



WHZ Westsächsische
Hochschule Zwickau
Hochschule für Mobilität

Diplomarbeit

Betrachtung von Methoden zum IT-Servicemanagement für IT-Dienste zur Optimierung von Dokumentation, Monitoring und Dependency Mapping

eingereicht von:

Hannes Braun

geb. am 06.10.1992

zur Erlangung des akademischen Grades des
Diplom-Informatikers (FH)

Erstgutachter:

Prof. Dr. Tim Neumann

Professur für Wirtschaftsinformatik,
Westsächsische Hochschule Zwickau

Zweitgutachter:

Prof. Sven Hellbach

Professur für Informatik/Intelligente Systeme
Westsächsische Hochschule Zwickau

Dresden, den 03.12.2025

Vorwort

Die informationstechnologische Landschaft der Sächsischen Landesbibliothek - Staats- und Universitätsbibliothek Dresden (SLUB) umfasst eine Vielzahl von Anwendungen und Systemen, deren Management eine erhebliche Herausforderung darstellt. Hierunter zählen neben größeren Projekten, wie etwa der deutschen Fotothek und dem arthistoricum, auch eine Vielzahl kleinerer Projekte. Diese werden von einem Team von 48 Mitarbeitern in der IT-Abteilung betrieben. Diese Komplexität wird durch fehlende Automatisierung in Bereichen wie Monitoring, Dokumentation und Management verstärkt, was zu substanziellen manuellen Arbeitsleistungen führt. Versuche, Automatisierung zu implementieren, stoßen oft an Grenzen, da die dafür notwendigen Informationsgrundlagen fehlen oder unzuverlässige Ergebnisse liefern. Diese Informationsdefizite sind auf Mängel im IT-Servicemanagement (ITSM) zurückzuführen. Im Fokus dieser Diplomarbeit stehen daher die Analyse des ITSM im Kontext der SLUB und die Entwicklung eines entsprechenden Konzepts zur Behebung dieser Informationslücken. Weiterhin werden Möglichkeiten zur Synchronisation dieser Informationen zwischen Dokumentations- und Monitoring-Systemen untersucht.

Inhaltsverzeichnis

Inhaltsverzeichnis	V
Abbildungsverzeichnis	IX
Tabellenverzeichnis	XI
Abkürzungsverzeichnis	XIII
Symbolverzeichnis	XV
Hilfsmittelverzeichnis	XVII
1 Einleitung	1
1.1 Thema	1
1.2 Ziele	2
1.3 Methodik	2
1.4 Aufbau	3
2 Theoretische Grundlagen	5
2.1 IT Service Management	5
2.2 Services im Detail	6
2.3 ITIL	6
2.4 Configuration Management	7
2.5 Configuration Items und deren Beziehungen	7
2.6 Configuration Management Database	8
2.7 Configuration Management System	8
2.8 IT Asset Management	9
2.9 Bezug auf DevOps und agile IT-Organisation	9
2.10 Zusammenfassung	9
2.11 Anpassung in der Nomenklatur	10

3	Analyse der bestehenden IT-Landschaft	11
3.1	Bestehende Systeme zur Dokumentation	11
3.2	Bestehendes ITSM-Konzept	14
3.3	Zusammenwirken der bestehenden Systeme	15
3.4	Probleme mit dem bisherigen ITSM-Konzept	16
3.5	Vorgehaltene Daten	18
3.5.1	CMDB	18
3.5.2	Weitere Hilfssysteme	22
3.6	Fazit	22
4	Konzeptionierung	23
4.1	Befragung zum Umgang mit CMDBs und ITSM	23
4.1.1	Aufbau	23
4.1.2	Auswertung	24
4.1.3	Fazit	26
4.2	Modellierung der CMDB	27
4.2.1	Klassen	27
4.2.2	Kategorien und Attribute	29
4.2.3	Beziehungsarten	32
4.2.4	Verantwortungsgrenzen	33
4.3	Bereitstellungsumgebungen	36
4.4	Verschränkung der Hilfssysteme	36
4.4.1	System-Logik	36
4.4.2	Problem	37
4.4.3	Angestrebte Lösung	37
5	Umsetzung	39
5.1	Pflege der CMDB	39
5.2	Verschränkung der Systeme	41
5.3	Fazit	44
6	Fazit	45
6.1	Ergebnisse	45
6.2	Ausblick	46
A	Verwendung von LLMs zur Formulierung	49
A.1	Eingabe	49
A.2	Ausgabe	63

