management or service management solution to incorporate other repositories and gain some CMDB functionality.

Task 5. Determine Build or Buy Strategy

The decision to leverage existing solutions or buy a new solution will be influenced by requirements, existing solutions, resource availability and skills, and budget. Refer to your business case and the resource commitments made when the project was approved. Your assessment of solution alternatives should be guided by these constraints.

If you decide to use existing in-house solutions, present your decision to the PEB and proceed to step 17, "Plan the CMDB Population." Otherwise, begin your vendor search.

Task 6. Develop RFP from Detailed Research

Use the requirements you developed in task 3 to develop a detailed RFP. Your goal is to narrow the potential vendor list to solicit detailed project bids.

Work closely with any IT staff in your organization who regularly participate in tool selection, so you can benefit from their guidance and advice. Also, segregate your requirements lists into categories, such as essential, desirable, and expendable. If no tool meets your complete requirements list, you might need to reduce your solution scope. Use a categorized list, and start by eliminating the tools that do not meet the essential category.

Many sources can help you identify CMDB solution vendors. Some sources of information include:

- » **PinkVerify™** — This service provided by Pink Elephant compares products with the ITIL processes. Products that are compatible with ITIL requirements obtain PinkVerify status.
www.pinkelephant.com
- » **Research services** — Many research firms provide analyst perspectives that may be helpful when choosing a CMDB vendor.
- » **Conferences and seminars** — These are good opportunities to meet potential suppliers and discuss and learn about their products.
- » **IT Infrastructure Management Association (ITIM)** — The ITIM is a dynamic organization, and its Web site has a section just for tools and technologies as well as discussion groups.
www.itimassociation.com
- » **Industry analysts** — Industry analysts who cover IT have expertise writing solution and vendor briefs that focus on IT service management and CMDB solutions.
- » **Vendors** — Vendors usually agree to conduct on-site presentations that can expand your understanding of emerging best practices and solutions.

Stage 3. Select CMDB Solution and Tools

Use Kepner-Tregoe decision analysis to first identify the “must-have” criteria for the tool, then to determine the “would-like” criteria. Then weight the would-like criteria in order of importance (from 10 down to 1). When you write the RFP, include questions that will identify whether — and how well — a tool meets all must-have and would-like criteria.

When the RFP responses are received, at least two people (for objectivity) need to grade the results to determine which tool best meets all criteria. Of course, reference checks, evaluations and other aspects of due diligence must be followed as well. This is the best way to end up with a tool that most closely meets the organization’s requirements. For more information about how to perform such an evaluation, see *The New Rational Manager*, by Charles H. Kepner and Benjamin B. Tregoe.

Task 7. Select Vendors for RFP

Select three to five vendors for a more in-depth solution evaluation, and send your RFP to them. Be sure to include questions about the pricing of products and solutions, and request completion by a particular deadline. Sign a nondisclosure agreement with these selected vendors, so you can gain more insight into their company direction, product strategy, and new functionalities in upcoming releases.

Task 8. Evaluate CMDB Solutions and Select Vendor

The final vendor and solution selection should include more than a technical evaluation. It helps to create a scoring and ranking system to evaluate the different CMDB tools you need. Rate each vendor, and select two or three for detailed evaluation. Set up meetings with these vendors to review the RFP and to see demos of the products.

Some best practices that will help your CMDB decision include:

- » Develop use cases and prioritize features and functions in terms of essential, desirable, and expendable.
- » Involve the right people, including the people who will design and build the solution, the people who will support it once it is in production, and the people from IT teams that will use the CMDB.
- » At a minimum, have vendors conduct detailed demos and allow plenty of time for questions and answers.
- » If possible, have the vendors install their software in a well-understood and controlled environment to conduct a proof of concept evaluation.
- » Be methodical in working through the use cases, and be thorough in recording the observed outcome.
- » Talk to reference customers and, if possible, visit them on-site. You will learn a lot about the use of the solution you might acquire, and you will gain insight into how easy the vendor is to work with after the purchase.
- » Devise a scoring system that allows evaluators to summarize their findings.

Figure 16.6 shows an example of a scoring system that considers solution fit as well as vendor viability.

CMDB Solution Evaluation Worksheet – Product “X”

| Rating Score | | | | | |
|---|--------|-----------------------------------|-----------------------------------|-----------------------------------|--------------------------|
| Criteria | Weight | Below Expectation (Score=0) | Meets Expectation (Score=1) | Above Expectation (Score=3) | Weighted Rating Score |
| 1.0 Product — Functionality | | | | | |
| 1.1 Visualization | 20% | | | | |
| 1.2 Report and query | | | | | |
| 1.3 CI data administration | | | | | |
| 1.4 Security | | | | | |
| 1.5 Data synchronization | | | | | |
| 1.6 Data reconciliation | | | | | |
| 1.7 CI data modeling | | | | | |
| 1.8 Integration | | | | | |
| 2.0 Product – Robustness | | | | | |
| 2.1 Scalability | 10% | | | | |
| 2.2 Performance | | | | | |
| 3.0 Product – Strategy | | | | | |
| 3.1 Research and development effort | 10% | | | | |
| 3.2 Product road map and vision | | | | | |
| 4.0 Product – User Appeal | | | | | |
| 4.1 GUI appeal | 5% | | | | |
| 4.2 Simplicity of workflow | | | | | |
| 5.0 Product – Market Penetration | | | | | |
| 5.1 Hype in the industry | 10% | | | | |
| 5.2 Number of organizations that purchased the product | | | | | |
| | | | | | |
| 6.0 Product – Pricing Structure | | | | | |
| 6.1 Initial purchase pricing | 15% | | | | |
| 6.2 Future purchase pricing options | | | | | |
| 6.3 Cost of ongoing maintenance | | | | | |
| 7.0 Vendor – Support | | | | | |
| 7.1 Availability of formal product training | 5% | | | | |
| 7.2 Product support coverage | | | | | |
| 7.3 Feedback from other users’ experiences | | | | | |
| 7.4 Availability of knowledgeable third-party resources | | | | | |
| 8.0 Vendor – Company Stability | | | | | |
| 8.1 Financial stability | 15% | | | | |
| 8.2 Company brand name | | | | | |
| 8.3 Alliance with other vendors | | | | | |
| 8.4 Number of years in business | | | | | |
| 8.5 Likelihood of merger or acquisition | | | | | |
| 9.0 Product / Vendor – Customer Reference | | | | | |
| 9.1 Feedback on product | 10% | | | | |
| 9.2 Feedback on vendor | | | | | |
| Total Weighted Score | | | | | |

Figure 16.6. Example of vendor scoring system

Stage 3. Select CMDB Solution and Tools

Unless your search reveals only one outstanding tool, you will need to apply other criteria to help you make a decision. Here are some suggestions, and you may also have other criteria you want to apply:

- » **Most matches** — The tool simply meets most of the requirements. However, never believe claims — always verify them.
- » **Cost** — Cost is obviously a factor, but remember that the cheapest solution is not always the best solution. Look at the total package, including renewal, maintenance, and licensing. Also consider the cost of customization and integration. Ultimately, the tool that has the best fit, out of the box, to must-have and would-like criteria will be the best value. Organizations that choose to purchase the least-expensive tool often discover that the manual work required to retrofit the tool brings the total cost of ownership to just as much, if not more than, the most-expensive tool.
- » **Renewal and maintenance** — You'll want to consider not only the cost, but also the deliverables that come with the renewal and maintenance contracts.
- » **Current supplier** — If you already have a good and reliable supplier, then this can be a good indicator for tool selection. However, make sure that you still carefully consider other vendor tools.
- » **Support** — A CMDB is the heart of service management, and if the CMDB fails, then fast-response support is essential. Make sure you are confident of the support provided by the supplier you select.
- » **Compatibility** — Although compatibility covers a number of items, be sure to consider federation, integration, and data sharing.
- » **Image of suppliers** — Your tool selection should be influenced by the answers to questions such as these: Is this supplier reputable? Is this a niche supplier? Does this supplier specialize in service management? What is the financial state of the supplier? What partners does the supplier have?
- » **Other products** — A supplier might have other products that match some of your other requirements lists.

Finally, begin the process of price negotiation with your first and second choices. Remember your budget. Revise your cost estimates based on details provided by the vendor.

Before moving on to the next task, you should have selected a tool that meets your criteria and other needs as closely as possible.

Task 9. Present Findings to Project Executive Board

Now, the purchase must be approved by the PEB. Plan to present a summary of the CMDB scope and objectives, and include the strengths, weaknesses, opportunities, and threats (SWOT) analysis evaluation scores for the top products and vendors. Also include a basic product deployment strategy and high-level implementation timeline, which you will expand in the next step.

If the selected solution exceeds your current budget, you may need to reduce requirements and scale back the project to fit. Or, you can select a phased approach and meet critical requirements now and additional requirements in a subsequent phase.

Summary

The key to success is to select the best possible CMDB to meet your needs and vendor criteria, within the budget that was approved when you presented your business case. Then add other tools to meet your requirements. In this step, you have selected a CMDB solution and the supporting technologies. Moving from requirements to a solution purchase is a big step. Congratulations. In the steps that follow, you will finalize the scope of your CMDB project, plan the implementation, and construct the CMDB.

In ITIL V2, although it was never explicitly stated, people made the assumption that the database was monolithic. By contrast, in ITIL V3, you have a CMS that includes one or more CMDBs in a federated model.

If you are considering implementing a CMS, then you should also be thinking about integration. Make sure that the CMDB you purchase has the ability to integrate with other data repositories. Then you will be able to implement a version of CMS that fits the model as defined by ITIL V3. What's more, you'll be able to integrate with other data repositories, such as other CMDBs.

Step 17

Plan the CMDB Population

Goals and Objectives

In this step, you will establish the exact plan for the initial population of CIs in your CMDB. You will consider the source of each CI and group the CIs into data sets you will enter sequentially. It is important to define business rules to reconcile duplicate data, both for initial population and ongoing CMDB maintenance. The planning at this stage will allow you to consider additional discovery and automation solutions you may need in step 18, “Select Tools to Automate CMDB Population.”

Overview

Whether the CMDB is new to the environment or an upgrade of an existing system, it will more than likely have interdependencies and connections with underlying systems that can have a significant impact on business continuity.

The actual population of data into the CMDB will be quite an undertaking and will require scoping and analysis beforehand. For a typical CMDB population project, you will need to do the following:

- » Create milestones, high-level project plans, and supporting repositories and procedures.
- » Schedule kickoff and weekly or biweekly meetings for progress updates.
- » Identify subprojects (one per data set) and create a list of requirements and goals for each. Identify interdependent development activity, determine development flow, and map to project schedule. Include:
 - Parallel development projects (user interface customizations, DSL population)
 - Linear development projects (discovery tools, reconciliation, etc.)

Stage 3. Select CMDB Solution and Tools

- » Assign a project lead or owner for each subproject who is responsible for reporting progress, escalations, and next steps.
- » Establish an open line of communication with all participants, external and internal (e-mail aliases, repository, alerts).
- » Create both a time and budgetary buffer for emergencies — minimum 10 percent.

Auto discovery is a wonderful way to populate and to maintain data, and you might be tempted to use a discovery tool to grab as much information as possible, as quickly as possible. A word of caution, however: a heavily populated CMDB does not mean that it's a good CMDB. Once you build and populate the CMDB, you will need to actively maintain it. With data that exceeds specific user requirements, the best-case scenario is that you will maintain unnecessary information, which is expensive and pointless. In the worst-case scenario, if you enter data that is not maintained, you risk users losing confidence in the quality and accuracy of CI data, which can quickly derail the adoption of the CMDB.

Design and deploy your CMDB to meet the CI requirements you identified in stage 2, “Define Requirements and Create IT Service Model Blueprint.” Start with your service model blueprint to identify CIs and group them into data sets. Identify the sources of the data for each data set, map the data sources to each CI, and plan the sequential order for entering the data sets into the CMDB. Finally, consider and begin to document the business rules needed to manage overlapping data brought into the CMDB.

In this step, you must concentrate on the final scope required for the CMDB and delivering reality — not just potential. You'll need to conduct careful analysis before you can finalize your plan. If you follow a thorough plan when building your CMDB, you will learn as you go and be able to use that knowledge to improve the actual population of the CMDB.

Task-by-Task Guide

Figure 17.1 shows the tasks for planning the population of your CMDB.

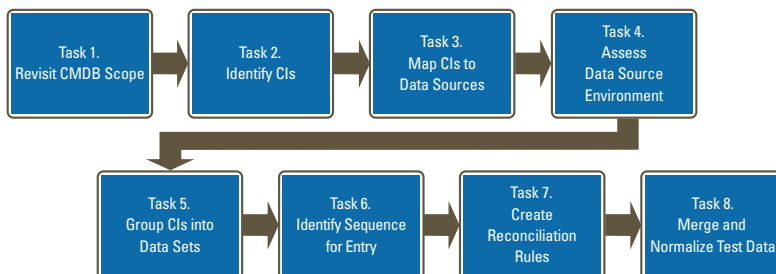


Figure 17.1. Planning the population of your CMDB

Task 1. Revisit CMDB Scope

Now for the reality check. Can you actually deliver the scope of CI data that you identified in stage 2, “Define Requirements and Create IT Service Model Blueprint”? Do you have the resources you need to maintain the solution and the CI data once implemented? These are important questions, because the CMDB solution you selected may easily scale beyond what is required and beyond what you can easily maintain.

As you plan the population of the CMDB, your mindset should be “less is more.” Walk before you run. Let the initial rollout of the CMDB prove itself before you expand the scope. You need to help temper the enthusiasm of both the CMDB team and the users of the CMDB data to avoid scope creep that can derail the implementation and user acceptance of the new solution.

Keep in mind several considerations that can help you maintain a focus on key requirements and respond to critical limitations:

- » **Cost** — Everyone must live with the realities of business life, including budgets and costs. Therefore, prioritize key requirements only within the scope of your CMDB project budget. If new ideas appear, they should be considered only if they come with new budget.
- » **Time** — You may need to complete your CMDB by a given time so your organization can meet a business requirement, such as Sarbanes-Oxley compliance, or support a new process for a key stakeholder. When you plan the sequence of CI data entry (described later in this step), be sure to prioritize CIs that have definite time limits.
- » **Practicality** — If you do not have the resources to build and maintain a CMDB to meet documented requirements, then you might need to reduce the scope so you can work within your practical limits. You might consider a two-phase implementation and move noncritical requirements into the second phase.
- » **Externally imposed priorities** — Some business practices, such as governance, data protection, and freedom of information, might drive your prioritization. You can reduce the impact of externally imposed priorities by staging your plan for CMDB population. If you were not given enough time from the outset to beat the business deadlines, then you can try to use this limitation as a lever for more resources. However, be aware that the later you apply for more resources, the more it will look like you were not able to adequately manage your project.
- » **Ownership** — Sometimes the owners of IT assets are not part of the IT organization. If the asset owners decide not to participate in the CMDB, their decision can severely restrict the CI content. You might offer to restrict access to certain portions of the CMDB to address their concerns.
- » **Geography** — The boundary of geographic location can limit the creation of a corporate CMDB. Determine whether the barrier is geographical or political, such as “Not another crazy scheme from the head office.” The best way to avoid geographical opposition is to involve all locations as soon as possible in the planning of your CMDB.
- » **Organizational structure** — Many organizations divide IT into distinct, autonomous divisions where ownership can limit the scope of the CMDB. For example, telecommunications could be a separate group that refuses to participate in the building of a CMDB that resides outside of its control. Involve other IT groups, but make sure that everyone understands who the project leader is.

Stage 3. Select CMDB Solution and Tools

As you plan your CMDB population, the more you can focus on the needs of key stakeholders and align with business objectives, the smoother your project will be. You will need to communicate to all interested parties the factors that affect, or could affect, the previously approved scope of your CMDB. Gather all of their replies and discuss further if required.

Task 2. Identify CIs

Use the IT service model blueprint that you created in step 15, “Design IT Service Model Blueprint,” as the basis for creating a list of CIs and the related attribute and relationship data required to populate the CMDB. The detail in this list should be at the data field level, so that you can identify and map the source or multiple sources of that specific data.

For example, if you list the attribute and relationship data for a particular server CI, you may find that there are existing data sources for inventory data and a discovery or network management tool that provides some attributes. But other data fields don’t have existing sources. You will use this gap in step 18, “Select Tools to Automate CMDB Population,” to help identify additional discovery tools needed to populate and maintain that data field. Alternatively, you may decide that that field will be populated and updated manually, which will require careful consideration of the data owner and supporting processes in step 21, “Create CI Lifecycle Management Processes.”

At this point, stay focused on the requirements that meet the project objectives. In steps 11 through 14, you defined touch points with other processes and specific CI requirements. Focus your CMDB CI data list on CI requirements that directly affect the key stakeholders’ processes.

Task 3. Map CIs to Data Sources

Now take your list of CIs and map each CI to the data source for the required information. A simple spreadsheet such as the one in figure 17.2 may suffice. Larger organizations that have more complex data requirements may require additional worksheets or links based on CI categorization.

The ultimate goal of this exercise is to not only identify the source of data to populate CI information in the CMDB, but also to identify the touch points to processes and platforms that are affected during CMDB population. This exercise is also critical to define reconciliation rules and precedence in task 7, “Create Reconciliation Rules,” later in this step.

The mapping tool, as shown in figure 17.2, should contain each category of CI, the corresponding attributes, corresponding relationship data, and the source of the data.

| 1 | Foundation Discovery | | | | | | Configuration Discovery | | | |
|----|----------------------------|--------------------|------------------------|----------------------|----------------------|--------------------|-------------------------|-----------------|---------|-------|
| 2 | Fields | Windows w/ WMI | Solaris (SSH / telnet) | HP-UX (SSH / telnet) | Linux (SSH / telnet) | AIX (SSH / telnet) | Windows w/ WMI | Windows w/o WMI | Solaris | HP-UX |
| 3 | Category: Machine | | | | | | | | | |
| 4 | - Name | X | X | X | X | X | X | X | X | X |
| 5 | - Manufacturer | X | X | X | X | X | X | X | X | X |
| 6 | Dedicated (type of device) | X | X | X | X | X | | | | |
| 7 | - Description | X | X | X | X | X | X | X | X | X |
| 8 | - Mac / Machine ID | | | | | | X | X | X | X |
| 9 | - Model | X | X | X | X | X | X | X | X | X |
| 10 | - Domain | X | X | X | X | X | X | X | X | X |
| 11 | - Serial number | X | X | X | X | X | X | | X | X |
| 12 | - Scan time | X | X | X | X | X | X | X | X | X |
| 13 | - Number of processors | | | | | | X | | X | X |
| 14 | - Type | | | | | | X | X | | |
| 15 | - Insert time | | | | | | X | X | X | X |
| 16 | - WMI version | | | | | | X | | | |
| 17 | - Aliases | X | X | X | X | X | X | | X | |
| 18 | - Boot time | | | | | | X | X | X | X |
| 19 | - Memory | In Memory category | In Memory category | In Memory category | In Memory category | In Memory category | | | X | X |
| 20 | DHCP Use | X | X | X | X | X | | | | |
| 21 | Category: OS | | | | | | | | | |
| 22 | | | | | | | | | | |
| 23 | - Manufacturer | X | X | X | X | X | X | X | X | X |
| 24 | - Product | X | | | | | X | X | X | X |
| 25 | - Version | X | X | X | X | X | X | X | X | X |

Figure 17.2. Example of a CI mapping blueprint between CIs and data sources

You might find several overlapping sources, especially at the inventory layer of CI attribute data. This data usually includes a description of the unique physical properties and location of a CI — for example, model number, serial number, location, and owner. This information might be stored in multiple places that may be sources for the CI population and ongoing maintenance.

Multiple asset and inventory data sources might include:

- » Audit (inventory or discovery databases; agentless and agent-based)
- » Asset management system
- » Procurement system and license management
- » Accounts payable (purchased versus received)
- » Contracts system
- » Change management system
- » Other financial applications and systems

Task 4. Assess Data Source Environment

To ensure data quality, you should assess the environment not only for the CMDB, connecting tools, and technologies, but also for each data source. The phrase “garbage in, garbage out” applies here. The success of the CMDB project may depend on upgrading current systems or infrastructure to accommodate increased network traffic and data load, and also to ensure the quality of the data from a data source.

Stage 3. Select CMDB Solution and Tools

As you plan your CMDB population, ask yourself these questions about the quality of the mapped data sources:

- » How and where is that information stored today — database, spreadsheet, word processing document?
- » Are audit tools (discovery), software distribution, configuration management, or procurement systems in place to electronically track and store that information? Or is the data collected and updated manually?
- » Are the systems based on open standards or proprietary technologies?
- » What standardized tools are available from the vendors of those systems? From the CMDB vendor?
- » What is the physical location of the data sets that need to be integrated?
- » Does the communication pathway between the CMDB and data set need to be bidirectional or unidirectional?

