

11.1 Ziele und Grundlagen des Service Configuration Management

“ Kurz erklärt

Service Configuration Management sorgt dafür, dass Informationen über IT-Services und deren Bestandteile vollständig, aktuell und nachvollziehbar verwaltet werden.

Dazu gehören unter anderem Server, Clients, Netzwerkgeräte, Anwendungen, Cloud-Ressourcen, Dokumentationen und ihre gegenseitigen Beziehungen.

Ziel ist es, jederzeit zu wissen, **welche Komponenten existieren, wie sie zusammenhängen und welche Auswirkungen Änderungen oder Störungen haben.**

Was ist Service Configuration Management?

Service Configuration Management ist eine ITIL Practice zur Verwaltung von Informationen über Services und deren Bestandteile.

Dabei werden nicht die Geräte selbst verwaltet,

sondern die Informationen über diese Objekte.

Beispiele:

- Server
- Clients
- Notebooks
- Switches
- Router
- Firewalls
- Drucker
- virtuelle Maschinen
- Container
- Cloud-Ressourcen
- Anwendungen
- Datenbanken
- Zertifikate
- Dokumentationen

Alle diese Objekte können Bestandteil eines IT-Services sein.

Warum Service Configuration Management wichtig ist

In modernen IT-Umgebungen existieren oft tausende oder sogar zehntausende Komponenten.

Ohne eine strukturierte Verwaltung entstehen schnell Fragen wie:

- Welcher Server gehört zu welchem Service?
- Welche Datenbank nutzt diese Anwendung?
- Welche Firewall schützt diesen Dienst?
- Welche Systeme sind von einem Ausfall betroffen?
- Welche Änderungen beeinflussen diesen Service?

Service Configuration Management beantwortet genau diese Fragen.

Ziele des Service Configuration Management

Die wichtigsten Ziele sind:

- aktuelle Informationen bereitstellen,
 - Beziehungen zwischen Komponenten dokumentieren,
 - Auswirkungen von Änderungen erkennen,
 - Incident Management unterstützen,
 - Problem Management unterstützen,
 - Change Enablement unterstützen,
 - Risiken reduzieren,
 - Transparenz schaffen,
 - Entscheidungsgrundlagen liefern.
-

Was ist ein Configuration Item (CI)?

Ein Configuration Item (CI) ist jedes Objekt,

das für die Bereitstellung oder Verwaltung eines IT-Services relevant ist.

Ein CI kann beispielsweise sein:

- Server
- Switch
- Router
- Firewall
- Notebook
- Datenbank
- Software

- Container
- virtuelle Maschine
- Cloud-Service
- Dokumentation
- SLA
- Zertifikat

Nicht jedes Inventarobjekt muss automatisch ein CI sein.

Entscheidend ist die Bedeutung für den Service.

Beispiele für Configuration Items

Kategorie	Beispiele
Hardware	Server, Notebook, Switch
Software	Betriebssystem, ERP, Datenbank
Netzwerk	Router, Firewall, VLAN
Cloud	virtuelle Maschine, Storage, Load Balancer
Dokumentation	Runbook, Architekturdiagramm
Service	E-Mail, VPN, Fileserver
Verträge	SLA, Wartungsvertrag

Nicht jedes Objekt ist ein CI

Ein einzelnes Netzkabel oder eine Computermouse wird häufig inventarisiert, aber nicht unbedingt als Configuration Item verwaltet.

Dagegen besitzen folgende Objekte meist einen direkten Einfluss auf Services:

- Server
- Netzwerkgeräte
- Anwendungen
- virtuelle Maschinen
- Cloud-Ressourcen
- Datenbanken

Diese werden häufig als CIs geführt.

Was wird über ein CI gespeichert?

Je nach Organisation können unterschiedliche Informationen gespeichert werden.

Typische Angaben:

- Name
- eindeutige ID
- Typ
- Hersteller
- Modell
- Seriennummer
- Standort
- Besitzer
- Verantwortlicher
- aktueller Status
- Version
- Beziehungen
- Dokumentation

Je kritischer das CI,

desto umfangreicher sind häufig die gespeicherten Informationen.

Der Lebenszyklus eines Configuration Items

Ein CI durchläuft mehrere Phasen.

Beispiel:



Außer Betrieb



Entsorgt

Alle relevanten Änderungen sollten dokumentiert werden.

Status eines Configuration Items

Typische Status:

- geplant
- bestellt
- installiert
- produktiv
- in Wartung
- außer Betrieb
- archiviert
- entsorgt

Der aktuelle Status erleichtert den Überblick.

Beziehungen zwischen Configuration Items

Ein großer Vorteil des Service Configuration Managements besteht darin,

dass Beziehungen dokumentiert werden.