Literatur	71
Selbständigkeitserklärung	73

Abbildungsverzeichnis

4.1	Klassendiagramm: <i>Meta</i> -Klasse und Abhängigkeiten	30
4.2	Klassendiagramm: Business und Technical Services	32
4.3	Beziehungsarten: Service-Komponente und <i>Inhalt / Logisch</i>	33
4.4	Klassendiagramm: Service Domain	35
4.5	Schwachstellenscanner-Logik	36
4.6	Sequenzdiagramm: Anlage Scans im Schwachstellenscanner	38
5.1	Abfrage für Hosts, Technical Service und Service Domain	42
5.2	Report-Vorschau für Service Domain-Hosts	43

Tabellenverzeichnis

3.1	CMDB-Klassen und zugehörige Kategorien	19
-----	--	----

Abkürzungsverzeichnis

BS	Business Service
CI	Configuration Item
CM	Configuration Management
CMDB	Configuration Management Database
CMS	Configuration Management System
DBoD	Databases on Demand
DSGVO	Datenschutz-Grundverordnung
GP	Geschäftsprozess
GSM	Greenbone Security Manager
IaC	Infrastructure as Code
IS	Infrastructure Service
ITAM	IT Asset Management
ITIL	IT Infrastructure Library
ITSM	IT Service Managemen
SDN	Software Defined Networking
SLUB	Sächsische Landesbibliothek — Staats- und Universitätsbibliothek Dresden (SLUB)
TS	Technical Service
TSe	Technical Service extern

VE Virtualisierungsumgebung

VM Virtuelle Maschine

Symbolverzeichnis

Hilfsmittelverzeichnis

Overleaf	Webbasierter LaTeX-Editor, zum Verfassen wissenschaftlicher Dokumente.
LanguageTool	Werkzeug zur Rechtschreib- und Grammatikprüfung.
ChatGPT	Zur Erzeugung von Text aus vorher definierten Inhalten. Das Generieren von Inhalten wurde dem Werkzeug mittels entsprechend definierten Eingaben untersagt. Dem Anhang kann exemplarisch die Eingabe und Ausgabe für Kapitel 3 entnommen werden.
Zotero	Werkzeug zur Erschließung und Verwaltung von Zitatquellen.

Kapitel 1

Einleitung

Der Betrieb komplexer IT-Landschaften stellt moderne Organisationen vor erhebliche operative und strategische Herausforderungen. Um den reibungslosen Betrieb solcher Landschaften zu gewährleisten, Herausforderungen effektiv zu bewältigen und die strategischen Ziele der Institution zu unterstützen, ist eine umfassende und aktuelle Dokumentation unerlässlich.

1.1 Thema

Die vorliegende Diplomarbeit thematisiert die Probleme, die sich aus Informationslücken, Inkonsistenzen und fehlerhaften Verknüpfungen in der bestehenden Systemdokumentation der SLUB ergeben. Die aktuelle IT-Landschaft der SLUB ist geprägt von einer Vielzahl von Systemen, die oft in komplexen Abhängigkeiten zueinander stehen. Um den Betrieb, die Überwachung und Dokumentation dieser Systeme zu gestalten, sind fundierte Informationen über ihre Konfiguration, Funktionen und Interaktionen erforderlich. Bestehende Hilfssysteme zur Erfassung und Verwaltung dieser Informationen stoßen jedoch oft an ihre Grenzen, was zu einer Fragmentierung des Wissens und einer erschwerten Fehlerbehebung führt.