You should also consider performance-related questions about the mapped data sources:

- » What is the current performance, capacity, and reliability of the data source?
- » What is the current activity level on the system?
 - Number of active users
 - Other tools currently integrated into it
 - Schedules for backup, antivirus scans, reporting, or data mining
 - Scan or queue schedules for audit tools (discovery or inventory)
 - Anything else that will affect performance
- » Is the hardware and environment sufficient to support the additional load and space required for data migration? For ongoing data synchronization? What is the calculated growth for a particular span of years?
- » Do any restrictions or special permissions need to be considered?
- » Do the vendors recommend best practices for integrating into their database (using hot backup versus production database to reduce load)?
- » What is the proximity of the data set to the CMDB and connecting technology?
- » Will physical topology limitations of the network (bandwidth, distance, etc.) have an impact?
- » What is the current version? Are there any plans to upgrade in the next six months or year? What are the benefits?
- » Are sufficient resources assigned to the current solution to assist with architecture, troubleshooting, and maintenance of the connected solution?

By answering some of these harder questions early, you can avoid post-rollout performance issues that may affect the success of the project. If the data source is deemed unreliable either for quality or performance reasons, go back to the key stakeholders and users that drove the requirement for that data, and determine if it is critical for their use. If not, drop that data requirement for the first phase of the CMDB plan. It is more important that the phase one rollout works and is adopted, than that it is extensive. If the data is considered critical, meet with the key stakeholders and owner of the data source to communicate the high priority, and jointly identify a resolution plan.

Task 5. Group CIs into Data Sets

Based on the CI data source mapping, determine logical groupings of data (data sets) for a sequenced CMDB population. Two basic approaches make sense. One is to create data sets based on CI type. Examples include telecommunications, workstation hardware, operating software, local servers, remote servers, business software, and workstation software. Another approach is to create data sets based on process touch points — for example, all CI data needed to enable an improved incident management process. Identify the data sets needed to include all required CI data in your list, remembering to stay within the scope of your CMDB. Be sure to distribute your list to interested parties for comments and suggestions.

Each data set will be loaded into the CMDB as a subproject. The amount of work required for each subproject will vary depending on the complexity of the integration, the amount of data that will be migrated, the skill sets available, the subproject's overall importance to the project, and the current state of the underlying interdependent system.

The high-level project plan and individual project requirements for each subproject should be mapped out to align any preplanning activities, additional resource requirements, and timelines with the overall implementation plan.

Task 6. Identify Sequence for Entry

Rather than entering as much information as possible into the CMDB as quickly as possible, plan the order in which you will put information into the CMDB based on the data sets you identified. This will help break up the overall CMDB population into smaller projects, and also will improve the accuracy and integrity of that data.

A smooth plan at this point will ensure a smooth passage later in the project. When you define the order in which you will add your CIs to the CMDB, keep in mind the following:

- » Preset constraints that cannot be changed, such as governance-dictated timing
- » Data that is easily discoverable (such data may allow you to populate a large amount of the CMDB quickly)
- » Information in existing databases that you can federate to supplement your CMDB
- » The complexity of components; you may need to wait until the end of the implementation, when your knowledge and experience have improved, before adding complex components to the CMDB

RECONCILING DATA FROM MULTIPLE DISCOVERY TOOLS

David Schrodel, CTO of Voyence, discusses consolidating information from disparate tools into your CMDB.

There are a multitude of discovery tools — each addressing a different kind of discovery need, each with its own data store, and each using its own naming conventions. The trick is to consolidate the appropriate data from those disparate tools into a single repository that provides consistent, accurate configuration information.

How do you do that? The answer is automatic reconciliation, a process that automatically resolves conflicts, duplications, and variations in data that is gathered from multiple sources. With automatic reconciliation, you can bring your multiple discovery tools together, consolidating data from these many disparate sources to create a single, composite instance of that data in your CMDB.

One obstacle to ensuring accuracy of CMDB data is the amount of overlap across discovery tools. Event management tools, for example, are great for initial detection. At the same time, these tools can discover high-level attributes such as device type and vendor, duplicating to some degree the configuration information captured by other tools.

This overlap requires that you create a sound discovery architecture that clearly defines which discovery application is responsible for each device type. The architecture must encompass the complete workflow of your discovery process — from device detection through configuration drill-down to topology discovery — and address such issues as scheduled versus on-demand scans and frequency of subsequent scans.

Your discovery architecture must also address what you mean by “real-time” configuration information; that is, how frequently you update the information. If you’re dynamically provisioning servers and the provisioning action involves topological changes, the degree of accuracy of the CMDB is critical. That means your discovery architecture must ensure that the network configuration management tool provides near-real-time updates to the CMDB.

However, other parts of the infrastructure, such as desktops and related peripherals, don’t change as often and aren’t as critical. Thus a more relaxed schedule of discovery and reconciliation is more appropriate. A weekly scan on a network connection may be sufficient for desktop environments.

Excerpt from “Reconciliation and the Power of One,” VIEWPOINT, Focus on: CMDB, published by BMC Software.

- » The location of information (Remote information and data that you cannot discover or federate can complicate your efforts to populate your CMDB. Leave this until you are closer to the end of the CMDB implementation.)

To add some of the CIs, you may need support from other IT groups so consider those groups’ availability as part of your planning.

Task 7. Create Reconciliation Rules

You will need to establish rules to govern how you will reconcile data elements that have multiple potential sources. Reconciliation is necessary to complete the initial CMDB population and on an ongoing basis to maintain data quality. To have a successful reconciliation process, you must determine, record, and configure the reconciliation rules for any data that will be populated and updated in the CMDB automatically. These rules must meet the overall requirements of the project and ensure appropriate federation with external components.

You should carefully plan reconciliation among various sources. Be sure to consider these aspects of reconciliation:

- » What is the frequency of data updates from each data set?
- » When do you schedule each data update? The recommended best practice is to avoid scheduling the data feeds simultaneously, so you can reduce the impact to the system or network.
- » How often does the data need to be updated and reconciled to accommodate the requirements of the consumers of the data? (For example, you might need a software-daily, hardware-weekly schedule.)
- » Are there any blackout periods (critical hours of operation, backup activity)?
- » Will you have accommodations for emergency reconciliation to override rules and schedules if required by the business?

There are different approaches and capabilities for data reconciliation, but all revolve around three important activities — identifying data, comparing data, and merging data — that work together to ensure consistency and accuracy across the CI data stored within a CMDB. These activities provide the means to organize data coming from multiple sources, create rules for how that data should be brought together, and determine how to reconcile incoming data with information stored in the CMDB. It is important to note that these three activities do not represent all of the functionality that may be part of your reconciliation process, but they do represent the more critical activities required for reconciling CMDB data. By understanding each of these activities and their roles in managing CI data, you can determine how to build a CMDB that you can successfully manage and maintain in a constantly changing and dynamic IT environment.

Identify CIs. When more than one data source provides CI information to the CMDB, these tools often will find identical CI data for a specific attribute. The initial step in reconciliation, therefore, is to identify and match duplicate instances across the incoming data sets.

A recommended approach is to create a unique CI identifier field associated with each unique CI. The identification activity lets you compare multiple data sets and, on the basis of unique identifiers, determine whether information contained in these data sets refers to the same CI. Not only is this step critical to identifying data across multiple data sources, it is also key to identifying similarities between discovered data and the data that is stored within the CMDB.

A unique identifier is CI information that distinguishes one CI from another in an IT environment. A unique identifier can be a host name, serial number, or other attributes that will likely remain constant, regardless of whatever CI changes occur. When there is not a single, unique attribute for CI data, a unique identifier can also be a list of attributes that, when combined, create a unique ID. For example, if a host name is not always unique and a serial number is not always unique, you can combine the two attributes to create a unique ID for a given CI.

Compare CIs. In simple terms, the compare activity creates a list of differences (deltas) between two data sets, such as the CMDB data and incoming data from various data sources. Consider this scenario: The compare activity just found a delta between a discovered data set and the CMDB data. What do you do about it? The compare activity is tremendously helpful in understanding

Stage 3. Select CMDB Solution and Tools

the deltas between multiple data sets, but without some type of business logic and workflow paradigm, the process of “doing something about it” would be almost exclusively manual. In other words, the compare activity would list the differences between data sets and then pass that information to someone in IT. The IT person would then determine whether or not to do something about it — not really the ideal way to manage a dynamic IT environment.

To achieve automation, the compare activity must be able to leverage defined business rules that tell the reconciliation process what to do when differences are found between data sets. If a difference is found, should someone be notified? Should precedence be given to a specific data source or the existing CMDB data? The IT processes that leverage CMDB data (such as change management) will answer most of these questions, but it is critical that reconciliation provides the means to automate the decision-making process.

The compare activity should provide workflow capabilities that allow you to create *If, If Else*, and *Then* scenarios based on your way of doing business. For example, IF a discovery source records an amount of memory for a server different from the memory recorded in the CMDB, THEN the compare activity workflow should determine if there was a planned change associated with the server or ELSE notify the change manager that an unplanned change may have occurred. (See figure 17.3.)

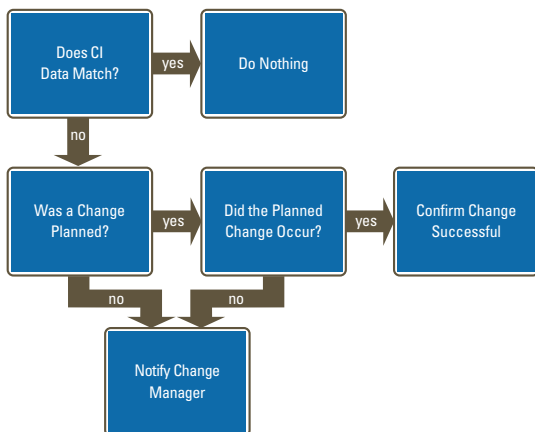


Figure 17.3. Workflow capabilities in the compare activity

Obviously, this example is rather simple. You can create many different scenarios based on unique IT environments and how IT processes leverage CMDB information. The important point to remember is that you cannot simply update the CMDB using information obtained from a discovery source, because you will not know whether unplanned or planned changes occurred. The compare activity and its associated workflow capabilities are essential for keeping CMDB data accurate while enabling IT processes, such as change management, to act on delta information found during the reconciliation process.

Task 8. Merge and Normalize Test Data

You should create a merged and normalized set of test data to complete the planning of your CMDB population. Merging is considered the final step in the reconciliation process. You have identified CIs across data sets, compared that data to information in the CMDB, and you now need to determine how and when to bring this information into a single, reconciled data set that can update the CMDB. You will need to accept or discard specific data elements from various sources based on reconciliation rules.

The merge activity is also essential when you have mapped two or more data sources to a single CI data element. Each data source will likely have areas of strength and weakness compared with the others. By creating rules that govern the merge activity, you can create one set of CI information that contains the best of all discovered data. You can break the role of the merge activity into two parts:

1. Merge data sets from various sources into a single set of CI data.
2. After initial population, merge data from the reconciled data set with information already stored within the CMDB.

When you bring together data from multiple data sources into a single, reconciled data set, you can complete the compare and merge activities against only two sets of data: the CMDB data set and a single, reconciled data set for discovered data. Simplifying data from multiple sources into a single data set creates a more approachable means to manage differences that may be found between the reconciled data set and the CMDB. It also reduces the amount of processing the CMDB is required to perform and limits the number of compare and merge activities that must occur against the CMDB.

Precedence rules for merging data. In instances where multiple tools discover information about the same CIs, the merge activity must be able to determine what data source takes precedence and is considered the source of record. Precedence rules allow you to assign a weight to individual data sources, giving one data source priority over another. Maybe you know that information captured from one data source will always be more reliable than information from another data source, so you want the first data source to have precedence — then the two data sources will always capture information about the same CIs.

Or, you may have data sources that provide different information about CIs and complement each other on the coverage of a CI's attributes, so you want to be able to choose which attributes will be used from which data source. The result is a reconciled set of data that represents the most accurate and comprehensive information available for CIs across all discovery sources.

Normalization. Regardless of where the data originates, the data from different data sets needs to be represented in the same way before reconciliation activities can occur. In other words, the data needs to be normalized into a common format that conforms to how the CMDB stores CI information. Data normalization enhances how manageable the merge and compare activities will be. Without a form of normalization, it becomes nearly impossible to compare or merge data that may be represented in different formats and value.

Stage 3. Select CMDB Solution and Tools

For example, consider how a library card catalog represents data about books. It normalizes data within the catalog by key attributes such as author and title. Searches based on these attributes are standardized across all books in the library. Imagine if the card catalog allowed readers to search for the author Ernest Hemingway using multiple formats such as “E. Hemingway,” “Ernest Hemingway,” or “Hemingway, Ernest” — but provided no way to understand that all formats refer to the same author. This example can extend to include the different ways publishers might define the “author” attribute for their books. Without a standardized, single format for “author” to normalize the book data within the card catalog, imagine how difficult it would be for readers to search for a certain author. They would need to know exactly how each publisher defined its “author” attribute to understand where to search for books.

The same holds true for multiple data sets and reconciliation. Across CIs, if information is not stored in a single, consistent format, then trying to reconcile data from multiple data sources, with multiple formats, becomes difficult and unmanageable. By bringing discovery data into a common format, you can dramatically simplify the accuracy and effectiveness of the reconciliation process in keeping a CMDB up-to-date.

Creating a merged and normalized set of test data as the final task in planning the CMDB population will force you to work through all the tasks up to this point. This task will also help you identify data collection gaps that will be addressed in step 18, “Select Tools to Automate CMDB Population.”

Summary

You may be tempted to rush through the process of populating a CMDB, because you can easily discover or federate much of the data. Avoid this temptation, because populating too much data will result in confusion in the long run. Plan carefully, and methodically work through the tasks outlined in this step before you build and populate your CMDB.

Step 18

Select Tools to Automate CMDB Population

Goals and Objectives

The goal of this step is to select any additional technology needed to automate the population of the CMDB and the ongoing maintenance of CI data (discussed in future steps). You will start by assessing the CI data and mapped data sources from step 17, “Plan the CMDB Population.” After pinpointing the technology gaps in your ability to automatically transfer that data to the CMDB, you will identify and procure the technology required.

Overview

You’ve spent a great deal of time preparing to implement your CMDB. Despite all of your effort, however, your CMDB will be only as good as the integrity of the data present at a given snapshot in time. Now that you have planned your CMDB population, it is worth reviewing opportunities to automate data management. Automating the initial population and ongoing upkeep of the CMDB will help you reduce risks (regulatory, external, internal), reduce initial and ongoing CMDB costs (people, technology, resources), and enhance overall CMDB and CI data integrity.

The selection of automation tools is a critical subproject within the overall program of implementing a CMDB. Not every data set within your environment will require an automation tool. Conduct a cost-benefit analysis to determine where it makes sense to automate. The impact of the cost-benefit analysis can be significant. For example, you may realize considerable cost savings by identifying areas that don’t need to be automated. Consider the following factors when undertaking your cost-benefit analysis:

- » Underlying systems from which you will pull the data
- » Performance of the CMDB
- » Timeliness and accuracy of information in the CMDB
- » Processes for reconciling information in dependent systems

Stage 3. Select CMDB Solution and Tools

The rollout of your CMDB will be most successful if you automate data collection and CMDB population for the categories of CIs in the service model blueprint that do not have a data source or that have an unreliable data source identified in step 17, “Plan the CMDB Population.” The following tasks guide you through the selection and justification of automation tools and technologies to populate and maintain your CMDB.

Task-by-Task Guide

Figure 18.1 shows the tasks for selecting tools to automate the population of your CMDB.

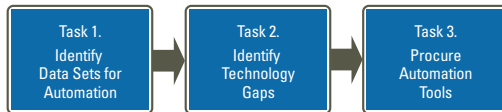


Figure 18.1. Tasks for selecting tools to automate population of CMDB

Task 1. Identify Data Sets for Automation

You will not need to automate every data set that consists of a collection of CI data elements in the CMDB. Only those data sets that have high time, risk, and volume factors will justify the cost. Although automation seems obvious in some cases, you should clearly define all the underlying benefits, requirements, costs, and success factors for automation. For example:

Recommendation

- » What data sets are you recommending for automation?
- » What data sets are you recommending for manual update?
- » What data sets are you recommending not to include in the CMDB, based upon this analysis?

Justification

- » Why are you recommending a given data set for automation?
- » What are the cost savings and driving factors per year over the next three to four years? Include high-level financial roll-up factoring in all costs from requirements gathering and environment assessment.
- » What are the overall project budget and resource requirements?

Success Criteria

- » Critical data sets to be automated
- » Environment factors that must be addressed (architecture, technology, location)

- » Required resources from each data set:
 - Business and technical
 - Consultants (include project management and technical skills)
 - Equipment
 - Time

Risks

- » Resource risks, including attrition and technology gaps
- » Time constraints, including corporate, divisional, or regulatory deadlines
- » Budgetary, including alternative solutions, phased approach, what will happen if you don't get the funding needed

Task 2. Identify Technology Gaps

For each data set you recommend to be automatically populated, you will need to identify technologies that can help automate data collection. You must review technology gaps not only for each data set but also for the data source environment. This critical task often is overlooked or left out but has the greatest impact on the proper function of the overall CMDB solution. By the end of this exercise, the project team should have defined several parameters.

Data. Identify the data source for each CI data element based on your mapping in step 17, "Plan the CMDB Population." Remember that just because you can collect inventory information on every application, patch, driver, and more does not mean you need to report on it in your CMDB. Try to eliminate non-essential data, such as specific drivers, applications that are bundled into the operating system, etc. Also, focus on the requirements of the consumers of the data.

Standard technologies. Create a list of standard technologies to be used across data sets and any gaps that may prevent integration. You will want to eliminate the use of proprietary technology where possible to reduce the costs and skills required. As you create this list, leverage your findings from step 16, "Select CMDB Solution." Do the selected technologies have common elements, connectors, or aspects to enable CMDB integration? Also ask yourself: Does the CMDB vendor selected offer enabling connectors, hooks, or out-of-the box integrations with any of the data sets?

A FEW THINGS TO CONSIDER

Val Sanford provides guidance on selecting a data collection solution.

With so many potential sources of information in many different formats, a tools audit can help minimize network complexity, thereby reducing the number of inputs into your CMDB. Simplifying your network infrastructure also has the benefits of improving efficiency, reducing costs, and ensuring the day-to-day stability of the disparate systems in your heterogeneous environment.

Questions to ask about tools in the data collection stage include the following:

- > What data is available for use by other systems?
- > What methods are available for extracting that data?
- > Are there license restrictions on data extraction that need to be addressed?
- > What teams are using the tool today and how do they use it?
- > Will configuring the tool to send data to the CMDB affect existing processes?

Even with the most efficient processes, you need to think about the possibility of losing some data when using tools to populate the CMDB. Questions to ask about handling data loss between tools and the CMDB include the following:

- > What level of data loss is acceptable?
- > What actions can we take to deal with data loss?
- > What impact will data loss have on the applications that rely on that data?
- > Will processes need to change to accommodate data loss?

You may need additional data collection tools to augment what you currently have. Once stakeholders in your organization have agreed to data selection

requirements and collection methodologies, the next logical step is to perform a build-or-buy evaluation. In some cases, organizations may decide to create a data collection solution in-house. However, there are off-the-shelf solutions available that may meet your needs. In either case, involve your vendors early in the process and set the expectation that they must work together. Heterogeneous environments are more efficient and effective when vendors cooperate to deliver a system to you, the customer.

When considering which tools to buy or build, include both practical and strategic needs. Think about the rate of change in your infrastructure — dynamic environments need more flexibility than static ones. The size and variety of your infrastructure is also a factor. Be sure to think about both the short-term and future needs of the company.

Possible requirements for a data collection solution include the following:

- > Out-of-the-box support for the systems with which you need to integrate
- > Bidirectional, multidirectional, or unidirectional integration between tools
- > Support for both real-time and batch collection
- > Use of existing standards
- > ISO or ITIL certification
- > Rapid deployment, configuration, and optimization to ensure quick ROI
- > Automated auditing to help meet compliance efforts
- > Extendibility and low service costs
- > Automated data scrubbing to reconcile data based on your Data Resolution Protocol

Excerpt from “Mission Possible: Gathering the Right Data in Your CMDB to Optimize IT Services,” VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Performance requirements. Define the minimum performance requirements for each data set and for the CMDB overall. This definition should include:

- » Expected time for first import of x records from y data set
- » Expected reconciliation time of records from each system after import
- » Expected return time of records for query, report, or dependent system
- » Maximum number of simultaneous connections
- » Overall capacity of the CMDB and data set — the maximum number of records (for example, must be able to grow to support 10 million CIs)

Platform, network, and environment requirements. Define the minimum platform, network, and environment requirements for integration, such as:

- » Operating systems UNIX, Microsoft Windows, other
- » Network bandwidth limitations
- » Maintenance practices in place at each system location

Functionality. Define the minimum functionality you'll need, such as reporting, debugging, and other features as defined by requirements in previous steps.