Beispiel:

Online-Shop



Webserver



Betriebssystem



Datenbank



Storage



Firewall

Dadurch wird sichtbar,
welche Komponenten voneinander abhängig sind.

Warum Beziehungen wichtig sind

Fällt eine Datenbank aus,
kann dies Auswirkungen haben auf:

- Webserver,
- ERP,
- Online-Shop,
- Reporting,
- mobile Anwendungen.

Durch dokumentierte Beziehungen lassen sich Auswirkungen deutlich schneller erkennen.

Service statt Einzelgerät

ITIL betrachtet nicht einzelne Geräte,
sondern den gesamten Service.

Ein Benutzer interessiert sich beispielsweise nicht dafür,
welcher Server ausgefallen ist.

Wichtig ist für ihn,
ob der E-Mail-Service funktioniert.

Service Configuration Management verbindet technische Komponenten mit den bereitgestellten Services.

Service Maps

Service Maps zeigen grafisch,
welche Configuration Items zu einem Service gehören.

Beispiel:

VPN-Service

- |
- └─ Firewall
- └─ VPN-Gateway
- └─ Active Directory
- └─ DNS
- └─ Zertifikate

Dadurch lassen sich Störungen und Änderungen schneller bewerten.

Warum aktuelle Daten wichtig sind

Veraltete Informationen führen häufig zu:

- falschen Entscheidungen,
- unnötigen Incidents,
- fehlerhaften Changes,
- längeren Ausfallzeiten,
- unvollständigen Analysen.

Deshalb müssen Configuration-Daten regelmäßig gepflegt werden.

Praxisbeispiel

Ein Webserver fällt aus.

Durch Service Configuration Management erkennt die IT sofort,

dass betroffen sind:

- Kundenportal,
- Zahlungsdienst,
- API,
- Mobile App.

Dadurch kann die Priorität schneller bestimmt und der Incident gezielt bearbeitet werden.

Typische Fehler

Fehler 1

Configuration Items sind nicht aktuell.

Fehler 2

Beziehungen fehlen.

Fehler 3

Services sind nicht dokumentiert.

Fehler 4

Änderungen werden nicht nachgetragen.

Fehler 5

Verantwortlichkeiten sind unklar.

Fehler 6

Zu viele unwichtige Objekte werden als CIs verwaltet.

Fehler 7

Service Maps fehlen.

Fehler 8

Configuration Management wird nur als Inventarliste genutzt.

Checkliste Configuration Items

- ☐ eindeutige CI-ID vorhanden
 - ☐ aktueller Status gepflegt
 - ☐ Verantwortlicher dokumentiert
 - ☐ Beziehungen hinterlegt
 - ☐ Servicezuordnung vorhanden
 - ☐ Änderungen nachgeführt
 - ☐ Dokumentation aktuell
 - ☐ regelmäßige Überprüfung geplant
-

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker arbeiten regelmäßig mit Configuration Items.

Typische Aufgaben:

- neue Systeme dokumentieren,
- Beziehungen pflegen,
- Service Maps aktualisieren,
- Configuration-Daten nach Changes anpassen,
- Incidents anhand der CI-Beziehungen analysieren,
- Dokumentationen aktualisieren.

Eine gepflegte Configuration-Datenbasis erleichtert nahezu alle Bereiche des IT-Service-Managements.

Zusammenfassung

“ Service identifizieren

↓

Configuration Items erfassen

↓

Beziehungen dokumentieren

↓

Status aktuell halten

↓

Änderungen nachführen

↓

Informationen für Incident, Problem und Change Management bereitstellen

Merksätze

Service Configuration Management verwaltet Informationen – nicht die Hardware selbst.

“ Ein Configuration Item ist jedes für einen Service relevante Objekt.

“ Beziehungen zwischen Configuration Items sind oft wichtiger als die einzelnen Komponenten.

“ Aktuelle Configuration-Daten beschleunigen Incident- und Change-Prozesse.

“ Gute Service Maps erhöhen die Transparenz komplexer IT-Umgebungen.

Verwandte Seiten

- 11.2 Configuration Items, Attribute und Beziehungen
- 11.3 CMDB und Configuration Management System (CMS)
- 11.4 Pflege, Discovery und Datenqualität
- 11.5 Zusammenspiel mit Incident, Problem, Change und Monitoring
- Incident Management
- Change Enablement
- Monitoring and Event Management

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Configuration Management
- ITIL Foundation – Version 5

Einordnung

Die dargestellten Beispiele und Vorgehensweisen orientieren sich an der ITIL Practice „Service Configuration Management“. Welche Objekte als Configuration Items geführt werden und welche Informationen gespeichert werden, richtet sich nach Größe, Komplexität und Anforderungen der jeweiligen Organisation.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