Diese Situation wird durch die Komplexität der IT-Landschaft selbst verstärkt, die durch die Integration verschiedener Dienste, Technologien und Schnittstellen gekennzeichnet ist. Die Notwendigkeit einer einheitlichen und konsistenten Dokumentation ergibt sich auch aus den steigenden Anforderungen an die IT-Sicherheit, der Einhaltung von Compliance-Richtlinien und der kontinuierlichen Verbesserung der IT-Prozesse. Die Diplomarbeit konzentriert sich daher auf die Entwicklung eines Konzepts für IT-Service-Management (ITSM) an der SLUB, das die bestehenden Strukturen und Prozesse optimiert und eine solide Grundlage für die kontinuierliche Verbesserung des IT-Betriebs schafft.

Im Rahmen dieser Arbeit werden insbesondere die Bereiche Configuration Management, Incident Management, Problem Management und Change Management betrachtet. Dabei werden bewährte Methoden und Best Practices aus den Bereichen ITIL (Information Technology Infrastructure Library) und DevOps berücksichtigt. Ziel ist es, ein maßgeschneidertes ITSM-Konzept zu entwickeln, das die spezifischen Anforderungen der SLUB berücksichtigt und einen Mehrwert für die Organisation schafft.

1.2 Ziele

Die vorliegende Diplomarbeit verfolgt folgende Hauptziele:

Erstellung eines Konzepts für ITSM an der SLUB

Die Entwicklung eines detaillierten ITSM-Konzepts, das die bestehenden Strukturen und Prozesse optimiert und eine solide Grundlage für die kontinuierliche Verbesserung des IT-Betriebs schafft.

Umsetzung des Konzepts

Pilotimplementierung des ITSM-Konzepts an zwei repräsentativen Systemen der SLUB, dem DBoD und dem Shibboleth-Authentifikationssystem. Ziel ist es, die Praktikabilität des Konzepts zu demonstrieren und Erfahrungen für die Weiterverwendung zu sammeln.

1.3 Methodik

Um die genannten Ziele zu erreichen, kommen verschiedene methodische Ansätze zum Einsatz.

Literaturrecherche

Eine umfassende Literaturrecherche zu den Themen ITSM, ITIL, CMDB, Monitoring und Automatisierung dient als Grundlage für die Konzeption und Bewertung der Lösungsansätze.

Befragung

Eine Befragung mit den IT-Mitarbeitern der SLUB liefern wertvolle Einblicke in die bestehenden Prozesse, Herausforderungen und Anforderungen. Diese Befragung dient dazu, die spezifischen Bedürfnisse der SLUB zu verstehen und die Konzeption des ITSM-Konzepts darauf abzustimmen.

Analyse der bestehenden Systemdokumentation

Die vorhandene Systemdokumentation wird kritisch analysiert, um Schwachstellen und Verbesserungspotenziale zu identifizieren sowie Lücken und Redundanzen in den gepflegten Daten zu ermitteln.

Pilotimplementierung

Die Pilotimplementierung des ITSM-Konzepts an den beiden ausgewählten Systemen ermöglicht eine praktische Überprüfung der entwickelten Lösungsansätze und eine Sammlung von Erfahrungen für die Weiterverwendung.