Security. Make sure the data sets and the overall system adhere to the minimum security and regulatory standards governing your organization.

Resources. Identify any additional skill sets or technical resources required to maintain or develop the solution between the selected technologies in step 16, "Select CMDB Solution," and the CMDB. Include both temporary resources, such as consultants and integration specialists, as well as permanent resources, such as a new position required or existing resources reassigned to the maintenance and upkeep of the system.

Environment improvements. Specify any prerequisite environment improvements required to implement the automation solution. These factors might include:

- » Minimum quantity of data required to support the CMDB
- » Architectural assessment per data set (new hardware, separate database, etc.)
- » Resources to implement improvements

Task 3. Procure Automation Tools

Once you have identified all the technology gaps, then it is time to procure additional resources and tools to complete the project. Refer to step 16, "Select CMDB Solution," specifically tasks 6, 7, and 8, for detailed guidance on creating an RFP and evaluating and selecting vendors. Whether you select a single vendor or several vendors will depend on the scope of your project and the underlying elements you identified.

Stage 3. Select CMDB Solution and Tools

Clearly define all requirements, deliverables, milestones, and consequences. Before your final confirmation of selection and prior to scheduling any activity with the vendor, ensure that all legal requirements, approvals, and documentation are in place.

Summary

Automation of data collection and population of CIs into the CMDB is critical to the successful rollout of your CMDB. As with any project, you must plan sufficiently so that you select the appropriate data sets for automation, the most suitable vendors, and the tasks for implementing automation tools. You can mitigate the costs and risks of automation and better fulfill the requirements by clearly communicating and defining the scope of the project with the business and technical representatives from each data set that will be automatically populated.

Step 19

Calculate Project ROI

Goals and Objectives

In this step, you will use data collected from the previous steps to create a financial analysis of the expected costs, benefits, and ROI for the project. You may not need to do this step if you were able to get funding approval using the estimated project costs and benefits you created in step 5, “Build a Business Case.” However, many organizations require a preliminary project business case and a more detailed financial analysis at time of purchase. Review your financial analysis with the PEB before presenting it to the project sponsor for approval. The analysis in this step will help guide the project implementation team as you move into stage 4, “Construct and Maintain Your CMDB.”

Overview

Developing a detailed financial summary of the expected costs and benefits of your CMDB project is part science and part art. The *science* is working with your finance organization to analyze anticipated hard costs and benefits of the project. Collaborate with the finance organization and use its recommended approach to build a picture of the financial value of the proposed project. The *art* is looking into the future to calculate the costs, hard benefits, and soft benefits of the project. Obviously, you don’t have a crystal ball. But at this point in the project, you have outlined the expected benefits, requested and received vendor quotes for the CMDB and supporting technologies, and created an implementation plan that includes resource requirements. Now you can use simple logic to convert the anticipated process improvements and gained efficiencies into financial terms. The finance organization can then run the numbers to calculate the financial viability of the project and to help determine if the project is worth pursuing.

Stage 3. Select CMDB Solution and Tools

Ultimately, you will use the financial analysis to obtain final approval on the CMDB project, if needed, and to provide budgetary parameters for the implementation team. To be successful in getting approval for the project, you also need to sell it to key stakeholders, using terms that matter to them. Financial folks care about the numbers. Operations managers are interested in the expected performance improvements and impact on resources. IT executives want to understand how the project aligns with current strategy and the overall impact on the business, as well as gain an understanding of the assumptions, risks, and resource impact on their organization.

Task-by-Task Guide

Figure 19.1 shows the tasks for calculating project ROI.

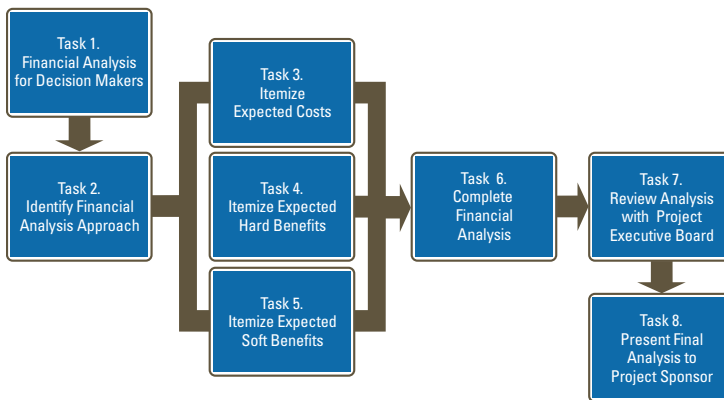


Figure 19.1. Tasks for calculating project ROI

Task 1. Financial Analysis for Decision Makers

To obtain budget approval for a CMDB project, especially with limited budget, you need to justify the project in financial terms. Should IT convert the data center to Linux or should it implement a CMDB? To answer that question, you need to conduct a financial analysis that is based on facts, reasonable assumptions, and logic. Financial analysis enables you to compare two or more alternatives while looking into the future. You will use the financial analysis to justify that implementing a CMDB is the best use of IT resources.

Your financial justification doesn't need to be perfectly predictive. No financial analysis method can forecast exactly how an implementation will roll out and how it will be used by an organization. However, the financial justification needs to make sense. If the tool, the data, and the assumptions about improvements are credible and "make sense," then it will be an effective part of a business case for a CMDB.

Task 2. Identify Financial Analysis Approach

You don't need to be a finance expert to create a financial summary of your proposed CMDB implementation. However, in selecting a financial analysis approach, you should seek — and use — the recommendation of someone in your finance organization who normally is involved in financial analysis of IT projects. You don't need to reinvent the wheel; use the approach that your financial analyst recommends. Meet with your financial analyst, outline the project, and ask what data is needed to conduct the analysis. At a minimum, you'll need to provide details about anticipated costs and benefits, as outlined in tasks 3, 4, and 5.

Some of the common approaches include:

Discounted cash flow — determines the present value of future cash flows by discounting them using the appropriate cost of capital. This is necessary because cash flows in different time periods cannot be directly compared since most people prefer money sooner rather than later (put simply: a dollar in your hand today is worth more than a dollar you may receive at some point in the future).

Risk adjusted ROI — a ratio of benefits divided by costs that provides a high and low range, based on differing sets of assumptions.

CASH FLOW MODELING BASICS

Financial decisions are based on cash flow analysis. Cash flow analysis uses projected costs and benefits to provide an estimate of the net financial impact of a potential solution. To construct such a projection, you must document not only all of the expected costs and benefits of the decision, but also the time period in which they occur. From that, various financial analysis techniques can be applied such as:

Return on investment (ROI) — presented as a ratio of benefits (return) divided by (on) costs (investments). Calculations may vary since there is no universal formula that is accepted by the Financial Accounting Standards Board (FASB). However, the typical approach includes using a ratio of net present value of costs and benefits.

Discounted cash flow — the basis for comparing different projects of different sizes covering different time periods, all discounted back to today's dollars.

Cost of capital — the rate assigned by your finance organization to discount cash flows (typically in the 8 to 15 percent range).

Payback period — an analysis that shows the number of months it takes to pay back the initial investment. Comparing only the payback of two projects has serious flaws, and therefore typically is not used as a decision tool.

Net present value (NPV) — the present value of costs minus the present value of benefits. This gives you the sense of the absolute size of the expected return based on discounted cash flow. The goal of your project is to have an NPV greater than zero.

Internal rate of return (IRR) — the interest rate at which the discounted cash flow yields a net present value of zero. It is best if your project has an IRR greater than the cost of capital. This metric is especially useful when comparing a variety of potential investments.

Total cost of ownership (TCO) — a financial estimate designed to help assess direct and indirect costs related to the purchase of any capital investment. A TCO assessment ideally offers a final statement reflecting not only the cost of purchase but all aspects in the further use and maintenance of the equipment, device, or system considered. This includes the costs of training support personnel and the users of the system, costs associated with failure or outage (planned and unplanned), diminished performance incidents (i.e., if users are kept waiting), costs of security breaches (in loss of reputation and recovery costs), costs of disaster preparedness and recovery, floor space, electricity, development expenses, testing infrastructure and expenses, quality assurance, incremental growth, decommissioning, and more.

Task 3. Itemize Expected Costs

You'll need to provide a detailed estimate of total project costs and in what time period they will occur. Use your vendor estimates and your implementation plan to gather this information. The estimate should include not only the initial cost of implementing the CMDB, but also the annual cost of running and maintaining the solution over time. Below are the typical costs you should capture:

- » **Software license** — includes the cost of licenses for the full stack of technology in the CMDB solution, from server operating system up through supporting-technology user licenses.
- » **Hardware** — consists of all costs related to hardware and facilities costs. You may be required to add overhead cost for such items as data-center rack space and network bandwidth.
- » **Annual maintenance** — comprised of the total cost of hardware and software maintenance based on a specific contracted period. This may also include the ongoing facilities and network usage costs.
- » **Professional services** — includes all costs associated with the team that is implementing and integrating the solution.
- » **Training** — includes the cost to train all of the various groups within IT that will be affected by the CMDB. Training is essential if you want users to adopt new solutions into their existing process flows.
- » **Administration** — includes all expenses for ongoing support and improvement of the system in terms of full-time equivalent of a system administrator's time. Your financial analyst can help convert people time into dollars.

An example of a completed cost portion of a cash flow analysis is shown in figure 19.2.

| Implementation and Maintenance Costs | | | | | | |
|--------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Estimated Costs | Year 0 | Year 1 | Year 2 | Year 3 | Total | NPV |
| License | \$2,500,000 | \$0 | \$0 | \$0 | \$2,500,000 | \$2,500,000 |
| Annual maintenance | \$0 | \$500,000 | \$500,000 | \$500,000 | \$1,500,000 | \$1,200,916 |
| Hardware, software, maintenance | \$0 | \$1,024,920 | \$1,024,920 | \$1,024,920 | \$3,074,760 | \$2,461,685 |
| Professional services | \$1,200,000 | \$0 | \$0 | \$0 | \$1,200,000 | \$1,200,000 |
| Training | \$0 | \$50,000 | \$0 | \$0 | \$50,000 | \$44,643 |
| Administrator | \$0 | \$150,000 | \$150,000 | \$150,000 | \$450,000 | \$360,275 |
| Total \$ | \$3,700,000 | \$1,724,920 | \$1,674,920 | \$1,674,920 | \$8,774,760 | \$7,767,518 |

Figure 19.2. Cost portion of a cash flow analysis

Task 4. Itemize Expected Hard Benefits

In this task, you will detail the logic of the expected value the organization will realize as a result of the CMDB project. People generally think of the CMDB as an enabler. That is, access to the CMDB results in an improved IT function or process. Therefore, use the metrics and improvement goals that you listed in step 4, “Review and Define Benefits,” to itemize hard benefits.

CMDB VALUE POINTS

CMDB users were asked to describe the value of a CMDB. The anecdotes below are just a few ways to capture the value points that a CMDB can bring to your organization:

- > **Bring higher visibility into the IT environment** — A leading provider of heating and cooling systems noted that it historically did not have a good understanding of the impact of unscheduled downtime on users. For example, when only 25 users were affected by a network problem, the IT department had to notify all 100 in the user base. This led to low customer satisfaction ratings. A CMDB enabled the company to understand the relationships between elements and know exactly who is impacted at any given time. The users now trust the communications coming from the IT department and perceive IT as an integral part of the business.
- > **Prioritize any system changes based on business objectives** — A large equipment manufacturer had to shut down all of its systems, and did not know what servers to bring back up first. A CMDB allowed the company to prioritize business-critical elements, ensuring reduced unscheduled downtime and potentially reducing the risk of revenue loss.
- > **Reduce costs of software licenses while ensuring overall license compliance for both user and server-based systems** — A semiconductor manufacturer was performing an audit and was amazed to find out how much the company kept paying to support and maintain servers that have been long gone. In reality, the company had inconsistently accounted for the retired assets. A CMDB helped the company to uncover this issue and reallocate the budget dollars to better support the existing infrastructure.
- > **Provide a deeper level of asset information and relationships to accelerate server consolidation** — A large financial service provider noted that companies in its industry commonly go through several acquisitions and mergers within a short time frame. It is a significant challenge (and sometimes an afterthought) to integrate all IT departments, and often they are left to work as silos. However, using a CMDB the company was able to efficiently integrate new acquisitions, thus saving money and creating a consistent image of IT as a business within the company.

Stage 3. Select CMDB Solution and Tools

Convert each metric that captures expected improvement from a performance number into a dollar figure. IT realizes hard benefits by reducing costs associated with:

- » People
- » Third-party services
- » Hardware
- » Software
- » Facilities

Take each performance improvement that you listed in step 4, “Review and Define Benefits,” and create a simple equation that links the benefit to a cost reduction. Convert performance improvements to cost reductions unless higher service levels improve revenue. Your financial analyst can then help put a dollar figure on your estimates.

Step 5, “Build a Business Case,” introduced the following example:

- » **Specific metric** — Change success rate, for example
- » **Definition of that metric** — Number of successful changes divided by number of planned changes
- » **Current measure** — For example, 80 percent
- » **Goal after implementation** — For example, 90 percent

Potential savings were calculated as follows:

- » Reduce failed changes from 20 percent to 10 percent per year
- » 3,000 total changes per year x 20 percent = 600 current failed changes per year
- » Each failed change requires 40 hours x \$90 per hour = \$3,600
- » Benefit = 300 failed changes avoided x 40 hours x \$90 = \$1.08 million annual savings

Total benefits should be listed as part of a cash flow table (see figure 19.4 in task 6, which shows a completed net cash flow summary).

Cumulative benefits over a time period can also be itemized for greater impact and to highlight the value of the project as shown below in figure 19.3.

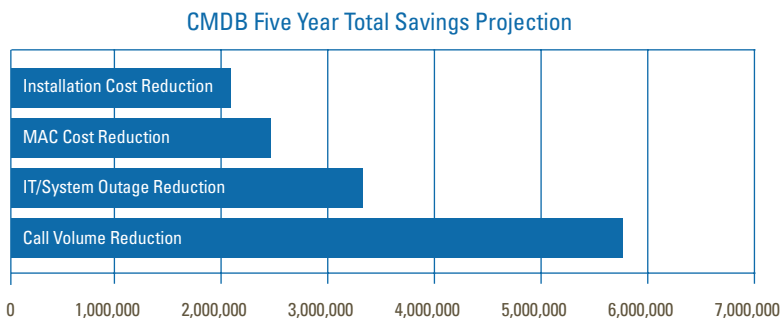


Figure 19.3. Sample cumulative benefits over a five-year time period

Task 5. Itemize Expected Soft Benefits

Hard benefits build a financial picture, but don't give you the complete CMDB story. There are likely to be other benefits that are difficult to quantify, but are nonetheless important to document as guidance for the implementation team. Some measures listed as project goals may not have a direct financial benefit. For a discussion of potential soft benefits, see task 1 in step 4, "Review and Define Benefits."

Also, consider the following examples:

Service desk — The CMDB can result in hard benefits by improving the efficiency and effectiveness of resolving incidents and problems. Also consider that the CMDB enables improvements, such as arming the service desk technician with information that is often unpleasant to collect from an aggravated user, which may increase support staff employee satisfaction and improve customer satisfaction. And, with better information available to the service desk team, you may be able to accomplish the same level of service with less skilled staff, resulting in a decrease in salary expenses.

Change management — This process is greatly enhanced with the CMDB, which enables better risk assessment, provides more information to evaluate success rate of past changes to similar CI types, and allows for an improved understanding of upstream and downstream dependencies with other infrastructure components. The result may well be that the business users are much more satisfied with the services provided by IT, but this is difficult to quantify as a hard benefit.

Continuity management — The CMDB becomes a great source of record for continuity management. Disaster recovery is greatly simplified by having the extensive and up-to-date record of what's in the IT environment, which can improve confidence across the organization. This is a definite benefit, but is not so easy to quantify.

Business impact and alignment — The CMDB enables a better understanding of CI dependency relationships. This understanding greatly simplifies the process of linking CIs to the business process or service that is dependent on the IT infrastructure. The benefits of tight IT-business alignment, such as improved responsiveness and better competitive advantage, are critical, but may be difficult to quantify as a hard benefit.

Task 6. Complete Financial Analysis

You have calculated the expected costs and benefits and identified when they will occur. Now, the financial analyst can put the two together and run an analysis using the preferred method recommended in task 2.

Stage 3. Select CMDB Solution and Tools

Typically, a financial analysis will result in two outputs:

- » **Net cash flow summary**, which combines annual costs and benefits and generates a simple picture of the financial impact of the project (see figure 19.4)
- » **Cumulative savings versus costs graph** (see figure 19.5)

| Net Cash Flow Summary | | | | | | | | |
|-----------------------|---------------|---------------|---------------|---------------|---------------|---------------|----------------|---------------|
| | Year 0 | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Total | NPV |
| Benefits | \$0 | \$3,123,863 | \$5,353,081 | \$6,813,013 | \$8,759,588 | \$9,732,875 | \$33,782,419 | \$22,995,545 |
| Costs | \$(3,600,000) | \$(1,707,236) | \$(1,674,920) | \$(1,674,920) | \$(1,674,920) | \$(1,674,920) | \$(12,006,916) | \$(9,666,565) |
| Net Cash Flow | \$(3,600,000) | \$1,416,627 | \$3,678,161 | \$5,138,093 | \$7,084,668 | \$8,057,955 | \$21,775,503 | \$13,328,980 |

Figure 19.4. Sample net cash flow summary

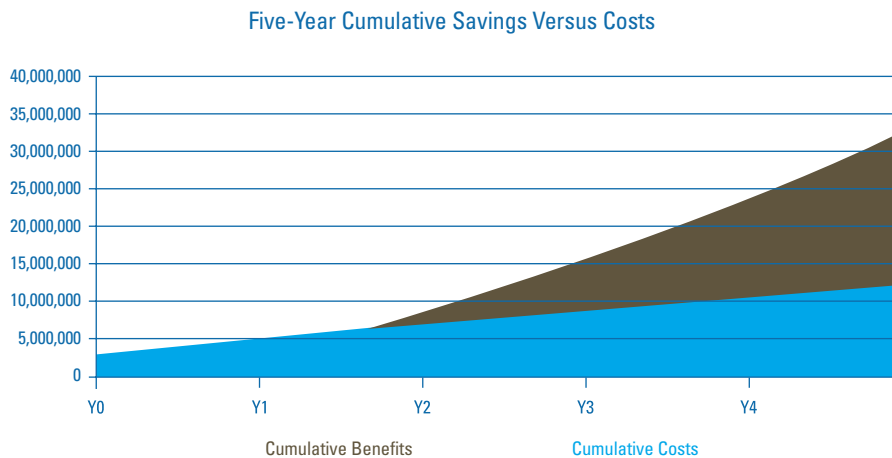


Figure 19.5. Sample cumulative savings versus costs graph

You should include the following in your comprehensive financial analysis:

- » Cash flow table showing costs and benefits over time, including calculations of NPV, IRR, or risk adjusted ROI as preferred by finance for making IT decisions
- » Itemized summary of benefits
- » Graph of cumulative savings versus costs

Have a two-way conversation with your financial analyst, and ask for ways to improve your analysis. Be sure to incorporate any feedback to reinforce the alignment of the finance organization with your project.

Task 7. Review Analysis with Project Executive Board

Now, it's time to share the financial analysis draft with key stakeholders:

- » **Finance** — Remember, don't talk to the finance organization about improved service levels and better alignment with the business. Present the cash flow table and the logic you used to convert projected measure improvements into hard savings.
- » **Process owners** — Check your assumptions with the process owners who will realize benefits as the result of your CMDB implementation. Show your measures and conversion into hard savings. Also share soft benefits, and make sure the process owners accept your logic. Be sure that the process owners have a clear understanding of the timeline for realizing the expected benefits.

Once you have ensured that your key stakeholders are on board, it's time to share your financial analysis with the PEB. Ask questions to get input and obtain buy-in. Ensure that the PEB is comfortable with the analysis and that it makes sense to them. If the PEB suggests changes to the analysis, be sure to make the changes before seeking final approval.

Task 8. Present Final Analysis to Project Sponsor

At this point, you want to present an executive summary of your financial analysis. You don't want to focus on the detailed financial calculations, but you do want to highlight key assumptions, risks, resources, and organizational impact. Make a point to discuss details that you know are particularly important to the project sponsor.

Summary

A good financial analysis of your CMDB project doesn't have to be complex. In the previous steps, you collected most of the information you need for your financial analysis. By pulling the key information together, and identifying the time period in which the costs and benefits will occur, you can create a simple financial picture of your CMDB project. Use the financial analysis approach recommended by your finance team, and rely on the finance folks to run the analysis. The result will be a compelling financial analysis that can be used to gain final purchase approval from key stakeholders, the PEB, and the CMDB project sponsor.



