

11.5 Zusammenspiel mit Incident, Problem, Change und Monitoring

“ Kurz erklärt

Service Configuration Management arbeitet eng mit vielen anderen ITIL Practices zusammen.

Die gespeicherten Informationen über Configuration Items (CIs), Services und deren Beziehungen bilden die Grundlage für fundierte Entscheidungen im IT-Betrieb.

Erst durch das Zusammenspiel mit Incident Management, Problem Management, Change Enablement und Monitoring and Event Management entfaltet Service Configuration Management seinen vollen Nutzen.

Warum das Zusammenspiel wichtig ist

Eine CMDB allein behebt keine Störungen und führt keine Änderungen durch.

Sie beantwortet jedoch entscheidende Fragen wie:

- Welche Systeme gehören zu einem Service?
- Welche Komponenten sind betroffen?
- Wer ist verantwortlich?
- Welche Auswirkungen hat ein Ausfall?
- Welche Änderungen betreffen weitere Systeme?

Diese Informationen werden anschließend von anderen ITIL Practices genutzt.

Zusammenspiel der Practices

Configuration Items



CMDB / CMS



Incident

Problem

Change

Monitoring

|



Bessere Entscheidungen

Die CMDB dient als gemeinsame Informationsbasis.

Service Configuration Management und Incident Management

Bei einem Incident muss häufig schnell geklärt werden:

- Welcher Service ist betroffen?
- Welche Systeme gehören dazu?
- Welche Abhängigkeiten bestehen?
- Wer ist verantwortlich?

Diese Informationen stammen aus der CMDB.

Beispiel

Das Monitoring meldet:

“ Webserver nicht erreichbar.

Die CMDB zeigt:

- gehört zum Kundenportal,
- verwendet Datenbank DB-01,
- läuft auf VM-Cluster A,
- Service Owner: E-Commerce.

Dadurch kann der Incident schneller bewertet werden.

Impact Analysis

Die Auswirkungen einer Störung lassen sich mithilfe der Beziehungen zwischen Configuration Items bestimmen.

Beispiel:

Storage



Datenbank



ERP-System



Produktion

Fällt das Storage-System aus,

sind mehrere Services betroffen.

Dadurch steigt die Priorität des Incidents.

Root Cause Analysis

Auch Problem Management nutzt die Beziehungen der CMDB.

Beispiel:

Mehrere Incidents betreffen:

- Webserver,
- API,
- Mobile App.

Die CMDB zeigt,

dass alle dieselbe Datenbank verwenden.

Dadurch lässt sich die eigentliche Ursache schneller identifizieren.

Service Configuration Management und Problem Management

Problem Management untersucht wiederkehrende Ursachen.

Configuration-Daten helfen beispielsweise dabei,

- betroffene Configuration Items zu identifizieren,
- gemeinsame Abhängigkeiten zu erkennen,

- Änderungen nachzuvollziehen,
- ähnliche Systeme zu vergleichen.

Je vollständiger die CMDB,

desto effizienter wird die Ursachenanalyse.

Known Errors

Wird eine Ursache dauerhaft dokumentiert,

kann sie mit betroffenen Configuration Items verknüpft werden.

Beispiel:



Dadurch lassen sich ähnliche Vorfälle schneller erkennen.

Service Configuration Management und Change Enablement

Changes verändern häufig Configuration Items.

Beispiele:

- neuer Server,
- neue IP-Adresse,
- Firmware-Update,
- zusätzliche Firewall,
- neue Datenbank,
- neue Cloud-Ressource.

Nach jedem erfolgreichen Change sollte die CMDB aktualisiert werden.

Vor einem Change

Vor einer Änderung beantwortet die CMDB beispielsweise:

- Welche Services sind betroffen?
- Welche Systeme hängen davon ab?
- Gibt es Redundanzen?
- Wer muss informiert werden?
- Welche Wartungsfenster sind geeignet?