1.4 Aufbau

Die vorliegende Diplomarbeit nimmt sich der Optimierung des IT-Betriebs an der Sächsischen Landesbibliothek – Staats- und Universitätsbibliothek Dresden (SLUB) an und beginnt mit einer Einleitung, die die Herausforderungen komplexer IT-Landschaften und die spezifische Problemstellung an der SLUB umreißt. Aufbauend darauf wird in Kapitel 2: Theoretische Grundlagen ein fundiertes Verständnis für IT-Service-Management vermittelt. Hier werden bewährte Methoden und Standards wie die ITIL sowie verwandte Themengebiete und etablierte Methoden zur ITSM-Implementierung detailliert erläutert. Um die Ist-Situation der SLUB zu verstehen, folgt Kapitel 3: Analyse der bestehenden IT-Landschaft, welches die aktuelle IT-Infrastruktur, Prozesse und die damit verbundenen Herausforderungen analysiert. Auf dieser Grundlage wird in Kapitel 4: Konzeptionierung ein maßgeschneidertes ITSM-Konzept für die SLUB entwickelt, das konkrete Verbesserungsvorschläge und die Integration der bestehenden Systeme beinhaltet. Dieses Konzept wird anschließend in Kapitel 5: Umsetzung durch eine Pilotimplementierung an den Systemen DBoD und Shibboleth-Authentifizierung praktisch erprobt und mit der Synchronisierung der Daten aus dem CMDB- und Monitoring-System weiter ausgebaut. Abschließend fasst Kapitel 6: Fazit die Ergebnisse zusammen,

bewertet die erreichten Ziele und gibt einen Ausblick für die weitere Entwicklung des ITSM-Konzepts an der SLUB, um den IT-Betrieb nachhaltig zu verbessern.

Kapitel 2

Theoretische Grundlagen

Moderne IT-Landschaften zeichnen sich durch eine hohe Komplexität aus, sowohl hinsichtlich der Vielzahl beteiligter Komponenten als auch der verwendeten Technologien. Ein erfolgreicher und stabiler Betrieb solcher Systeme setzt Transparenz, strukturierte Organisation, kontinuierliche Überwachung und gezielte Steuerung sämtlicher Elemente voraus. Zentrale Konzepte, welche zur Realisierung dieser Anforderungen beitragen, sind das IT-Service Management sowie das Configuration Management (CM). In enger Verbindung hierzu stehen weitere Begriffe und Werkzeuge, wie etwa ITIL als Implementierungs-Framework für ITSM, die Configuration Management Database (CMDB), das Configuration Management System (CMS) und der Servicedesk. Das vorliegende Kapitel widmet sich der Definition, Einordnung und Darstellung dieser Begriffe sowie ihrer wechselseitigen Zusammenhänge. Weiterhin werden angrenzende Disziplinen wie das IT-Asset-Management (ITAM) sowie moderne Praktiken, insbesondere DevOps, betrachtet.

2.1 IT Service Management

ITSM beschreibt die Gesamtheit aller Methoden, Praktiken und Prozesse, die notwendig sind, um IT-gestützte Services zu planen, zu steuern, zu überwachen und letztlich bereitzustellen.¹ Die im ITSM betrachteten Services sind immaterielle Leistungen, die darauf abzielen, einen geschäftlichen Mehrwert für interne oder externe Kunden zu schaffen.² Zu den zentralen Teilgebieten, beziehungsweise Prozessen, des ITSM, zählen unter anderem das Incident Management, Problem Management, Change Management, Release Management und Configuration Management.³

¹Vgl. [IBM21c, Abs. „Was ist ITSM?“] und [ITI19, S. 6].

²Vgl. [ITI19, S. 12].

³Vgl. [IBM21c, Abs. „ITSM-Prozesse und -Verfahren“].

Diese Teilgebiete stehen nicht isoliert, sondern profitieren wechselseitig voneinander. Ein besonders prägnantes Beispiel für die Synergien innerhalb des ITSM ergibt sich im Zusammenhang mit einem gepflegten Configuration Management: Im Fall eines Incidents ermöglicht eine detaillierte Dokumentation der Systemkonfiguration eine schnellere Lokalisierung des Fehlers und damit eine beschleunigte Fehlerdiagnose. Auch im Rahmen des Change Managements lässt sich durch das Configuration Management präzise analysieren, welche Systemkomponenten von einer geplanten Änderung betroffen wären, was zur Minimierung potenzieller Fehlerquellen beiträgt und eine fundierte Risikobewertung erlaubt.