STAGE 4

Construct and Maintain Your CMDB

Stage 4. Construct and Maintain Your CMDB

During stage 4 of your CMDB project, you will construct your CMDB and build CI lifecycle management processes and CMDB supporting processes. You will populate your CMDB and ensure that the CMDB team is trained to ensure CMDB accuracy and integrity. You will also train the users of CMDB data to ensure adoption and that the CMDB achieves the expected benefits.

In stage 1, “Assemble the Project Team and Define the Project,” you collected information step by step to build a business case that was approved by the PEB and project sponsor. In stage 2, “Define Requirements and Create IT Service Model Blueprint,” you worked with key stakeholders to document and obtain approval on a broad range of solution requirements. In stage 3, “Select CMDB Solution and Tools,” you selected a CMDB and planned the population, including federation and automation.

In this stage, you will implement the CMDB solution and supporting technologies. In order to maintain CI data quality, you will implement a CI lifecycle approach to managing CIs, and build supporting processes to enable ongoing maintenance and upkeep. You will then populate your CMDB with data and move the system into production. Finally, you will train the CMDB team, as well as the users who will incorporate CI data into their work procedures, to ensure that the system is fully adopted and that expected benefits are realized.

Figure 1 shows the steps to construct and maintain your CMDB. Each step contains specific goals that you should meet before moving forward.

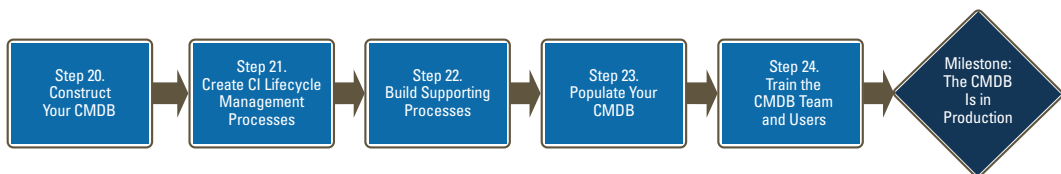


Figure 1. Steps for constructing and maintaining your CMDB

You may want to use the following as a checklist.

Step 20.

Construct Your CMDB

Goal: It's time to install your CMDB hardware and software, as well as all supporting technologies, to make your plans a reality. You will also need to determine your federation strategy so that you can identify and establish integration points to federated data stores.

Completed: ____

Step 21.

Create CI Lifecycle Management Processes

Goal: When you have completed this step you will have processes in place for adding new CIs to your CMDB, and updating CI data throughout the CI lifecycle. You will define a lifecycle plan for each CI type and then document how you will perform and control each part of the plan.

Completed: ____

Step 22.

Build Supporting Processes

Goal: You will document all of the processes required to support and manage the CMDB. Any flaws in current processes must be eliminated, and any new processes must be built and documented.

Completed: ____

Step 23.

Populate Your CMDB

Goal: In this step, you will fully populate the CIs, along with their associated attributes and relationships, into your CMDB. Populating the CMDB also includes federating data stores, gathering data, and reconciling data. At the end of this step, the CMDB will be ready for use.

Completed: ____

Stage 4. Construct and Maintain Your CMDB

Step 24.

Train the CMDB Team and Users

Goal: You will identify and provide all of the training and education that the configuration management team and CMDB users require. Training is critical for ongoing effective management and maintenance of the CMDB. To ensure adoption and achieve the expected improvements that justified the CMDB project, it is also essential to train users.

Completed: ____

Milestone: The CMDB Is in Production

The CMDB is built and CI management and supporting processes are in place. You have populated the CMDB with CI information and moved the CMDB into production. The CMDB team and CMDB users have been trained.

Milestone achieved: ____

Step 20

Construct Your CMDB

Goals and Objectives

In this step, you will install the hardware and software needed to build the CMDB so that you are ready to populate it with CI information. You will plan the production environment, install all supporting technologies, and integrate federated data stores.

Overview

You've worked diligently to plan and justify the CMDB. Now it's time to actually install the hardware and software to make your plans a reality. Begin by deciding on the approach for scaling the entire CMDB solution for the long term. Your scalability approach will help you establish your requirements and plan the production environment. You will also consider and install all supporting technologies. Finally, you will determine your federation strategy so you can identify and establish integration points to federated data stores.

Task-by-Task Guide

Figure 20.1 shows the tasks for constructing your CMDB.

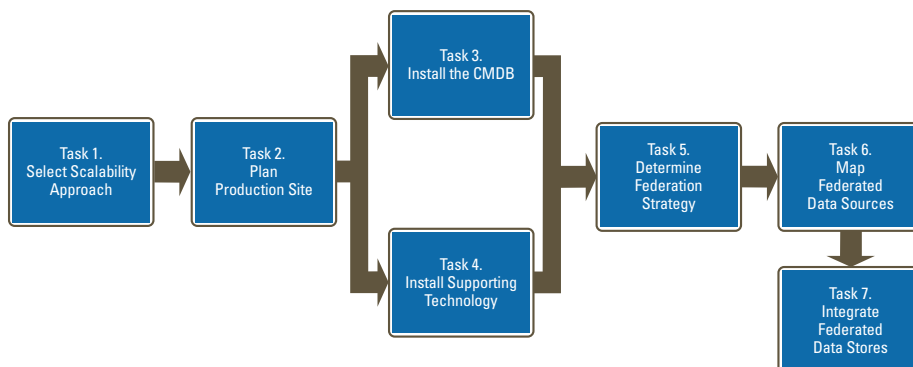


Figure 20.1. Tasks for constructing your CMDB

Task 1. Select Scalability Approach

You need to effectively plan for the scalability of your CMDB to ensure that it meets short-term and mid-term performance and usage requirements. Collaborate with a systems architect to identify how you will scale the CMDB as the number of CIs and overall usage increase. If you don't plan to scale, the performance and usability of your CMDB may suffer just as you are rolling out the solution to key users.

There are several basic approaches to scaling a CMDB implementation. Choose the approach that makes the most sense to you. Key factors in determining an approach include:

- » Number of anticipated CIs
- » Expected transaction volume
- » Number of CMDB users
- » Geographical distribution

The basic approaches expand the three tiers of the CMDB solution stack: 1) the client and mid-tier, which provide the user interface via the client and Web browser; 2) the server tier, which manages workflow, the data model, and input/output; and 3) the database tier, which manages data storage and retrieval. Consider the following basic scenarios, listed in order of increasing scalability.

Scenario 1. The server and database tier are installed on the same machine. This basic installation meets modest scalability requirements.

Scenario 2. As the number of CIs grows, the server tier and database tier can be installed on separate machines. This increases total system resources and overall performance. You also can add a disk array to the database tier to expand capacity.

Scenario 3. As more user and federated-system queries and reports are run, an archive and reporting server with a separate database can be added to the server tier. By putting the reporting server and database on a separate machine, you can minimize the impact of report-related transactions and database traffic and can further enhance performance.

Scenario 4. As user and CI count grows to the very large enterprise level, you can add multiple systems at the server level to create a server group or cluster. This functionality allows the systems to be used in a high-availability environment.

Scenario 5. As more functions tap into the CMDB and more browser-based users are added, you can introduce additional mid-tier servers to offload user-interface processes to a separate machine. You can use this approach with both single-machine and multimachine server-tier scenarios.

Scenario 6. Load balancers can be added to various multiserver and mid-tier scenarios to architect a fast-response, highly reliable system that is capable of supporting many users — regardless of how they access the system.

Scenario 7. You can deploy redundant and synchronized setups to scale across several geographies. In this scenario, all three tiers are replicated at another geographical location, and local traffic is managed remotely. Database tiers are synchronized to maintain identical data in this scenario.

Standard failover, redundancy, maintenance, security, and backup methods used in your data center should apply to any of these scenarios.

Task 2. Plan Production Site

Implement your CMDB solution and supporting technologies just as you would implement any other critical data-center infrastructure component. Place your shared data-center infrastructure components in secure, highly resilient, and closely monitored facilities. Processes and procedures for implementing and supporting any production system will also apply to the CMDB. Also develop a preproduction test environment that supports development and testing of precedence and merging activities.

Your scalability configuration will determine your specific combination of hardware and software. Be sure to take into account the physical and communications requirements when defining your production environment.

Consider the following requirements for the physical environment:

- » Rack space
- » Power
- » Cooling
- » Power cords
- » Network and serial physical cabling

Consider the following communications requirements:

- » Gigabit network
- » 10/100 Mbps network port
- » IP addresses
- » File Transfer Protocol (FTP) server access for moving data
- » External database access
- » Simple Network Management Protocol (SNMP) server access
- » Lightweight Directory Access Protocol (LDAP) services
- » Simple Mail Transfer Protocol (SMTP) server for sending

Be sure to also consider the prerequisites and version requirements of the various solution components. Some applications are limited to working with specific combinations of operating system, patch, application server, and database versions.

Stage 4. Construct and Maintain Your CMDB

Task 3. Install the CMDB

Installation of the CMDB includes the hardware, database, and application functionality. First, install and configure the server and database. Keep in mind the following:

- » Rack installation
- » Power connections
- » Data and management connections
- » Initial power-on and appliance configuration
- » Domain configuration
- » License installation
- » Admin login

Follow each vendor's instructions to complete the installation of the database, data model, and CMDB-related applications.

The CMDB-related systems should be integrated into normal data-center maintenance routines. When deploying a CMDB, make sure that:

- » All current documentation is available (electronically or physically)
- » Log files are monitored and regularly reviewed
- » Spare parts are available for maintenance (depending on your support contract)

Task 4. Install Supporting Technology

You may have considered a variety of supporting technologies when you gathered information about the CMDB in step 2, "Obtain CMDB Knowledge," and selected technologies in step 16, "Select CMDB Solution." Install any supporting technologies now. As with the core CMDB, consider the prerequisites and version requirements of the solution components.

Task 5. Determine Federation Strategy

After you have installed the CMDB and supporting technologies, the final step in constructing your CMDB is to determine and establish the points of integration needed to support your federation strategy.

As discussed in step 16, "Select CMDB Solution," federation is a method to provide access to data held in different locations. CMDB federation allows users from various IT functions to search and utilize CI data without needing to understand exactly where the data resides or the technology behind accessing the data.

As an IT organization evolves, it implements many different types of technologies and systems that store different types of data for different applications and users. By linking these disparate data sources, data federation helps provide a consistent way for users to access cross-functional data without requiring organizations to replace their specialized solutions.

THE FEDERATED DATA MODEL

Doug Mueller, *CTO of the BMC Software Service Management business unit*, discusses the federated data model.

A federated data model eliminates the need for several data migrations and application integrations to move change requests, help desk tickets, and other CI data into the CMDB. Rather, applications that use this data can continue to access it where it is currently stored. As a result, the CMDB becomes an enabler of information, and is not a bottleneck. The requests for related data are handled by other databases, so the CMDB does not need to accommodate all of the traffic related to processing that information. The load can be spread across multiple systems, reducing the potential for outages caused by changes. Although you could store your CMDB extended data in one place, you don't have to. The different types of data in this layer are not necessarily linked to or related to each other. The only thing they need to have in common is a link to or from the CMDB.

Excerpt from "The Federated Data Model," VIEWPOINT, Focus on: CMDB, published by BMC Software.

For any CMDB implementation to be successful, a CMDB must support some very basic concepts. Keep these concepts in mind as you determine your federation strategy. At the highest level, a CMDB must provide:

- » A means of allowing data to be published into the CMDB
- » A means for IT processes to consume CI and relationship data stored within the CMDB

There is no right or wrong approach to establishing federated links. You should evaluate each federation method and understand user requirements to determine which federated link option will work best for each element of CI data. To federate CMDB data elements, consider the following methods:

- » **Documentation within CI** — The simplest concept of federation is to include documentation in the CI details that describe how the CMDB user can access data stored in another system. An example of this method is:
 - When a user looks at procurement information related to a CI in the CMDB, data may indicate where to find the maintenance contract for that CI.
- » **Link to application** — The CMDB user can open up another application that stores information about the CI. Similarly, while viewing data in another application, the user clicks on a button that opens the CMDB application so the user can access CI data. The application is launched through a Web service, a run-process command, or other program execution options. Examples include:
 - While viewing a network router CI in the CMDB, the user clicks on a button or follows a URL that launches a network configuration management application to view more-detailed network configuration management information.
 - While looking at an incident record in the service desk application, the user clicks on a button to open the CMDB application to review change history in the CMDB.

In addition to these user-initiated federation methods, you can use more tightly integrated and automated methods. In a federated CMDB ecosystem, many different applications have direct

Stage 4. Construct and Maintain Your CMDB

relationships with CMDB data and frequently need to access that data as part of their standard processes. Various applications can access CMDB CI data, search on that data, and bring back results to their own applications or update the CMDB based on the results of an action or trigger that is programmed in the application workflow. More tightly integrated methods to federate CMDB data elements include the following:

- » **Embedded user interface** — By leveraging the user-interface technology provided through the CMDB, you can create user interfaces within the CMDB environment that link into a federated data store and have the same look and feel as the CMDB interface. Examples include:
 - While viewing a CI, a user can open an additional user interface that was developed with CMDB user-interface technology. This additional interface displays data stored in a different data repository.
 - While viewing an incident record, a user can open an additional user interface that was developed on the same platform as the service desk. This additional interface displays CI data from the CMDB.
- » **Direct integration** — An application can both obtain data from and populate data in the CMDB. Alternatively, the CMDB can both get data from and put data into another application data store. This level of programmatic interface is transparent to the user of the CMDB or the other application. The data transfer can be launched through a Web service, a run-process command, or other program execution options such as an application programming interface (API). Examples of direct integration include:
 - When the service desk user opens an incident record, the service desk application queries the CMDB to fill change history fields automatically.
 - When a change management application completes a change, the change history and approval information is automatically updated in the CMDB.

Task 6. Map Federated Data Sources

You need a complete mapping of data sources to prepare for integration. To identify specific integration points, use the list of CI and mapped data sources you developed in step 17, “Plan the CMDB Population.” Each integration point may require “planning” as a sub-project. Depending on the level of complexity, openness of APIs, and tools, integration can be as simple as an API call, or require integration with an enterprise integration bus. In some cases, you will need to make application code changes to packaged applications in the form of user exits or CI modifications, while other projects may be scheduled and transparent to users.

Task 7. Integrate Federated Data Stores

You have pinpointed and mapped the data elements that will be federated, and have identified the appropriate form of integration. Now, for those data elements that require integration, you need to integrate the federated data stores. To facilitate the ability to publish data into the CMDB or for the CMDB to publish data into federated data stores, consider the following technologies:

- » **Open API** — The API includes the programmatic calls for determining how to communicate with the CMDB. The API should provide the ability to publish data into the CMDB, search and pull information from the CMDB, and modify existing information stored within the CMDB. In many cases, an API is the most prominent way to integrate with a CMDB because

it lets applications and IT processes that need to interact with the CMDB do so through a standard open interface.

- » Data integration and transformation — Data transformation capabilities are critical in situations where you need to integrate data structures that represent data in different ways. Many applications and IT process tools store information in a proprietary format. When these data stores contain information that needs to be published into a CMDB, a data transformation tool provides the means to create mappings between the external data store and the CI data elements — allowing you to integrate into the CMDB data that may be structured completely differently. Data transformation tools typically should include mapping capabilities, as well as logic for determining when to exchange data between two or more data sources. In other words, the tools should provide the ability to define event triggers that allow you to say, “If data is modified or created in one data source, then automatically kick off a process to update that information in the data source to which it is integrating.”
- » Web services — Web services provide a standards-based approach for communicating and exchanging data between different data sources, applications, processes, etc. A CMDB should include the ability to define and publish Web services to allow third parties to create entries, modify existing entries, and search the CMDB for CI and relationship information. Web services typically are used for single transactions between the CMDB and other applications, as opposed to data transformation and APIs, which can support the bulk loading of data into the CMDB.

The three technologies described above provide the means for integrating any existing data source with the CMDB, allowing you to successfully populate it with information from discovery tools, existing asset or inventory repositories, and other applications and tools that provide CI data.

You should test this integration of data sources with the reconciliation tools and business logic you developed in step 17, “Plan the CMDB Population.” Data that comes into the CMDB from these integrated and federated data sources should be normalized and reconciled against data stored in the production CMDB data set.

Summary

Constructing the CMDB is an exciting step that brings all of your previous planning efforts together into a real solution. Recognize the combined CMDB hardware, software, and technologies as an important infrastructure component, and apply your data-center best practices to the implementation and support of the solution.

Step 21

Create CI Lifecycle Management Processes

Goals and Objectives

In this step, you will put processes in place for adding new CIs to your production CMDB and updating CI data throughout the CI lifecycle. You will define a lifecycle plan for each of the CI types and then document how you will perform and control each part of the plan.

Overview

To ensure that your CMDB stays accurate, you must manage a CI throughout its lifecycle and enter and update CI attributes into the CMDB as soon as they are available, or as soon as they change. Updating the CI attributes in a timely manner enables CMDB users to quickly note where a CI is in its lifecycle and gives users confidence in the accuracy of CMDB data.

Task-by-Task Guide

Figure 21.1 shows the tasks for creating a CI lifecycle management process.

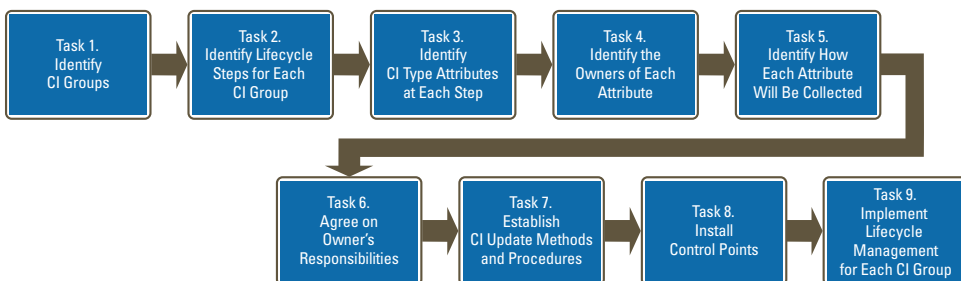


Figure 21.1. Tasks for creating a CI lifecycle management process

Task 1. Identify CI Groups

A CI group is, quite simply, a group of similar CIs that share the same lifecycle. A typical lifecycle for new equipment, for example, would be similar for many different types of equipment, such as workstations, laptops, printers, scanners, and servers. Each type of equipment may have different attributes and perform different actions, but the lifecycle process remains the same.

In this task, identify your CI types and then organize them into related CI groups. Figure 21.2 shows how types of CIs (workstations, laptops, printers, scanners, and servers) can be part of a distributed hardware CI group.

| Configuration Item Group | Configuration Item Types |
|--------------------------|--------------------------|
| Distributed hardware | Workstations |
| | Laptops |
| | Office Printers |
| | Scanners |
| | Servers |

Figure 21.2. CI types in a CI group

Before moving on to the next task, categorize and list all of your CI groups and their associated CI types. As the project progresses, you can add any missed CI types or groups as required.

Task 2. Identify Lifecycle Steps for Each CI Group

Now, create a lifecycle flow for each CI group. This task is not difficult, but the more CI groups you have, the longer it will take. The easiest way to create the lifecycle steps for your CI group is to plot all of the steps a CI group goes through from conception to retirement. Figure 21.3 illustrates a lifecycle for the distributed hardware CI group from task 1. You might create a similar lifecycle for software and other physical CIs, as well as lifecycles for service CIs, process CIs, and people CIs.



Figure 21.3. Lifecycle for a distributed hardware CI group

This flow reflects the lifecycle of typical hardware CIs:

- » Request for new equipment — You must identify the starting point of the lifecycle. In Figure 21.3, a request for new equipment — such as a service request to the service desk — has started the lifecycle.
- » Request authorized — The request must be authorized to ensure that it meets budget and other requirements, such as governance. Most organizations have a formal procedure for the authorization of requests for new technology.

- » Order placed — Here, you create a formal order and forward it to the supplier. The amount of detail in the order will vary depending on the complexity of the hardware you are ordering. For example, an order for a new mainframe is likely to be more complex and precise than an order for a scanner.
- » Equipment received — The equipment is received at your specified delivery location.
- » Equipment tested — You need to confirm that the equipment matches the order specification. If it does not meet the specification, then you'll need to take appropriate steps, such as contacting the vendor. If the equipment does meet the specifications, then you'll need to assemble it, if required, configure it, and test it to meet change management requirements.
- » Equipment installed — After you have tested the equipment, then you can install it. The equipment is now ready for use, and you might follow a formal hand-off procedure at this point.
- » Equipment maintained — Once the equipment is in full operation, you will make a wide variety of changes throughout its useful life, such as performing repairs, maintenance, and upgrades.
- » Equipment retired — When the equipment is no longer needed or is not performing to expectations, it is retired from use. Then the cycle may start again with a new piece of equipment.