Dadurch sinkt das Risiko von Fehlentscheidungen.

Nach einem Change

Nach erfolgreicher Umsetzung werden beispielsweise aktualisiert:

- Versionen,
- Beziehungen,
- Status,
- Verantwortliche,
- Dokumentationen,
- Service Maps.

Dadurch bleibt die CMDB aktuell.

Service Configuration Management und Monitoring

Monitoring überwacht Configuration Items.

Die CMDB beschreibt sie.

Beispiel:

Monitoring erkennt:

Server ausgefallen.

Die CMDB ergänzt:

- Servicezuordnung,
- Kritikalität,
- Verantwortlicher,
- Beziehungen,
- Dokumentation.

Erst beide Informationen zusammen ermöglichen eine vollständige Bewertung.

Service Maps

Monitoring und CMDB können gemeinsam Service Maps erzeugen.

Beispiel:

CRM-Service

|

├─ Webserver

├─ API

├─ Datenbank

├─ Storage

└─ Firewall

Fällt eine Komponente aus,
werden betroffene Services sofort sichtbar.

Monitoring nach Changes

Nach einem Change überprüft Monitoring beispielsweise:

- Erreichbarkeit,
- Antwortzeit,
- CPU,
- Speicher,
- Fehler,
- Alarme.

Configuration Management dokumentiert,
welche Version jetzt produktiv eingesetzt wird.

Monitoring unterstützt Discovery

Discovery-Werkzeuge erkennen:

- neue Server,
- virtuelle Maschinen,
- Container,
- Cloud-Ressourcen,
- Betriebssystemversionen.

Die Ergebnisse können anschließend in die CMDB übernommen werden.

Service Configuration Management und Service Level Management

SLA-Ziele beziehen sich häufig auf Services.

Die CMDB beschreibt,

welche Configuration Items zu einem Service gehören.

Dadurch können Verfügbarkeiten und Ausfälle korrekt bewertet werden.

Service Configuration Management und Availability Management

Availability Management benötigt Informationen darüber,

welche Komponenten für einen Service notwendig sind.

Beispiele:

- Server,
- Load Balancer,
- Datenbank,
- DNS,
- Firewall,
- Storage.

Die CMDB liefert diese Informationen.

Service Configuration Management und Capacity Management

Kapazitätsplanung nutzt unter anderem:

- Anzahl Server,
- Speicher,
- CPU,
- Datenbanken,
- Cluster,
- Cloud-Ressourcen.

Die CMDB zeigt,

welche Ressourcen vorhanden sind und welchen Services sie zugeordnet sind.

Service Configuration Management und Information Security Management

Viele sicherheitsrelevante Informationen werden mit Configuration Items verknüpft.

Beispiele:

- Betriebssystemversion,
- Patchstand,
- Firewall,
- Zertifikate,
- Verantwortliche,
- Sicherheitsklassifizierung.

Dadurch lassen sich Risiken gezielter bewerten.

Service Configuration Management und Continual Improvement

Eine gepflegte CMDB unterstützt Verbesserungen.

Beispiele:

- fehlende Beziehungen erkennen,
- Discovery erweitern,
- Datenqualität verbessern,
- Service Maps ergänzen,
- Verantwortlichkeiten aktualisieren.

Dadurch steigt der Nutzen der gesamten Configuration-Datenbasis.

Praxisbeispiel

Ein Switch wird ersetzt.

Vor dem Change zeigt die CMDB:

Betroffen sind:

- ERP,
- VoIP,
- WLAN,
- Druckserver,
- VPN.

Nach erfolgreicher Umsetzung werden:

- Seriennummer,
- Firmware,
- Beziehungen,
- Dokumentationen,
- Status

aktualisiert.

Monitoring bestätigt anschließend,

dass alle Services wieder ordnungsgemäß funktionieren.

Typische Fehler

Fehler 1

Changes werden nicht in der CMDB dokumentiert.