2.2 Services im Detail

Ein Service lässt sich im Kontext von ITSM als eine Dienstleistung beschreiben, die einen spezifischen Wert für einen Kunden generieren soll.⁴ Der durch einen oder mehrere Services erbrachte Wert, dient direkt oder indirekt der Zielerreichung eines Geschäftsprozesses. Kunden können sowohl innerhalb der eigenen Organisation als auch außerhalb verortet sein.

Greift ein Nutzer über IT-Systeme auf einen Service zu, handelt es sich um einen IT-Service. Im weiteren Verlauf dieser Arbeit wird der Begriff „Service“ stellvertretend für IT-Services verwendet. Jeder Service wird durch einen sogenannten Service Provider bereitgestellt, während ein Service Consumer diesen Service nutzt, um daraus einen Wert zu ziehen. Dieser Wert kann entweder in der direkten Verwendung des bereitgestellten Service bestehen, beispielsweise beim Zugriff auf Internet oder Telefonie durch Endnutzer, oder in der Weiterverwertung zur Bereitstellung eines neuen Service, etwa bei der Verwendung eines Datenbank-Service zur Entwicklung einer Webapplikation. ITIL beschreibt sowohl die direkte Nutzung des Wertes eines Services, als auch die Aufbereitung für die Erbringung eines neuen Service als Service Relationship Model.⁵

2.3 ITIL

ITIL ist ein Framework zur Umsetzung von IT Service Management. Es ist das am häufigsten verwendete Framework zur Realisierung von ITSM⁶ und hat sich als De-facto-Standard etabliert. Die aktuell gültige Version, ITIL 4 Edition, bezeichnet das

⁴Vgl. [ITI19, S. 7-9].

⁵Vgl. [ITI19, S. 14-15].

⁶Vgl. [IBM21c, Abs. „ITSM-Frameworks“].

Configuration Management als Service Configuration Management und betont damit explizit die Serviceorientierung dieses Teilbereichs.⁷

Das Ziel von ITIL besteht in der Definition von Best Practices, welche es Organisationen ermöglichen sollen, ein standardkonformes ITSM – einschließlich eines konsistenten Configuration Managements – umzusetzen. Dabei definiert ITIL bewusst keine konkreten Werkzeuge, sondern fungiert als übergeordneter Rahmen. Neben den allgemeinen ITSM-Zielen wie Planung, Steuerung, Überwachung und Betrieb von IT-Services, wird in ITIL ein besonderer Fokus auf die Configuration Management Database gelegt.⁸

2.4 Configuration Management

Das Configuration Management ist ein integraler Bestandteil des ITSM. Es befasst sich mit der systematischen Erfassung, Pflege und Überwachung aller Komponenten und Ressourcen, die zur Erbringung eines IT-Service erforderlich sind. Ziel ist eine vollständige und nachvollziehbare Darstellung der gesamten IT-Infrastruktur, wodurch Transparenz geschaffen, Fehlerquellen reduziert und eine effiziente Prozessgestaltung ermöglicht wird.⁹

Die zugrunde liegende Infrastruktur setzt sich aus Configuration Items (CIs) sowie deren Beziehungen untereinander zusammen. Diese Struktur lässt sich als Graph modellieren, wobei die CIs als Knoten und die zwischen ihnen bestehenden Beziehungen als Kanten fungieren.

2.5 Configuration Items und deren Beziehungen

Configuration Items bilden die elementaren Bausteine des Configuration Managements. Dabei handelt es sich um physische oder logische Einheiten, Dokumente oder auch Services, die zur Erbringung eines bestimmten IT-Service notwendig sind. Typische Beispiele umfassen Server, Netzwerkinfrastrukturen, virtuelle Maschinen, Betriebssysteme, Anwendungen, Datenbanken und Webservices.