Think of the lifecycle as a journey, and keep asking yourself: What is the next step? You may have many CI groups, although you should try to keep them to a minimum. Depending on your resources, one person could progress to task 3 after creating the first CI group lifecycle, while another person builds the next CI group lifecycle, and so on. This approach will accelerate your project delivery time considerably. As you complete tasks 3 through 8, be sure to document each task as well as any business rules you used for precedence when reconciling multiple sources.

Task 3. Identify CI Type Attributes at Each Step

In this task, you will identify the CI data to be collected or updated at each lifecycle step. This task entails combining the CI data you identified in step 17, “Plan the CMDB Population,” with the lifecycle flows you created in the previous task. The goal is to identify which data you can enter or update into the CMDB at each step of the lifecycle. By matching the data and lifecycle step, you can manage the lifecycle process in a structured manner. Figure 21.4 illustrates this structure.

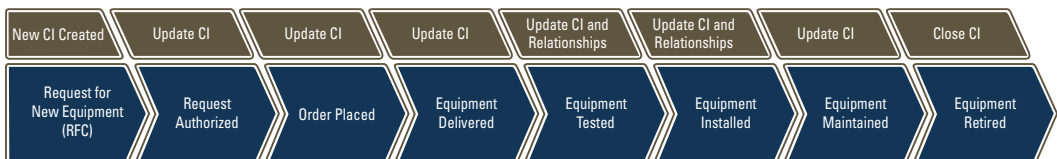


Figure 21.4. Match attributes to the CI lifecycle

Stage 4. Construct and Maintain Your CMDB

You can generate a new CI either when an RFC is raised or by an alternative notification method. In the example lifecycle shown in figure 21.4, the CI is created before it is authorized. Some organizations prefer to generate the CI after the request has been authorized. The order does not matter, but make sure the trigger point is clear and always obeyed. If you have a CI lifecycle and the information, CI attributes, and CI relationships from the earlier steps in this book, then it should be straightforward to match the attributes with the lifecycle. The following descriptions build on the example in task 2:

- » **Request for new equipment** — At this lifecycle step, the attributes describe when the CI was opened, who opened it, which equipment the request is for, and when the equipment is required.
- » **Request authorized** — These attributes describe who authorized (or rejected) the request, the date and time it was authorized, budget allocation, and any condition that might be applied, such as ownership.
- » **Order placed** — Ordering equipment requires a clear and detailed specification of the equipment, which means many valuable attributes will be available, including make, model, technical specifications, and features. If the order is placed and accepted electronically, then you can automatically populate the CI at the same time.
- » **Equipment delivered** — Attributes include a mixture of basic data (such as delivery date and time) and more critical data (such as serial number and technical specifications) that you can compare with the technical data supplied for the order.
- » **Equipment tested** — At this lifecycle step, you can start to add data such as ownership and relationships (see task 4). Before you can sign off on testing, you must check the CI (or CIs) to see that the information you have about the CI is complete and accurate.
- » **Equipment installed** — If you have collected the data diligently and updated the CI accordingly, then adding the installation date and time attribute data to the CI should be a quick formality.
- » **Equipment maintained** — The attributes that you will need to manage while the equipment is active depends upon any changes to the CI and the governance requirements to track those changes. The best method to follow is the ITIL approach: Any change of status to a CI must go through the change management process, where the attributes will be updated as part of the change.
- » **Equipment retired** — At this lifecycle step, the attributes are usually straightforward, describing the date and time of retirement, who authorized the retirement, and the date when you can remove the CI from the live CMDB.

After you have fully implemented your CMDB, you should continually review the lifecycle updates so that they are always placed as efficiently as possible in the flow. As a rule of thumb, the nearer the beginning of the lifecycle they are, the greater the benefit.

Task 4. Identify the Owners of Each Attribute

In the previous task, you matched each CI data element against the various steps of the lifecycle flow. In this task, you need to identify the data source and the owners of those attributes. Figure 21.5 builds upon the example in previous tasks, adding suggestions to show the potential data source for most of the attributes. IT is the first data owner because ITIL recommends

that the service desk handle service requests, which can include requests for new equipment. You'll need to identify the first time that data is available for updating the attributes and then identify that source.

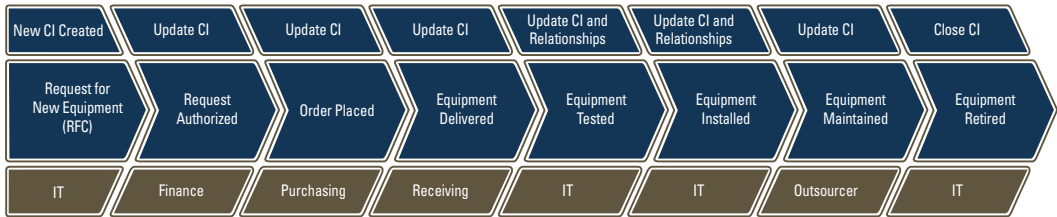


Figure 21.5. Location of attributes through the CI lifecycle

It is very important to establish ownership of the data contained in the CMDB. For example: Who will update the data once it is in the CMDB? Who has responsibility for the accuracy of the data? Who is responsible for the timeliness of entering the data? Is a security level associated with the data? You'll need to answer these questions for all of the data in the attributes. You should clarify who owns the data and who will manage the data. This is not such an issue for the data entered and managed by the CMDB management team, but it is essential for all other data sources. Contact the owners of all data and establish clear lines of responsibilities.

For federated data, you also should clarify who owns the databases that provide the source of federation. If a federated database is altered, then the federated data may no longer be available or accurate. Establish a process with the federated database owners, so they notify the CMDB management team about updates or changes to their databases.

Task 5. Identify How Each Attribute Will Be Collected

Now you must verify that the mechanisms are in place to update CI data. In step 17, "Plan the CMDB Population," you mapped CI data elements to data sources and determined precedence and business rules to deal with multiple sources. When you review the data sources with the data owners identified in the previous task, keep in mind that electronically generated data through methods such as discovery tools is far more likely to be accurate and punctual. To illustrate the concept of this task, consider one step from the example lifecycle:

- » **Order placed** — Figure 21.5 identifies the Purchasing organization as the location where the attributes will be collected. The key question, however, pertains to how those attributes will be collected. Is there an existing computer system to update so you can confirm that an order has been placed? If so, can you federate that information into the CMDB or copy it into the CMDB? Is it a manual system? If so, how quickly can you obtain a copy of the order confirmation so that you can enter it into your CMDB? Better still, can you automate the process and dynamically update your CMDB?

The key to this task is to identify how the data is managed at each location and how that data can get into the CMDB. Before proceeding to the next task, you'll need to identify the method used to collect the attributes for each lifecycle step and how the attributes will be entered, or federated, into the CMDB.

WHO OWNS THE DATA?

Tim Mason, founding director at TRM Associates, an information management consultancy firm, talks about the importance of establishing data ownership and clear measures.

Clear accountability for data goes hand in hand with the process element. We found that everyone wanted information from the CMDB, but no one wanted to pay to maintain it. People pointed to the configuration management team and said, “It’s their responsibility.” Configuration managers, however, cannot own the data because they have no way of knowing if the data is accurate. They can own the processes related to checking accuracy, but not the data itself.

With this in mind, we built a model of data ownership with clear measures that put the onus of maintaining CMDB data on the people who know the data best. This involved implementing a network of data owners and data managers. The data managers are members of the support group for each CI. We aligned their roles — for example, *operating system support* — to the data they must support on a CI — in this case, *operating system information*. Data managers are accountable to a data owner for data quality.

We incorporated data quality targets into contracts for third parties and established personal objectives for internal staff. The data owner acts as an escalation point to which configuration management can raise issues. We modeled all this in the CMDB. Now, when users click on a field, they can immediately see who owns the data.

Excerpt from “6 Tips for Effectively Managing the ‘People’ Factor,” VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Task 6. Agree on Owner’s Responsibilities

Locating the source of the data and the method used to create it does not entitle you to access that data. You will need to negotiate with the data owners for their permission to access and use their data. In some cases, you may have to alter what you collect, and then renegotiate. When you cannot get permission, you will need to explain to IT management why you cannot include those attributes in your CMDB. For example, you might not include attributes such as the names of authorizing officers because of data protection or security.

Task 7. Establish CI Update Methods and Procedures

Next, implement electronic methods to update CI data in the CMDB or establish procedures for manually collecting data for the CMDB. You will need to verify that those rules are valid as part of each CI lifecycle.

Although this task is simple in theory, it can require large amounts of resources. The most important thing to keep in mind is to automate as much as possible. Returning to an earlier example, if you could automate the process for the Purchasing organization to complete a form for an order, it would save them time and it could automatically update the CMDB. Most, if not all, data usually can be collected electronically. If some of the data resides permanently in other well-maintained databases, then consider federating the data rather than duplicating it.

Task 8. Install Control Points

Establishing methods and procedures to populate CI data is one thing, but ensuring that the data is correct is an altogether different challenge. In this task, you should install checks and controls to ensure the accuracy and integrity of the data as it is collected. Be aware that any data that cannot be checked or verified may distort the value of the accurate data in your CMDB.

Many organizations already use best practices, such as COBIT, to help them comply with the controls required by governance such as Sarbanes-Oxley. Those same principles could be used here. If you are not using a best practice for controls and checks, consider adopting one to ensure the accuracy and integrity of your CMDB. Visit www.isaca.org for more information.

Task 9. Implement Lifecycle Management for Each CI Group

After you document the lifecycle for each CI group, you can then release the lifecycle, using change management. Make sure that you publish all of the materials you have documented, so they are available to all key IT and business organizations.

Summary

This step can be long and tedious, but it is important. A well-planned and documented CI lifecycle management process for each CI group will ensure that processes are followed, will prevent the CI data update from becoming a bottleneck for change management and the other IT service management processes, will keep your CMDB accurate, and will help you easily maintain your CMDB. If you populate a CI throughout its lifecycle, you can achieve a greater degree of granularity than you can if you try to populate the CI in one sitting. Remember, you can federate data from other databases. Using discovery technologies, wherever you can, will help streamline the CI population and update processes. Finally, check with your compliance officer throughout the CI lifecycle to ensure that you are meeting governance, integrity, and security requirements.

Stage 4. Construct and Maintain Your CMDB

Figure 21.6 provides a final look at the example in this step.

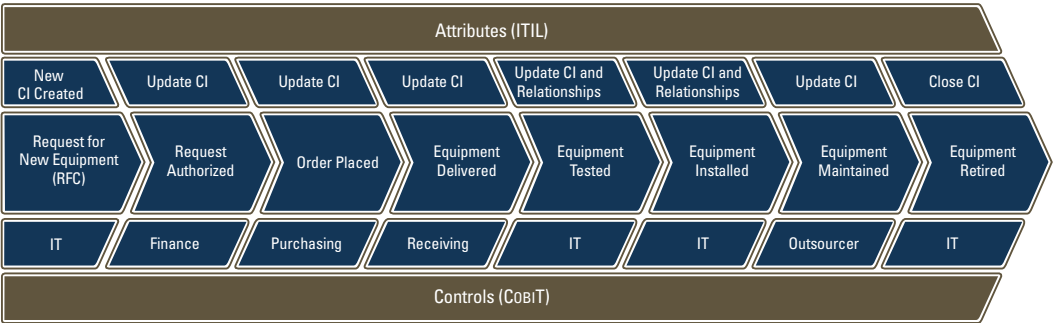


Figure 21.6. CI lifecycle management process

At the end of this step, you should have created a lifecycle for all of your CI groups, identified which attributes you can collect at each step of the lifecycle, installed either electronic or manual methods to populate the attributes, and installed control points to ensure that you have populated the attributes accurately.

Step 22

Build Supporting Processes

Goals and Objectives

In this step, you will build, document, and implement all of the processes required to support and maintain your production CMDB. You must also eliminate any flaws in current support processes.

Overview

Building, documenting, and implementing supporting processes to maintain the CMDB is an obvious and very important step. If you neglect this step, the CI data contained in the CMDB will quickly become corrupted, and users most likely will lose confidence in your CMDB and stop using it.

Many of the processes and procedures required to support a CMDB apply to any database — just the principles and application differ. Most of the processes are described in the ITIL publications.

Although this step is intended to ensure that all of the processes required to manage and support your CMDB are in place, you may identify additional necessary processes down the road. If so, you should repeat this step to ensure that you maintain a high quality of management and support for your CMDB.

Task-by-Task Guide

Figure 22.1 outlines the tasks for building supporting processes.

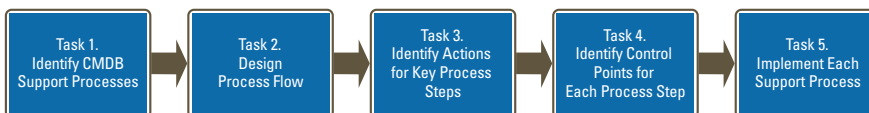


Figure 22.1. Tasks for building supporting processes

Task 1. Identify CMDB Support Processes

Use these five key sources to identify and document potential CMDB support processes:

- » Current database support processes
- » Standard configuration management activities as defined by ITIL
- » ITIL processes
- » Other best practices and governance
- » Supporting processes identified during the project

Current database support processes. Start by reviewing how you manage and control your existing databases. This review will help you to identify many of the processes and procedures required to manage your CMDB. Identify existing databases that are well managed, and consider applying those management processes to your CMDB. At the same time, examine those databases that currently are not well maintained, and avoid repeating those mistakes. The more useful processes you can identify and apply here, the more successful your CMDB supporting processes are likely to be. Be sure to record your findings in a *CMDB Support Processes* document.

Standard configuration management activities. Next, consult the standard ITIL configuration management activities to identify any supporting processes that are unique to the CMDB:

- » **Management and Planning** — Planning and defining the purpose, scope, objectives, policies, and procedures, and the organizational and technical context for configuration management
- » **Configuration Identification** — Selecting and identifying the configuration structures for all infrastructure CIs, including the owners, their interrelationships, and configuration documentation. This activity includes allocating identifiers and version numbers for CIs, labeling each item, and entering information in the CMDB.
- » **Configuration Control** — Ensuring that only authorized and identifiable CIs are accepted and recorded, from receipt to disposal. This activity ensures that no CI is added, modified, replaced, or removed without appropriate controlling documentation (e.g., an approved change request) and an updated specification.
- » **Status Accounting and Reporting** — Reporting all current and historical data concerned with each CI throughout its lifecycle. This activity enables changes to CIs and their records to be traceable (e.g., tracking the status of a CI as it changes from one state to another, such as “under development,” “being tested,” “live,” or “withdrawn”).
- » **Verification and audit** — Reviewing and auditing to verify the physical existence of CIs and checking that they are correctly recorded in the configuration management system.

When considering each of these standard configuration management activities, ask some simple questions: Do they apply to my CMDB? Do we have processes already in place for them and are those processes documented? If so, add these to the *CMDB Support Processes* document.

THE CMDB AND STATUS ACCOUNTING

Javier Leyva Novoa of Quitze Tecnología explains the importance of status accounting in ensuring the accuracy of your CMDB data.

To effectively perform status accounting, you will need to maintain the status of CIs within the CMDB, as well as other related information, including:

- > Lifecycle state of each registered CI
- > Incident, problem, change, and release history of each CI
- > Service level agreement with the business associated with each CI

This process of obtaining and maintaining status information can be manual or automatic, but we suggest that you use a robust configuration management solution that supports CIs of varying complexity, such as entire systems, releases, single hardware items, software modules, or hierarchic and networked relationships between CIs.

Your configuration management tools should facilitate the impact assessment of requests for changes (RFCs) by storing information about the relationships between CIs. This will enable the status accounting to be auditable.

When an organization understands how important the CMDB is in supporting business needs, it will automatically publish on its intranet the status accounting of critical CIs stored in the CMDB, sorted by status and type. Any users that need to know the full status accounting can run a report manually.

Using the statistical results from this accounting, you can find:

- > Behavior patterns
- > Baseline and release identifiers
- > Latest software item versions for a system build or application
- > The number of changes for a system or IT component
- > The number of baselines and releases
- > Usage and volatility of CIs
- > Comparisons of baselines and releases

Tips for effective status accounting include the following:

- > Use statistical reports from your configuration management solution or IT service management applications to assist with this activity
- > Have executive and detailed reports that feed the business needs
- > Record vital statistics (for example, change requests) about the product
- > Filter status accounting according to the permission schema
- > Have flexibility for getting reports from the configuration management tool

If you don't effectively perform status accounting and have IT managers review the reports, then you will lose control of the CI data. This will cause CIs to have an unknown state. You will also lose pattern identification, which means the CI data cannot be used to find the root causes of problems with CIs. Finally, you will lose information about changes and releases associated with CIs.

Excerpt from "Using ITIL's Configuration Management Activities for CMDB Success," VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Stage 4. Construct and Maintain Your CMDB

ITIL Processes. Now, review other ITIL processes that either support or rely on your CMDB. Be sure to include any required processes in your *CMDB Support Processes* document. The ITIL processes that you may need to evaluate are:

- » **Service level management** — Your service catalog, SLAs, OLAs, and UCs may all need processes to support them. If the CMDB is listed as a service in your service catalog, then it will require SLAs, OLAs, and UCs, and you will need to implement processes and procedures to support these activities. Even if you do not have a service catalog, you may decide to have SLAs, OLAs, and UCs. If so, you will need to have supporting processes and procedures in place.
- » **Capacity management** — Concentrate on the processes and procedures required to ensure that the CMDB can meet its current and future capacity requirements.
- » **Availability management** — You need to agree on availability targets with the CMDB customers, and you must put supporting processes in place to ensure that the agreed-upon levels of availability of the CMDB are met and measured. You may decide to specify availability management in an SLA.
- » **IT service continuity management** — To protect your CMDB, you need to have backup procedures in place, secure off-site storage, and planned baselines that are also stored securely off-site. A baseline is a snapshot of the configuration that depicts the structure and details of the CMDB at a specific point in time. The baseline is then available for you as an indication of the original state at that point in time.

Take baselines to provide a sound basis for future work (e.g., a point in the lifecycle of a CI from which you can progress, such as an “accepted” application), a record of what CIs were affected by an RFC and what CIs were actually changed, and a point you can fall back to if things go wrong.

You will also need to determine the role that the CMDB will play when you replace IT services that have been damaged or destroyed. You should already have processes in place for database backups.

Also consult with your continuity management team to identify if the CMDB will play a role during disaster recovery. If so, then you must build the necessary processes here.

- » **Financial management** — You will need processes to manage and report the costs of managing, processing, and maintaining your CMDB.
- » **Incident management** — The service desk is the first line of maintaining the integrity and accuracy of the CMDB. You will need to coordinate with the incident management team to ensure processes are in place to provide prompt notification of any CMDB anomalies. You also need to have processes in place to inform incident management of any significant CMDB updates or changes.
- » **Problem management** — Similar to incident management, you’ll need processes that report and track CMDB anomalies to support problem management.

- » **Change management** — A close relationship must exist between the CMDB and the change management and configuration management processes, so you'll need to identify and implement processes to ensure that they are tightly aligned. In fact, one of the golden rules of ITIL is that any change that alters the status of a CI must go through the change management process. One of the most common reasons for a CMDB to have inaccurate data is that someone has broken this golden rule. You will minimize this risk by implementing strong, appropriate processes between the CMDB and change and configuration management.

A word of caution: Do not implement your CMDB unless you have change management implemented and operating properly. If the change management process is not in place, the CMDB will almost immediately be out of date, and the information will be worse than useless — it will be expensive and wrong.

- » **Release management** — You'll also need to put processes in place to ensure that release and change management are very closely aligned. Release management is also responsible for the Definitive Software Library (DSL). The DSL is a repository where authorized versions of all software CIs are stored and protected. Include version numbers, release numbers, licenses, and possibly patch numbers in the DSL. You must create processes to ensure that the DSL and the CMDB match exactly and that they are audited regularly to enforce synchronization.

Other best practices and governance. In stage 1, "Assemble the Project Team and Define the Project," you identified other best practices and governance requirements. Now you must create and document the processes required to ensure that those best practices and governance requirements are maintained, supported, and integrated with the other CMDB supporting processes.

Supporting processes identified during the project. As you analyzed and created the requirements for your CMDB, you may have identified other necessary support processes. If so, you will need to identify and build these support processes.

At this point, you should have identified and included all of the processes required to support your CMDB in your *CMDB Support Processes* document.

Task 2. Design Process Flow

For every process you identified and documented in task 1, you must now design and document a process flow. This task can be very time-consuming to execute. If you have the resources, assign different individuals to work on different process flows so that this work can be accomplished simultaneously.

Figure 22.2 shows how you might start to document a process flow for a basic configuration audit process that focuses on the CI subcategory of UNIX servers. The purpose of this process is to identify discrepancies between the actual state of the CIs and their recorded state in the CMDB.