Fehler 2

Incident Management nutzt die CMDB nicht.

Fehler 3

Monitoring kennt keine Servicebeziehungen.

Fehler 4

Discovery-Ergebnisse werden ignoriert.

Fehler 5

Service Maps fehlen.

Fehler 6

Configuration Items besitzen keine Verantwortlichen.

Fehler 7

Problem Management nutzt historische Configuration-Daten nicht.

Fehler 8

Mehrere Systeme besitzen widersprüchliche Informationen.

Fehler 9

SLA-Bewertungen berücksichtigen keine Serviceabhängigkeiten.

Fehler 10

Configuration-Daten werden nur nach größeren Projekten aktualisiert.

Checkliste Zusammenspiel

- ☐ Configuration Items aktuell
 - ☐ Beziehungen gepflegt
 - ☐ Incident Management nutzt die CMDB
 - ☐ Changes dokumentiert
 - ☐ Monitoring integriert
 - ☐ Discovery aktiv
 - ☐ Service Maps aktuell
 - ☐ Verantwortlichkeiten gepflegt
 - ☐ Dokumentationen verknüpft
 - ☐ Datenqualität regelmäßig überprüft
-

Checkliste Service Configuration Management

- ☐ CMDB aktuell
 - ☐ CMS integriert
 - ☐ Discovery eingerichtet
 - ☐ Monitoring angebunden
 - ☐ Incident-Verknüpfungen vorhanden
 - ☐ Change-Prozess berücksichtigt
 - ☐ Problem Records verknüpft
 - ☐ Servicezuordnungen vollständig
-

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker pflegen und nutzen Configuration-Daten nahezu täglich.

Typische Aufgaben:

- neue Systeme dokumentieren,
- Configuration Items aktualisieren,

- Beziehungen ergänzen,
- Monitoring integrieren,
- Discovery prüfen,
- Service Maps pflegen,
- Auswirkungen von Changes bewerten,
- Incidents schneller analysieren.

Eine gepflegte CMDB reduziert Suchaufwand, verbessert Entscheidungen und unterstützt nahezu alle ITIL Practices.

Zusammenfassung

“ Configuration Items verwalten



Beziehungen dokumentieren



Informationen in der CMDB bereitstellen



Incident, Problem und Change unterstützen



Monitoring integrieren



Datenqualität verbessern



Services sicher betreiben

Merksätze

Service Configuration Management liefert Informationen – andere Practices nutzen sie.

“ Eine aktuelle CMDB beschleunigt Incident-, Problem- und Change-Prozesse.

“ Monitoring beschreibt den Zustand – die CMDB beschreibt die Struktur.

“ Service Maps erleichtern Impact- und Ursachenanalysen.

“ Gute Configuration-Daten sind eine wichtige Grundlage für professionelles IT-Service-Management.

Verwandte Seiten

- 11.1 Ziele und Grundlagen des Service Configuration Management
- 11.2 Configuration Items, Attribute und Beziehungen
- 11.3 CMDB und Configuration Management System (CMS)
- 11.4 Pflege, Discovery und Datenqualität
- Incident Management
- Problem Management
- Change Enablement
- Monitoring and Event Management
- Availability Management
- Capacity and Performance Management
- Continual Improvement

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Configuration Management
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Problem Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- PeopleCert – ITIL Practice Guide: Monitoring and Event Management

- ITIL Foundation – Version 5

Einordnung

Die dargestellten Zusammenhänge entsprechen den Empfehlungen der ITIL-Practices. Service Configuration Management stellt strukturierte Informationen über Services und Configuration Items bereit, die von Incident Management, Problem Management, Change Enablement, Monitoring und weiteren Practices genutzt werden. Die konkrete technische Umsetzung erfolgt organisationsabhängig mithilfe von CMDB-, CMS-, Monitoring- und ITSM-Systemen.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