Stage 4. Construct and Maintain Your CMDB

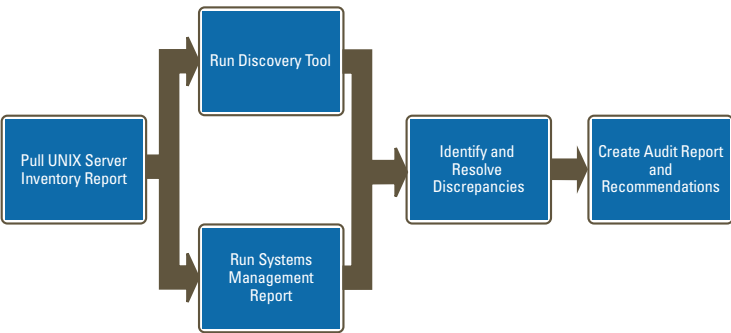


Figure 22.2. Example configuration audit process flow diagram

Task 3. Identify Actions for Key Process Steps

Next, you need to identify, create, and document the actions for each step in the process flow as well as the inputs and outputs for each step. Again, this is straightforward process engineering.

Figure 22.3 shows the example from the previous step and highlights the inputs, outputs, and actions performed at the step “Identify and Resolve Discrepancies,” in the process flow.

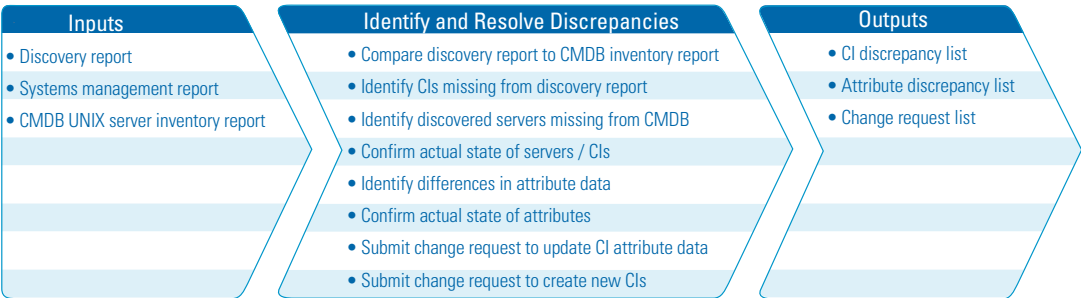


Figure 22.3. Example inputs, outputs, and actions for the process step “Identify and Resolve Discrepancies”

Task 4. Identify Control Points for Each Process Step

You should implement controls on key process steps to ensure quality and performance. You may have selected a control framework earlier in the project, such as COBIT, that can be applied here. Refer to your identified risks, and then identify a control that controls the risk and a test that confirms the function of the control. For example, figure 22.4 maps a key risk: Nonauthorized CI owners might pull a CMDB inventory report of CIs for which they are not responsible. The control might be to clearly identify CI type and subcategory owners, and then use an access management solution to limit their access to CMDB information so they can access only the CI types and subcategories they own. A test might be to have a CI subcategory owner try to run a CMDB inventory report for a different CI subcategory.

| Pull UNIX Server Inventory Report |
|--|
| Risk – CMDB users can pull CI data for all CI types and subcategories they do not own |
| Control – Use access management features to limit data access to CI type and CI subcategory owners |
| Test – Have CMDB user who is not the UNIX server CI owner try to access the UNIX server inventory report |

Figure 22.4. Example risk table for the process step “Pull UNIX Server Inventory Report”

Task 5. Implement Each Support Process

Distribute the process flows, inputs, actions, outputs, and controls to the stakeholders and other relevant teams that may be affected by the supporting processes, and then gather feedback. When you, the stakeholders, and other relevant teams have agreed to the processes, you can then implement each process.

Summary

Strong supporting processes are necessary to ensure the accuracy of your CMDB data. Incorrect CI information will quickly reduce user confidence and jeopardize user acceptance. Take the time now to build the appropriate supporting processes to ensure that your CMDB is useful and achieves the expected benefits.

Step 23

Populate Your CMDB

Goals and Objectives

The goal of this step is to fully populate your CMDB with all of the CIs and their associated attributes and relationships. Populating the CMDB also includes gathering and cleaning data, developing import scripts, writing reconciliation rules, and verifying that each data import is successful. At the end of this step, the CMDB will be ready for use.

Overview

Now that you have implemented the required technologies, tools, and resources, it is time to populate your CMDB. You can choose among three basic methods for populating the CMDB. One is to import data from a source such as a spreadsheet, a download from an application's database, or a list. After the initial population, that source isn't used again. A second method is to import data from a source or repository where ongoing synchronization with the CMDB will occur. This unidirectional or bidirectional synchronization requires the use of reconciliation and synchronization rules that manage both the initial and ongoing process of data import. A third method is to manually update data fields using a graphical user interface (GUI) to the CMDB. The plan you developed in step 17, "Plan the CMDB Population," should include some combination of these methods.

In this step, you will follow the plan you created, and you will populate the CMDB. The overall approach is to create each temporary data set, clean the data, import, verify, and repeat until the CMDB is populated.

If any step of data import is not correct, you will clean that failed import, adjust the import scripts and reconciliation rules if appropriate, and repeat until successful. Then you manually will input any additional CI data, such as service CI to infrastructure CI relationships. After verification of the CMDB population, you will release it to production.

Task-by-Task Guide

Figure 23.1 shows the tasks for populating your CMDB.

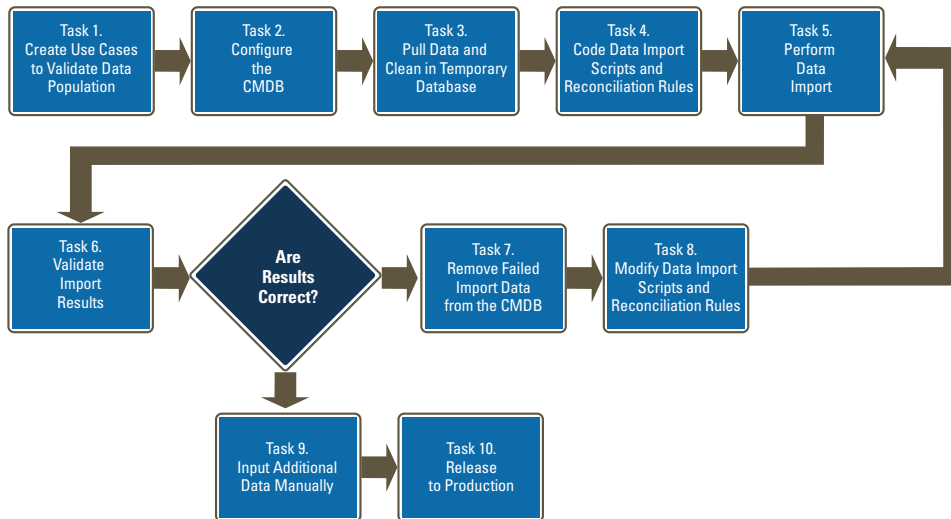


Figure 23.1. Tasks for populating your CMDB

Task 1. Create Use Cases to Validate Data Population

As you follow the plan you developed in step 17, “Plan the CMDB Population,” you should manage the import of each data set as a subproject. Developing use cases can help you test the data imported into your CMDB for each subproject and verify the results after multiple subprojects have been completed. Some import subprojects will create new CIs. Others may modify the CI attributes of CIs already in the CMDB. Others may create or modify relationships between CIs. Each subproject should have a use case to confirm that each import worked as expected and that all attribute and relationship data elements are correct.

Depending on your integration and federation strategy for applications and data sources, your use cases should test two methods of data population: import from a temporary file to the CMDB, and unidirectional and bidirectional synchronization and reconciliation between the data source and the CMDB.

As an example of how you might populate your CMDB in stages and then test that each subproject succeeded, consider the creation of a CI class for UNIX servers. You might start by doing a one-time pull of data from an asset management system, filtered to collect all inventory information related to UNIX servers. That data can be filtered and cleaned as you pull it, or once it is in a temporary database. That basic asset information might be imported into the CMDB as a one-time import and not have ongoing synchronization with the CMDB. The use case should verify that the CIs are created and the inventory attributes are populated. You may then use a discovery tool to capture the current operating system and application attribute data for those UNIX servers. Or, you might set up this import for initial population as well as

regular synchronization based on all UNIX server CIs in the CMDB. The use case should test the initial import and verify that the business rules that manage the ongoing synchronization are working to support an ongoing update. You might then set up a regular collection of CI status attribute data from a network management system. That might not require a data pull to a temporary database, but may only need to be set up as a regular synchronization. The synchronization could run as the initial import, and the use case should verify proper function and data cleaning. Finally, you may want to conduct a use case test that verifies that all UNIX servers have inventory, software, and status attributes populated, so you can confirm that the UNIX server CI class was successfully created.

Examples of use case tests for the initial creation of CIs include the following:

- » Creation of new CIs with related attributes and relationships
- » Creation of relationships for existing CIs
- » Update or modification of any of the CI attributes and relationships
- » Data transformation through format conversion, text truncation
- » Text extension through suffix or prefix
- » Remapping of data through mapping tables
- » Data reconciliation and merging — naming
- » Inheritance — if one source has certain properties, then does the second source take on those properties and attributes
- » Deletion of any combination of CI attributes and relationships, or the entire CIs

By developing use cases for the import of each data set, you are identifying the criteria you will use in task 6, “Validate Import Results.”

Task 2. Configure the CMDB

Now that you have prepared your use cases, it is time for final preparation of the CMDB. In step 20, “Construct Your CMDB,” you completed the installation of your CMDB solution. Whether your CMDB is a packaged or custom solution, your data model should match your requirements before you begin importing data. In this task, you should change the settings of the CMDB application and modify the out-of-the-box data model to meet your specific data model requirements.

Work from the requirements you developed in steps 12 through 15, and step 17, “Plan the CMDB Population,” to configure the solution to meet your requirements for:

- » Definition or modification of CI classes and CI types
- » Definition or modification of attributes (narrow down or add new ones)
- » Definition or modification of the relationships (narrow down or add new ones)

When possible, focus on removing capabilities that aren’t explicitly required, so that you can minimize complexity and prevent scope creep.

Task 3. Pull Data and Clean in Temporary Database

During this task, you will pull from target data sources and prepare each data set for import into the CMDB. In many cases, you will do a bulk data pull from a specific data source and create a data set in a temporary database.

The key consideration in this task is data quality. As you planned your CMDB population, you should have assessed the quality of each data source. At this point, the data sources should meet some minimum level of data quality and consistency. However, even if each data source is a high-quality and trusted source, you still need to check quality before you import data into your CMDB.

Cleaning the data can occur at three different stages. First, you can clean and filter the data as you pull from the data source. For example, you can use SQL commands to pull data about all UNIX servers that have a purchase date later than 2000 and a serial number that is not blank. Second, you can clean the data once the data is pulled and in the temporary database. For example, you can pull data into a Microsoft Excel spreadsheet, sort by the purchase date, and delete all rows that have a purchase date before 2000. Third, you can clean during import to the CMDB. For example, if you plan to have ongoing synchronization with the data source, you can filter out servers purchased before 2000 each time you import.

Check the data using your use case. You developed each use case to verify that data is accurate after it is imported into the CMDB. Verify that the same data is accurate in the temporary database before you import as well. This check will help you troubleshoot data errors found in the CMDB.

Also consider the dynamic nature of the source. Your initial data pull is a snapshot in time. Determine how often the data you are pulling changes. For data that is very dynamic, try to minimize the time between data pull and CMDB production release.

If you have a high volume of data to import, use automated data-checking tools or write special data-checking scripts to conduct the data scan and profile the data against expected norms. Consider options such as these:

- » Run statistical tools, such as a box plot, on numerical data to verify that data ranges make sense. Then identify outliers and correct the data before import to the CMDB.
- » Sort data by field and look for outliers and correct data before import.
- » Use a random sample of CIs to manually check data. Are all fields filled in? Are the formats correct? Do the data ranges make sense?

Examine the following criteria:

- » The field is not blank
- » The data conforms to a consistent data format (e.g., date format, integer versus floating point, text versus numeric)
- » The data is normalized (i.e., no duplicates or variation of the same thing)
- » The data is of the right level of granularity consistently

- » The data has a reasonable range of values (i.e., the data does not fall below a minimum or above a maximum range)
- » The data has an expected distribution of values compared to the real world (e.g., x percent of data belongs to this category and y percent belongs to another category)
- » The data conforms to certain business rules (i.e., if the field x is xxx , then the field y should be yyy)

Whether you clean data as you pull from the data source or verify the quality once it is in a temporary database, you should be confident that your data set is pristine before you import data into the CMDB.

Task 4. Code Data Import Scripts and Reconciliation Rules

The final task before you import data is to code import scripts and reconciliation rules. Automating the import and reconciliation of data is critical if you plan to integrate and synchronize the data source for ongoing updates. The most common approach to populating the CMDB is to create a data set and import for initial population, then activate the integration and reconciliation rules for ongoing update.

The two general methods to import data into the CMDB are the following:

- » From a file that is defined in XML format or a delimited file (e.g., Excel CVS format) that is then imported into the CMDB using the CMDB data import utilities
- » Data exchange between the temporary database and the CMDB database programmatically using APIs and/or Open Database Connectivity (ODBC)

The tools and features of your CMDB will largely determine the method you choose and the complexity of the task. Some CMDB solutions have extensive import and reconciliation features. That greatly simplifies this task. If not, then you may be writing code. Make sure that you have program source controls in place to track changes to the developed code. You will likely import and modify scripts multiple times. Source-code control will greatly reduce confusion and rework. Also, use a development environment so that developers can perform unit testing.

You determined the logic of your reconciliation and business rules when you identified data sources in step 17, “Plan the CMDB Population.” Now is the time to code and activate the business rules. Implement business rules that guide reconciliation before the CMDB is rolled into production. As soon as the CMDB is populated with data, the information can become obsolete as normal IT functions alter the infrastructure, applications, and services across the organization. You should implement your reconciliation tools and logic immediately to help keep the CI data accurate and usable.

Task 5. Perform Data Import

Follow your plan to import the data from your temporary data set into the CMDB. The majority of data import most likely will be bulk imports using custom written scripts or tools provided with the CMDB — not real-time data import from discovery tools.

Stage 4. Construct and Maintain Your CMDB

You might decide to take a “big bang” approach for the initial CMDB data load, or you might take an incremental approach that lasts several weeks. If you choose a “big bang” approach, set up a preproduction environment, especially when you are loading the CMDB for the first time and plan to have a high volume of data. A preproduction environment can help you:

- » Gather information on the length of time the data population process will take. This information is especially useful if, on the day of the implementation, the data population will affect other source data systems that are in production. You can use the collected data to predict the actual time the data population will require on the production system.
- » Identify the acceptance and rejection rate of the CMDB population.
- » Gain an opportunity to closely examine rejected data and determine the required fixes necessary for the import scripts and/or reconciliation rules.
- » Establish a staging area to control the “good” data that will be transferred to the CMDB on the day of the implementation rollout.

Task 6. Validate Import Results

The challenge with populating anything in bulk is that large data sets make it more difficult to validate that everything is correct. Follow the use cases you have developed and validate the data you imported. Involve the CMDB owner, data owners, and process owners. Make sure everyone has performed due diligence and is comfortable before releasing the CMDB to production.

For each data set imported, verify that the results of each import are correct. If they are, repeat the import process for each data set. Then proceed to task 9, “Input Additional Data Manually.”

If your use case failed, you need to decide if you are going to back out the import or try to manually correct the errant data. You may also try an additional partial import to correct just the errant data. Note that this is intricate, complex work. With large volumes of data now partially imported into the CMDB, both the likelihood and consequences of making errors that impact overall CMDB quality are high. Be careful. Turn off instant messaging and ignore the phone. Double-check each time you attempt to correct data in the CMDB.

Task 7. Remove Failed Import Data from the CMDB

If your use case detected a problem with your data import, you should remove the failed data before modifying your import scripts and re-importing the data.

Another important concept to consider here is how to respond if the data source is the source of the data error. The decision to either correct the data in the CMDB or correct the data at the source and re-import depends on the following:

- » The amount of effort required to correct the data using either of the two methods
- » Whether the data source could be updated (e.g., the source data may be used by other applications that require a different format than the one used on your CMDB)
- » If corrected at the source, would the data stay corrected before re-loading in the CMDB?

In either case, if you need to remove the errant data from the CMDB, go to the next task.

As you proceed through multiple iterations of importing a single data set, you should consider some form of data version control that matches your import script version control, so that you can keep your data and tools synchronized.

Task 8. Modify Import Scripts and Reconciliation Rules

After you remove an errant data set from an import that failed, you need to modify your import routine and then import again. At this point, the assumption is that the source of the failure is not the data, because you cleaned the data before you conducted your initial import.

You may need to import and test the reconciliation rules multiple times, especially for imports that will be repeated for ongoing data synchronization. You should import, test, remove, edit import rules, and repeat until you are confident that the quality of the import process is high enough to automate.

Task 9. Input Additional Data Manually

After you have imported all your data sets, then you can manually add any additional CI data to your CMDB. Not all data will come from an existing data source or discovery tool. Some data, especially relationship data related to mapping service CIs to system CIs, needs to be entered manually. Making associations assumes that the CIs are fully populated, so this task should be the last before the production release.

Be cautious if you plan to manually update more than a few dozen CIs using the GUI provided with your CMDB. Manual updates are notoriously prone to error. If you have hundreds or thousands of relationships to code, you may want to consider creating a temporary data set that you can clean and then import.

Task 10. Release to Production

Your final task is to release your CMDB to production, and you could follow two basic approaches. You could copy the database from your preproduction instance to the production instance that has no data. Or, if you do not have a preproduction development environment, then you can make the CMDB available to production users.

Timing is important here. You pulled data from various data sources as snapshots. Your infrastructure is changing. You need to move your CMDB into production and synchronize your data sources to prevent drift. On the day of your production release, you should synchronize your CMDB with integrated data sources. Depending on how much time has elapsed since you pulled from various sources to create your staged data sets, your initial synchronization may cause some performance issues for the integrated data sources.

Summary

Congratulations. You have built and populated a CMDB. This is an exciting milestone. The previous data mapping and planning should pay off with a smooth production rollout. Remember the fundamental requirements of the data consumers and mitigate risks to ensure the long-term success of your CMDB.

Step 24

Train the CMDB Team and Users

Goals and Objectives

In this step, you will identify and provide all of the training and education that the configuration management team and CMDB users require. Training is critical for ongoing effective management and maintenance of the CMDB. Training of CMDB users and process owners who now use CMDB data as part of their functional processes is also critical to ensure user adoption and to achieve the expected improvements that justified the CMDB project.

Overview

This step outlines critical points for you to consider when defining the training requirements and building a training plan for both the team that will maintain and manage your CMDB as well as the process owners who will use CMDB data.

Task-by-Task Guide

Figure 24.1 shows the tasks for training the CMDB configuration management team and the users of CMDB data.

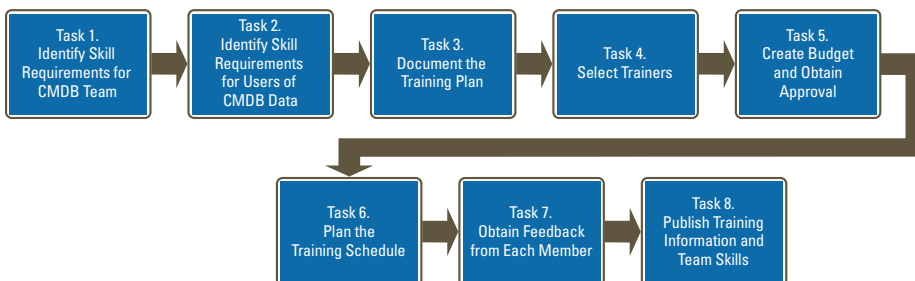


Figure 24.1. Tasks for training the CMDB configuration management team and users of CMDB data

Task 1. Identify Skill Requirements for CMDB Team

Start by reviewing the supporting processes that you documented in step 22, “Build Supporting Processes,” and clearly identify the skills required for those who are involved in each process. While the specific skills that are necessary depend on the processes and CMDB application you have selected, most CMDB implementations require specialized skills in the same key areas.

Consider the core functions of the CMDB introduced in step 2, “Obtain CMDB Knowledge.” In this step, you learned that every CMDB solution should include several basic functions.

Figure 24.2, originally introduced in step 16, shows the core functions included in a CMDB.

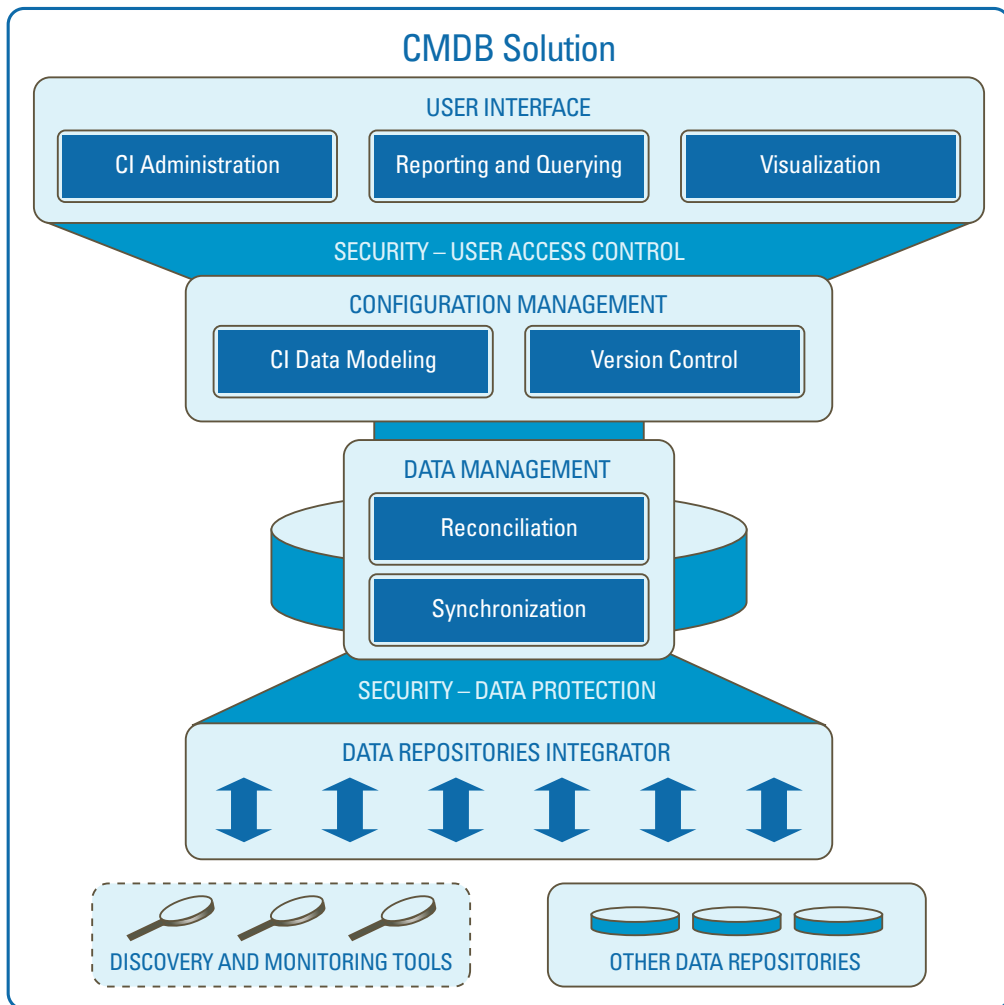


Figure 24.2. Taxonomy of CMDB core functions

Use these generic functions, along with basic process requirements, to start a checklist of required skills. Following is a sample skills checklist:

CMDB Implementation Skills Checklist

- » Administer discovery and monitoring tools
 - Configure discovery passes
 - Maintain the repository of discovered data
- » Administer the integration between discovered data repositories and the CMDB
- » Reconcile CMDB data
 - Identify and manage the distinct containers or partitions for discovered data
 - Define rules for identifying when instances in multiple containers refer to the same CI
 - Define rules for merging instances in multiple containers into a master container or the single source of truth
 - Schedule reconciliation identifications and merges
 - Maximize other vendor-specific reconciliation features
- » Define links between CIs in the CMDB and federated data
- » Manage the data model
 - Add attributes or classes to the object model as needed
 - Hide aspects of the object model you do not plan to use
- » Manage access control
 - Assign permissions to perform administrative tasks such as creating reconciliations or customizing the object model
 - Assign permission to the CI data stored within the CMDB
- » Manage the user interface
 - Define queries and build reports against CMDB data
 - Administer the data visualization component
 - If the CMDB application you have selected allows for the customization of this component, understand how to perform these customizations
 - Maximize other vendor specific user interface features
- » Move CMDB data, structures, and configurations from a development server to production
- » Understand the processes, defined by the CMDB project team
- » Understand basic ITIL processes, especially the configuration management process
- » Understand the integration points allowed by your CMDB application (It is not necessary to be fully versed on integration points, such as an open API, if you do not plan to use these integrations in your implementation. However, it is useful to be aware of the possibilities provided by your tool as your CMDB implementation matures.)

THE IMPORTANCE OF TRAINING

Two experts stress the importance of training to ensure that your CMDB provides the expected benefits.

Frederieke C.M. Winkler-Prins, co-founder of Service Management Partners (SMP): “Enterprises need to ensure that they have the time in the schedule and money in the budget for training. Training that covers process awareness, as well as how to use supporting tools, fosters broader organizational acceptance and makes the transition to new or changed ways of working.”

Excerpt from “Accelerate Your CMDB Implementation with a Best Practice Process Model,” VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Carlos Bertaso, systems engineer for CSC Brasil: “Throughout our engagement with the customer, there was a constant need to enhance their knowledge of ITIL processes related to the CMDB solution. The good news is that the more that individuals learned about ITIL concepts and the benefits of a CMDB, the more enthused and dedicated they became to ensuring the deployment was a success.”

Excerpt from “15 Recommendations to Achieve CMDB Success,” VIEWPOINT, Focus on: CMDB, published by BMC Software.

Review this checklist and augment it with any additional skills required by your implementation or the CMDB application you have selected. Refer to the documentation you created in earlier steps regarding your CMDB tools and processes. Take care to customize the list for your needs and ensure that you have clearly identified all necessary skill sets. For example, you may have multiple discovery tools from multiple vendors and each will require specialized knowledge.

When your skills checklist is complete, a good way to verify your work is to follow the path of several CIs through the lifecycles you documented in step 21, “Create CI Lifecycle Management Processes.” From each possible starting point, follow a CI as it moves from discovery or manual entry, to synchronization, reconciliation, and consumption. As you follow the CI, consider each time it is “touched” or modified and verify that you have recorded the necessary skills at each juncture. Finally, remember that the content of the CMDB should be the responsibility of the data owners, particularly in the federated model. Do not add skills related to content validation to this list.

Task 2. Identify Skill Requirements for Users of CMDB Data

Beyond the skills required for the configuration management team, consider those required for those process owners and users who will leverage CMDB data. Typically, direct access to CMDB data is restricted to select individuals, and may be limited to those on the configuration management team. However, the data stored in a CMDB is usually consumed by one or more applications with a much larger user community, such as a change management or incident management application. Training users on applications that leverage the CMDB is critical to ensuring that the CMDB is accepted and that it provides the expected benefits to the organization. In previous steps, you identified process areas that you expect will benefit from process improvements as a result of the CMDB implementation, and defined CI requirements based

on these expected benefits. Now, work with the key constituents and process owners for each process area to identify and document:

- » Functional processes that will change with the addition of CMDB data
- » Integrated or federated applications that will use the new CMDB data
- » Work procedures that will change
- » New work procedures
- » Additional skills required as a result of the new work procedures
- » A detailed list of all users who will follow the new work procedures

You will use this information to create a training plan to update user skills.

Task 3. Document the Training Plan

Once you have defined all of the skills necessary for a successful implementation, analyze the current knowledge and skills of your configuration management team and users. Remember the implementers of your CMDB must not only understand the software you have selected, but the processes you have defined. Use the skills checklist you created in task 1 to produce a gap analysis of the skills required.

After you have identified the knowledge gaps, begin to build a training plan. You will complete this plan when you define the training schedule in task 7.

Make certain to include the following components in your plan:

- » Knowledge transfer from the CMDB project team
 - Identify the project history and goals
 - Explain the governance and compliance issues related to the CMDB and the corresponding best practices you have defined
 - Explain all processes and procedures created by the CMDB project team
- » ITIL's configuration management process
 - Determine if you can achieve this component through the knowledge transfer from the CMDB project team or if you require formal ITIL training from a vendor
- » Detailed training on the administration and use of the CMDB software application you have selected
- » Detailed training on new work instructions for all users of affected processes and applications
- » Information about how using best practices will benefit the business, as well as the configuration management team and users. Benefits to the business include: fewer errors, less downtime, a higher quality of service. Benefits to the configuration management team and users include: lower stress levels; a less chaotic work environment; more time and energy for non-work activities; and more time for innovation and proactive, creative solutions.

Task 4. Select Trainers

Next, identify vendors and training courses that meet the needs of your skills checklist and training plan. When selecting a vendor for your training, consider the following questions:

- » Can the vendor fulfill all of my training needs?
 - Does the vendor offer ITIL and best-practices training as well as software training?
 - Does the software training explain how the tool supports the ITIL configuration management process?
- » Can we benefit from training customized for our specific CMDB implementation?
 - If so, does the vendor offer customized training?
- » Do the courses offered by the vendor meet the requirements outlined in my skills checklist?
- » Does the software training allow for hands-on practice with the tool?
 - Investigate the amount of hands-on exercises or labs included in the training
 - Inquire about the format for the labs (exercises that promote thought and allow participants to work through real-world challenges in a classroom environment are much more effective than simple labs that reproduce steps you can find in product documentation)
- » Does the trainer seek to inspire and motivate staff?
 - Find out what benefits the vendor anticipates that the trainees will receive from implementing best practices
 - Learn whether the vendor includes a discussion of these benefits in training, or simply focuses on learning definitions and passing an exam.

Finally, obtain the prices for private and public courses as well as a training schedule, and make a vendor selection.

Task 5. Create Budget and Obtain Approval

Record your training and vendor selections in a training budget. Submit this budget, your recommendations, and justifications to the management team for approval. Many organizations assign a training budget as a per-annum cost per employee. The budget for your CMDB project also might have included training. Review the training budget again when you obtain approval, because the management team might have revised it.

Task 6. Plan the Training Schedule

Review the latest timetable of course offerings supplied by your selected vendor. Use this table to discuss the ideal schedule with the team members to be trained — and their managers. You will need to take into account the team's current tasks and commitments. Make sure everyone agrees on the training schedule and locations, and adheres to that agreement, and then book the training. During booking, clearly identify the costs and the cancellation policy. It is also important to note the vendor's policy for cancelling course offerings.

Task 7. Obtain Feedback from Each Team Member

After people have completed the training (which may include more than one course), find out whether the course delivered on the expectations and prospects published. Ensure each attendee is satisfied and ready to put into practice the newfound knowledge and skills. The more that attendees can use their new knowledge upon returning to the workplace, the better their retention of the new information. When significant time elapses before the skills acquired in training are put to use, the information learned is forgotten or ignored and the overall benefits are lost.

If any problems occurred with or during the training, address them as quickly as possible with both the attendee and the training vendor. If the course completely fell short of expectations, insist on a refund and re-book the training with another vendor.

Task 8. Publish Training Information and Team Skills

Publish and make available information about the training courses and costs, together with a list of the training vendors selected and the attendees. This information will help to ensure that everyone knows who has what role, responsibility, and skills, as well as where to go for further training.

Summary

Use the guidelines defined in this step to clearly outline your training requirements. Select one or more training vendors who can meet these requirements and insist that your software training includes extensive hands-on and thought provoking practice. Training is sometimes left to an afterthought, but be sure to recognize its importance. All of the work you have devoted to your CMDB project will be judged against the success of your implementation. A critical component of your implementation success is ensuring the configuration management team and the users of CMDB data are properly trained.



STAGE 5

Driving Ongoing Value

Stage 5. Driving Ongoing Value

During stage 5, the final stage of your CMDB project, you will implement measures and metrics for ongoing CMDB management. You will also follow a continual service improvement program to ensure targets are met for key metrics.

In stage 1, “Assemble the Project Team and Define the Project,” you collected information step by step to build a business case that was approved by the PEB and project sponsor. In stage 2, “Define Requirements and Create IT Service Model Blueprint,” you worked with key stakeholders to document and obtain approval on a broad range of solution requirements. In stage 3, “Select CMDB Solution and Tools,” you selected a CMDB and planned the population, including federation and automation. In stage 4, “Construct and Maintain Your CMDB,” you built and populated your CMDB, and moved into production with supporting processes.

In this stage, you will identify a broad range of metrics to track in order to effectively manage CMDB-related support processes, as well as the services and processes that use the CMDB to drive benefits to the organization. Then, when you implement your continual service improvement program, you will focus on a narrow set of key metrics and related targets. If met, these key metrics will ensure that the CMDB continues to have quality data and that it will enable the key benefits that were expected when the overall project business case was approved.

Figure 1 shows the steps followed to drive ongoing value.

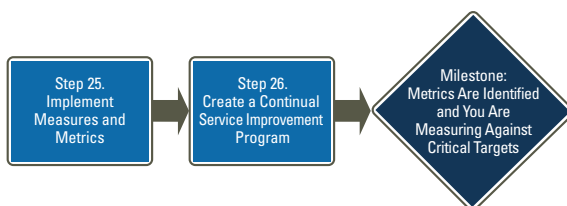


Figure 1. Steps for driving ongoing value

Each step contains specific goals that you should meet before moving forward. You may want to use the following as a checklist.

Step 25.

Implement Measures and Metrics

Goal: The goal of this step is to implement metrics that enable effective ongoing management of CMDB-related processes. You will identify metrics that will support the overall goals of the project, and metrics that help the CMDB manager and CI owners with the ongoing management and update of the CMDB.

Completed: ____

Step 26.

Create a Continual Service Improvement Program

Goal: The goal of this step is to follow a continual service improvement program that will help ensure overall project success. Identifying a few critical metrics will ensure that the key processes that rely on the CMDB achieve expected benefits, and will also ensure that the CMDB remains a quality system.

Completed: ____

Milestone: Metrics Are Identified and You Are Measuring Against Critical Targets

A broad range of metrics is identified that help manage maintenance of the CMDB and help ensure expected project goals and benefits are achieved. You are closely monitoring a few metrics against critical targets.

Milestone achieved: ____

Step 25

Implement Measures and Metrics

Goals and Objectives

In this step, you will implement metrics that enable effective ongoing measurement and management of CMDB-related processes. You should identify metrics at three levels. First, identify metrics that align with the goals and objectives of the business. (For example, “Reduced MTBF as a result of accurate CMDB data, which translates to X dollars saved.”) Second, identify metrics that will help the process owners for configuration and asset management to manage and improve these processes. Third, identify metrics required for all other stakeholders, including other process owners who interface with CMDB data, and staff who deserve recognition for their performance.

The CMDB is an enabling technology. It gives you visibility and, therefore, control of the components and relationships that make up the IT infrastructure. Your metrics need to reflect this, along with identifying areas for continual service improvement.

Overview

Metrics provide the visibility into what is currently going on and the basis for identifying critical success factors. They are essential for planning, managing, and improving service management processes. Be sure you know who is using this information, and how they are using it, in order to both measure the right things and to benefit from that effort.

Metrics should be communicated in appropriate ways to their intended audience. Good metrics help IT staff to focus on what is important to the business and, therefore, to make better, more business-aligned IT decisions.

Metrics can assure customers that their needs are both understood and being met. They can be used by service providers to identify areas of weakness and implement improvements. They can be used by managers to motivate staff and ensure the correct behaviors. Metrics also provide essential data to achieve and maintain regulatory compliance.

Stage 5. Driving Ongoing Value

As a reminder, metrics and measures must always meet SMART criteria: specific, measurable, achievable, realistic, and time-based.

All metrics should be implemented in the context of a broader performance monitoring methodology used by the IT organization. Figure 25.1 shows a good general measurement cycle. This cycle begins with determining goals and then proceeding to identifying measures that will show whether the goals are being achieved. Ongoing measurement and analysis of the data will either enforce the current process or enable the correction and adoption of the process for continual service improvement.

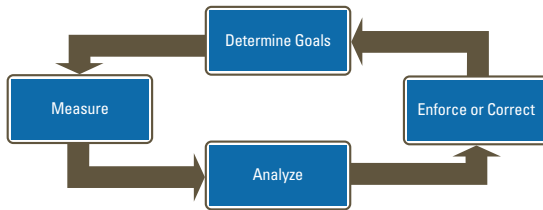


Figure 25.1. General measurement cycle

This step presents a simple process of identifying measurement domains tied to key project goals and CMDB management processes. For each domain, you will create a simple scorecard that you can use to track key measures. You'll set targets and tailor reports to the audience. You'll also look at how the focus of the scorecard should change over time, obtain approval of the metrics, and clarify how you will use the metrics for communication with management.

Task-by-Task Guide

Figure 25.2 shows the tasks for identifying and installing metrics.

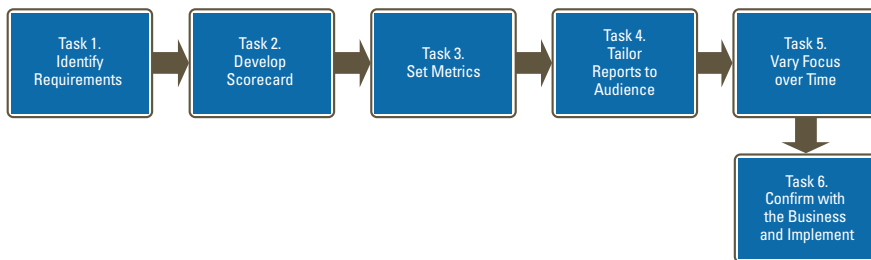


Figure 25.2. Tasks for identifying and installing metrics

Task 1. Identify Requirements

Identify requirements to measure compliance, make business decisions, and check process maturity to facilitate continual improvement. Begin by deciding what you are trying to achieve. After all, you need to identify your destination before you can plot your course. As you list the requirements, include project goals identified in stage 1, key stakeholders, risk assessment, and other considerations.

DEFINING WHAT TO MEASURE

Effective service measures concentrate on a few vital, meaningful indicators that are economical, quantitative, and usable for the desired results. If there are too many measures, organizations may become too intent on measurement and lose focus on improving results. A guiding principle is to measure what matters most to the business.

IT has never lacked in the measuring area. In fact, many IT organizations measure far too many things that have little or no value. But care must be taken to align measures to the business and IT goals and objectives. And, organizations need a measurement framework to guide them in the area of service measurement. Be sure that the proper measures are in place to support the following:

- > Service performance against the strategic business and IT plans (part of a balanced scorecard or IT scorecard)
- > Risk and compliance with regulations and security requirements in the service design and transition practice
- > Business contribution, including but not limited to, financials (how IT supports the business in delivering services)
- > Key IT processes that support the service
- > Internal and external customer satisfaction (measuring customer satisfaction to ensure customers' needs are being met)

Task 2. Develop a Scorecard per Domain

A scorecard helps provide a more comprehensive view of a business, by measuring from four different perspectives (rather than just looking at financial information). This in turn helps organizations make the best decisions to support their long-term interests.

Once you have identified the measurement domains, you can then identify specific measures and metrics for each. You can use a simple four-part scorecard that helps balance the measures across four metrics with four different purposes, as shown in figure 25.3.

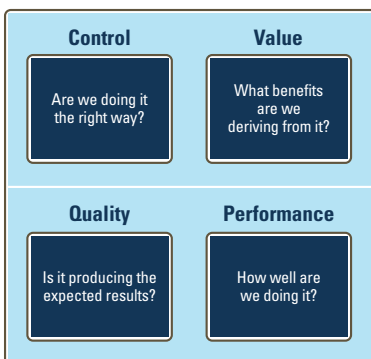


Figure 25.3. Scorecard that balances four types of measures

The four types of measures include:

- » **Control measures** — Control measures answer the question, “Are we doing it the right way?” Control measures are used to determine whether people are following the defined process. If the process is documented, but not followed, there likely will be such wide variation in measures that the data quality is compromised. Control measures might include the percentage of detected changes that are not approved.
- » **Value measures** — Value measures answer the question, “What benefits are we deriving from it?” Value measures are used to monitor improvements directly related to project goals and expected impacts that were used to justify the project. Example value measures are those used in step 5, “Build a Business Case.”
- » **Quality measures** — Quality measures answer the key question about the output of the process, “Is it producing the expected results?” These measures might include SLAs related to the process being measured, such as responsiveness, timeliness, or accuracy.
- » **Performance measures** — Performance measures answer the key question about executing the process, “How well are we doing it?” These measures might include availability, stability, and performance, as well as efficiency and cost-effectiveness.

As part of this task, it will be necessary to identify the source of each measure.

A wide range of information is available about IT metrics. The ITIL books are a good source of information about the IT service management processes outlined here. Another publication worth mentioning is *Metrics for IT Service Management*, by Peter Brooks, which provides specific detail about how to establish effective metrics for ISO processes. In addition, *Making Metrics Work*, which is part of the *Achieving ISO/IEC 20000 series*, gives sample metrics and reports for every process.

The following example uses incident management to illustrate how you can use the scorecard to track CMDB key stakeholder metrics. In step 4, “Review and Define Benefits,” you developed an impact estimate table for a hypothetical configuration management goal, as shown in Figure 25.4.

| Goal | Benefits | Impact |
|--|--|--|
| Provide accurate information on configurations and their documentation to support all other service management processes | <ul style="list-style-type: none">• Hard benefit: improve process integration, communication, decision making, and compliance• Hard benefit: experience fewer failed changes, increase availability, reduce incident rate, and improve first-fix rate• Soft benefit: improve customer satisfaction | <ul style="list-style-type: none">• 30% cost savings as a result of reduction in failed changes; 20% cost savings as a result of increased availability; 35% reduction in incidents; 40% cost savings realized by incidents solved by first level support staff• IT perceived as supporting and enabling business |

Figure 25.4. Example impact estimate assessment

You can use a wide range of sample metrics to track these goals and the achievement of expected benefits of configuration management. After identifying measures currently used by the organization and after identifying new measures, you might create a potential sample metrics list that includes:

- » Percentage of CIs that have data updated and verified automatically
- » Customer satisfaction
- » Number of unused licenses
- » Number of failed RFCs from bad CMDB data
- » Number of unauthorized configurations
- » Number of incidents from failed changes caused by wrongly documented CIs
- » Number of breached SLAs because of CMDB error
- » Percentage of inaccurate CIs

All metrics need to be under the control of change management.

After reviewing available sources and grouping these metrics based on the types of measures in the four-part scorecard, you might then focus on the key metrics, as figure 25.5 demonstrates for a CI.

| Control (Are we doing it right?) | Value (What are the benefits?) |
|---|--|
| <ul style="list-style-type: none"> Evidence of bypassing the process | <ul style="list-style-type: none"> End-to-end visibility, such as the percentage of total services delivered that is mapped |
| Quality (Is it producing expected results?) | Performance (How well are we doing it?) |
| <ul style="list-style-type: none"> Output related, such as data accuracy | <ul style="list-style-type: none"> Efficiency of process Cost effectiveness Availability |

Figure 25.5. Scorecard developed from an impact assessment

The CMDB team should develop a set of metrics that measures the performance of the team and CMDB-related processes. The team also should collect information that indicates the accuracy of the CMDB data. These metrics may also be shared with individuals in IT functions who have been identified as CI owners. Special versions of the reports of CMDB metrics may be created for each owner, so that the data focuses on just the CIs that are the owner's responsibility.

You can develop metrics for a wide range of CMDB-specific measurement areas to track the performance of the CMDB and data accuracy, as outlined in the following text.

Stage 5. Driving Ongoing Value

Configuration management and CMDB information:

- » Information on the number of CIs by category, type, version, and status
- » CI growth and CMDB capacity information
- » CI and CMDB rate of change information

CMDB data accuracy information:

- » Results from configuration audits, including information on non-registered or inaccurately registered CIs detected in the audit
- » CI aging reports that show CIs that haven't changed or been updated over a period of time
- » Incidents and problems traced back to configuration management issues
- » Percentage of changes that did not follow a change process
- » RFCs that were not completed successfully because of poor impact assessment, incorrect data in the CMDB, or poor version control
- » The number of CI changes/updates per month due to CMDB errors

After identifying the measures currently used by the organization, and after identifying new measures, you might create a scorecard that includes the measures shown in figure 25.6.

| Control | Value |
|---|--|
| <ul style="list-style-type: none">• Percentage of changes to CMDB without authorized RFC | <ul style="list-style-type: none">• RFCs that were not completed successfully due to inaccurate data in CMDB or poor version control• Incidents and problems traced back to configuration management issues |
| Quality | Performance |
| <ul style="list-style-type: none">• Number of CI errors identified in monthly CI audit• CI aging reports | <ul style="list-style-type: none">• CI and CMDB rate of change information• Percentage of CIs updated through automated, rather than manual, process |

Figure 25.6. Scorecard to track configuration management and CMDB manager metrics

Task 3. Set Metrics

As you chart your course, you will need to set the metrics to be used for each process or output. Revisit what you need to measure and why. Determine what the process or output will be used for. Also consider whether it can be measured accurately.

Do a maturity assessment (of people, process and technology) at the outset, as the lower the maturity of the process, the less likely it is that the data will be highly accurate. Such an assessment is important in order to prevent the business from making a critical decision based upon faulty data. You'll also want to factor the maturity of the process into the measurement for this reason.

There is another reason to do the maturity assessment. A process owner must have an understanding of the people and technology, as well as the process, in order to accurately identify an optimum metric that will be both challenging and achievable, as outlined in the following text.

Task 4. Tailor Reports to Audience

You must identify the audience for the metrics you are tracking. The way you report or publish results may vary depending on who is reading the data. Even though you have a goal, related metrics, and targets set, you should think about the different audiences of the reported data, so you can determine how best to present the information.

For example, you may have a service availability goal of 99.9 percent. An executive may want to see only the average and be notified if the trend goes below goal. At the department level, reporting requirements may include information about each event that threatens but does not breach the goal, so that the department may conduct root-cause analysis.

Executive reports tend to summarize and use red, yellow, and green colors to visually identify metrics that are not meeting expectations. Division- or system-level reports usually have more detail and often use trend information to help proactively monitor changing conditions.

Task 5. Vary Focus over Time

Implementing lasting change is difficult, and this challenge holds true for CMDB projects as well. One key concept you can apply to help make change “stick” is to purposefully focus on the types of measures that are most important soon after you move the CMDB into production. Focusing on the control and quality measures early in the four-part scorecard will help you validate that the process is being followed and that the underlying data used for measures is high quality and reliable. Once this has been verified with confidence, then the users’ focus can shift to the performance and value metrics.

You can prioritize the importance of various measures using the Kepner-Tregoe methodology. This method weights each measurement on a scale of 10 (for most important) down to 1 (for least important) and then multiplies a weighted ranking against each metric. The data source for the most highly weighted metrics will receive the most scrutiny, so it will be necessary to ensure that these metrics are reliable. Note that as the business requirements change, the weighted rankings of metrics will change.

BE CAREFUL WHAT YOU MEASURE

The metric you choose may influence results. For example, a customer service organization that evaluates staff performance based on the number of calls handled in an hour may be encouraging representatives to give quick answers that don’t necessarily get to the root of the problem. The result? A large number of repeat calls from clients about problems that were not solved on the first call.

Task 6. Confirm with the Business and Implement

The final task in the identification and installation of metrics is to obtain approval of the scorecard and other reports, and to obtain agreement on how the new metrics will be communicated and used by the organization. If you want the organization to use CMDB-related metrics, you

Stage 5. Driving Ongoing Value

should get agreement from various levels of managers that they will use the reports to communicate status and progress to the organization. If management references the CMDB reports, the organization will follow.

Keep in mind that governance requires business visibility and commitment. Reports can have great implications, not least in terms of facilitating continual improvement through visibility and control of information.

The CMDB reports also can be used as a source of information for department or division-wide bonus measures. If people are getting paid based on the data in your report, they will pay especially close attention.

Summary

Identifying and implementing metrics related to your CMDB project is an important step. Linking goals to metrics to targets, and making sure the resulting reports are used by the organization, is a powerful way to ensure that your CMDB project will meet expectations and improve the effectiveness and efficiency of the IT organization. The metrics also form the basis of continual service improvement, as discussed in step 26, “Create a Continual Service Improvement Program.”

Good quality metrics are essential to have the visibility required to control all processes, as well as to really understand what is going on and to identify areas of weakness. Without measuring the right things well, an organization is unable to make the best decisions, to the detriment of quality, cost effectiveness, and business alignment.

Metrics are also used for compliance with external auditors and a number of regulatory requirements, such as Sarbanes-Oxley, HIPAA, and ISO/IEC 20000. When possible, you’ll want to design metrics that will meet multiple regulatory requirements.

Step 26

Create a Continual Service Improvement Program

Goals and Objectives

In this final step, you will set up a continual service improvement (CSI) program to help ensure that project goals are met and that the CMDB remains accurate and relevant.

Overview

The concepts behind a CSI program are deceptively simple: pick key metrics, set targets, systematically identify root causes of problems, implement fixes, and track and communicate progress.

Most organizations are now aware of the benefits of process improvement frameworks, and a significant number have implemented frameworks such as ITIL. These implementations are not always as successful as they could be, however. A key reason for failure to achieve the benefits of implementing ITIL is insufficient focus on CSI. No process will work perfectly as soon as it is implemented. For optimum success, CSI begins with implementation and continues until the service is retired. Be sure to build CSI into every part of the service lifecycle: strategy, design, transition, and operation.

To create your CSI program, use the metrics and targets identified in step 25, “Implement Measures and Metrics.” The key is to focus CSI efforts on just a few metrics that are most important, and keep a top-down focus over an extended period of time until targets on these metrics are reached. Implement a CSI program at the two levels that match the measurement domains identified in the previous step: one to monitor and improve the CMDB data quality, and one to improve the key metrics that drive the benefits and value of the overall CMDB project.

Figure 26.1 shows the tasks for implementing a CSI program. Readers familiar with ITIL V3 may recognize this as the 7-Step Improvement Process.

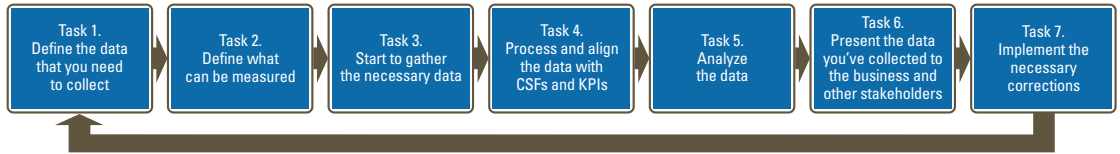


Figure 26.1. The 7-Task Improvement Process

Why Implement Continual Service Improvement?

CSI uses the results of measurement and analysis to prioritize improvements to IT to support changing business needs. CSI is an iterative process, meaning that it is not “once and done,” but happens repeatedly over time, adding more value with each iteration.

The purpose of introducing the processes, roles, and responsibilities of CSI is to maximize the business value and focus of the IT services your organization provides, with a focus on the overall effectiveness of IT service management, the alignment of IT services with the business requirements, and the development of mature IT processes.

CSI helps IT managers clearly define, measure and, subsequently, control the services the IT organization provides to the business. It’s not possible to control a process if you can’t measure it, and measuring it requires that you first define it. When a service is business critical, it’s even more important to focus on it and provide essential improvements.

With IT under increasing pressures from many different parts of the business, some IT managers may be asked why, once a program or hardware is installed and working correctly, their team can’t just step back and let it run. After all, it takes a lot of effort and time to properly implement CSI — resources that the IT team could be using on other important tasks.

But in a dynamic business and IT environment, change is a constant. And having a program that just runs as planned doesn’t take into account the evolving needs of the business. By focusing on how to continually improve each service to address those changing needs, you’ll be providing a range of benefits that improve performance and create good will for the customer or the business, increase the bottom line, allow for innovation, and improve your entire organization. Be aware, however, that just because a change has been implemented, that does not necessarily mean it is optimized.

ENABLING CONTINUAL SERVICE IMPROVEMENT

IT experts Kevin Behr, Gene Kim, and George Spafford discuss key metrics for continual service improvement in change and configuration management.

Based on what we have learned from studying top performing IT organizations, we recommend that you monitor a core set of performance metrics to enable continual service improvement. For change and configuration, consider the following:

- > **Number of changes authorized per week** — How many changes, as measured by the change management process, are implemented? In general, higher is better, as long as the change success rate remains high as well.
- > **Number of actual changes made per week** — How many changes, as measured by detective controls, are implemented? In general, higher is better, but should not be higher than the changes authorized by the Change Advisory Board (CAB).
- > **Number of unauthorized changes** — How many changes circumvent the change process? This is typically measured by using the detective controls, or worse, through unplanned outages. Lower is better.
- > **Change success rate** — How many changes are successfully implemented, without causing an outage or episode of unplanned work? Higher is better: Best-in-class organizations achieve better than 99 percent.
- > **Number of service-affecting outages** — How many changes result in service impairment or an outage? Lower is better.
- > **Number of emergency changes** — How many changes require using the CAB emergency change process? Lower is typically better, since it indicates a higher percentage of planned work.
- > **Number of “special” changes** — How many changes, for whatever reason, are made outside of the change process? Lower is better, because these indicate that a change process is not fully functional and that management is allowing certain categories of changes to bypass change management entirely.
- > **Number of “business as usual” changes** — How many low-impact changes occur? This metric reflects the number of changes that have been identified as “standard” and do not require review. This metric also reflects the number of changes that can pass through without requiring change management scrutiny, but are still logged. In general, higher is better.
- > **Change management overhead** — How much effort (in hours or staffing) is the change management function consuming? In general, this number should be low. A high number may indicate a bureaucratic process, rather than one that enables productivity.
- > **Changes submitted versus changes reviewed** — What is the ratio of evaluated change requests against the total change requests turned in? A lower number is better.

Excerpt from “CMDB: The Key to Jump-Starting ITIL Success,” published in VIEWPOINT, Focus on: CMDB Leadership, published by BMC Software.

Task 1: Define the data that you need to collect.

The first task includes discussing the essential measurements with the customer, the business (if different), and IT management. Decide what data is most important to collect in order to assess progress toward the business goals.

To effectively determine what to measure, you'll need an understanding of the vision, mission, goals, and objectives for both the company as a whole and IT. The bottom line is, what is the company's business, and how is IT contributing to its success? What metrics will help you determine how IT is supporting the business?

Make sure, as you plan improvements, that you know what the business actually needs or wants. By keeping the lines of communication open and listening to your customers' needs, you will be able to create a targeted service improvement program (SIP). You will also be able to determine what data matters most and from what data you will likely get the most value.

Task 2: Define what can be measured.

How will you know whether the data you want to collect can actually be collected? First, gather the measurement tools at your disposal, such as tools for monitoring, reporting, investigating, and service management. Next, analyze the gap between what the customer wants you to measure and what you are currently able to measure. It's possible that not everything the customer wants to measure can be measured. If that's the case, do not include that data category in an SLA.

As you define what you can measure, use that information to create target objectives. These must be phrased in SMART statements, meaning they are specific, measurable, achievable, realistic, and time-based.

Develop a scorecard to track the specific measures and metrics for each domain. For an example of a balanced scorecard, see the four-part scorecard discussed in step 25 and depicted in figure 25.3. Another example, Kaplan and Norton's balanced scorecard, also has four quadrants: Financial Goals, Customer Goals, Innovation Goals, and Internal Goals. Kaplan and Norton's balanced scorecard provides a more comprehensive measurement than looking at financial profitability alone. The balanced scorecard is not used exclusively by IT, but is often used throughout the business.

Keep in mind that CMDB metrics can be used to measure the maturity of other processes. The CMDB provides the hub that all other processes rely on.

Task 3: Start to gather the necessary data.

If you're going to collect data on IT processes, you will also need to monitor for CSI. Whether this is accomplished with tools or manually, it's important to focus on quality. Look for opportunities for improvement, including preventing problems from occurring in the future, reducing costs, and ways to provide better service.

In general, you'll direct your CSI efforts toward only some of the vast quantities of data that will be collected during your monitoring activities. You may need to collaborate with the business to determine what data is most important, or to determine what is possible to monitor with current technology.

Be sure to include in your service requirements any monitoring that CSI will need. Note that CSI may monitor different aspects of a service at different times and that having an agreed-upon process established in advance will facilitate a smooth transition.

You'll also want to create CSI policies for gathering data. For example, "Data must be gathered for reports and reviews on a consistent basis, rather than ad hoc, and used in defined ways."

Task 4: Process and align the data with CSFs and KPIs.

After you collect raw data using the tool(s) you've selected, make sure that it is aligned with the CSFs and KPIs. Processing data helps clarify relationships between the components, including the impact of each component on the rest of the infrastructure as well as on IT in general. It's wise to package this information from the customer's perspective, showing the process from one end to the other.

You'll need to decide how frequently to process the data, in what format the output should be, what tools or systems are required, and how you will evaluate the data's accuracy once it has been processed.

When data is processed by technology, the output may be in report form or a spreadsheet. The resulting information is then more readily turned into knowledge when it's analyzed in task 5. Be aware that manually gathered data is subject to more errors than automatically gathered data; staff members should follow strict procedures and guidelines in order to avoid errors. Organizations should consider automating the data-gathering process wherever possible and practical.

Task 5: Analyze the data.

When you analyze the information you get from processed data, you transform it into knowledge. This job requires a greater level of expertise and experience than that needed for monitoring and data collection or data processing. Data analysts must assess whether the goals and objectives of the process are being met and whether any value has been added through the process.

By interpreting the metrics, you will be able to use them in task 6 to validate and justify prior activities, to intervene in current activities, or to direct future activities.

Task 6: Present the data you've collected to the business and other stakeholders.

Target audiences generally fall into one of these three categories: the business, senior IT management, and internal management.

The business is most interested in questions such as: Has IT delivered as promised? Do they need to make corrections? Are they doing so? Senior IT management wants to know: How close have we come to meeting our plan? Are customers satisfied with the service they are receiving? How close have we come to the CSFs and KPIs? Did we hit or miss our targets for costs and revenue? Internal IT asks: How will the KPIs help me plan for improvement? What activity metrics can I use to schedule improvements?

Be sure to report the good along with the bad. If the service has helped the business, don't forget to highlight those improvements. If a gap remains between the improvements made and those desired, then be sure to report on any plans or measures taken to adjust them. The point is to be able to prove how IT has helped the business.

Reporting takes time and resources. Do not underestimate how much you'll need for either of them. If a report does not contain information that is useful to your target audience, think about discontinuing it.

Develop a business case for future activities based on the results of prior and current activities. Determine the return on investment (ROI) and value on investment (VOI) so far, as well as the anticipated ROI and VOI of future improvements. Business value measures include time to market, customer retention, and market share. Value measures for IT include gaining agility, managing knowledge, enhancing knowledge, reducing costs, and reducing risk. Not all identified possible improvements can be implemented due to lack of budget, time, staff, etc. Decide what will create the most value for the business.

Task 7: Implement the necessary corrections.

Even a well-run process is unlikely to be perfect every time. When you notice a discrepancy between the ideal and actual states, take corrective action as quickly as possible. Be sure to inform all stakeholders of the discrepancy as well as what you are doing to fix the problem and improve service.

You may not be able to correct everything that goes awry. Prioritize the changes that are needed, then make those changes that are in line with the organizational priorities and/or are needed for compliance with regulatory requirements. Then consider additional factors (internal politics, competitive pressures, etc.) and make other changes as time and resources permit.

It's helpful to implement best practices such as ITIL V3. Other best practice frameworks that support CSI include COBIT, Six Sigma, and Capability Maturity Model® Integration (CMMI). The international standard for IT Service Management, ISO/IEC 20000, requires organizations to demonstrate continual service improvement in order to obtain and to retain certification. It may be useful for an organization to consider certification in ISO/IEC 20000 in order to provide an external incentive or mandate to facilitate the required cultural and organizational changes.

Once you've decided to make a correction and improve a service, the process goes through the service lifecycle once again.

Summary

Be sure to follow up your hard work of building a useful CMDB with a CSI program. If you don't, you risk that the CMDB will not have high-quality data, and you are gambling that the organization may not be able to achieve the business benefits that everyone expects. Don't take this risk. Implement a CSI program to ensure that you meet targets for key metrics required for the ongoing success of your CMDB.

Epilogue

Congratulations for making it to the end! Building a CMDB is a significant undertaking and successfully completing it is a major accomplishment.

Now that you've read and applied the *Step-by-Step Guide to Building a CMDB*, we'd love to hear from you. How did your CMDB project go? Do you have lessons learned that you would like to share? Were there additional steps that you needed to include in your CMDB project, or steps that are included in this book that weren't necessary for your project? Please send your comments and feedback to the publisher of this book, BMC Software, at strategic_publications@bmc.com.

If you haven't yet started and would like to further your CMDB education first, be sure to check out the BMC Business School and their educational programs related to the CMDB at www.bmc.com/education.

Best regards,

BMC Software

Step-by-Step Guide to Building a CMDB

About BMC Software

BMC Software delivers the solutions IT needs to increase business value through better management of technology and IT processes. Our industry-leading Business Service Management solutions help you reduce cost, lower risk of business disruption, and benefit from an IT infrastructure built to support business growth and flexibility. Only BMC provides best-practice IT processes, automated technology management, and award-winning BMC Atrium technologies that offer a shared view into how IT services support business priorities. Known for enterprise solutions that span mainframe, distributed systems, and end-user devices, BMC also delivers solutions that address the unique challenges of the mid-sized business. Founded in 1980, BMC has offices worldwide and fiscal 2008 revenues of \$1.73 billion. Activate your business with the power of IT. www.bmc.com