Jedes CI verfügt über mindestens einen eindeutigen Bezeichner sowie eine Vielzahl definierter Attribute wie etwa Status oder Version. Die CIs stehen in unterschiedlichen Beziehungstypen zueinander, beispielsweise *läuft auf*, *gehört zu* oder *abhängig von*. Diese Beziehungen sind in der Regel gerichtet, können jedoch bidirektional interpretiert

⁷Vgl. [ITI19, S. 139].

⁸Vgl. [IBM21b, Abs. „Was ist ITIL?“].

⁹Vgl. [IBM24] und [ITI19, S. 139-140].

werden.¹⁰

Das Modell zur Beschreibung der CIs folgt einem objektorientierten Paradigma: CI-Typen entsprechen dabei den Klassen aus der objektorientierten Programmierung, während konkrete CIs den Objekten zugeordnet werden. Die CI-Typen legen fest, welche Attribute für die jeweiligen CIs verfügbar sind.

2.6 Configuration Management Database

Die CMDB ist der zentrale Speicherort für sämtliche im Configuration Management erfassten Informationen. Sie dient sowohl der Vorhaltung als auch der Pflege und Präsentation von CIs, deren Attributen und Beziehungen.¹¹ Neben objekt- und dokumentenbasierten Ansichten bietet die CMDB häufig auch Graph-basierte Visualisierungen mit weitreichenden Filter- und Anzeigeeoptionen.

Besonders im operativen IT-Betrieb ist eine aktuelle und konsistente Datenbasis in der CMDB von essenzieller Bedeutung. Angesichts der hohen Komplexität moderner IT-Infrastrukturen ist eine manuelle Pflege kaum praktikabel und mit erheblichen Risiken hinsichtlich Fehleranfälligkeit und Inkonsistenz verbunden. Daher empfiehlt sich zunehmend automatisierte Mechanismen zum Einsatz zu bringen, welche relevante Informationen identifizieren, verarbeiten und in die CMDB integrieren können.

2.7 Configuration Management System

Ein CMS umfasst alle Werkzeuge, Schnittstellen und Datenbanken, die zur Erfassung, Pflege, Analyse und Darstellung von Konfigurationsinformationen notwendig sind. Es integriert die CMDB als zentralen Datenbankbestandteil und stellt darüber hinaus zusätzliche Funktionen bereit, etwa zur automatisierten Informationsgewinnung aus Monitoring- oder Ticket-Systemen.¹²

Ziel ist es, sämtliche für den Betrieb einer IT-Landschaft relevanten Informationen und Aktivitäten zentral zu bündeln. Das CMS unterstützt die Umsetzung der im ITSM und CM formulierten Ziele, wie etwa Schwachstellen- und Risikoanalyse, Change-Management sowie Ursachenanalyse bei Störungen. Ergänzt wird das CMS durch Präsentationsmöglichkeiten wie Dashboards und Berichtsfunktionen.

¹⁰Vgl. [ITI19, S. 140] und [IBM23].

¹¹Vgl. [IBM21a].

¹²Vgl. [ITI19, S. 141].

2.8 IT Asset Management

ITAM stellt ein weiteres Teilgebiet des ITSM dar und fokussiert sich auf die Verwaltung materieller und immaterieller IT-Ressourcen. Dazu zählen unter anderem Lizenzen, Hardware, Software, Wartungsverträge oder Garantien. Diese sogenannten IT-Assets durchlaufen innerhalb des ITAM ein vollständiges Lifecycle-Management, das von der Beschaffung über den operativen Einsatz bis hin zur Ausmusterung reicht.

IT-Assets werden ebenfalls in der CMDB dokumentiert. Während das ITSM den Fokus auf die Service-Erbringung richtet, konzentriert sich das ITAM auf die Verwaltung der hierfür eingesetzten Ressourcen.¹³

2.9 Bezug auf DevOps und agile IT-Organisation

Im Zuge der fortschreitenden Automatisierung und immer agilerer IT-Organisationen gewinnen Verfahren wie DevOps, Continuous Delivery und Continuous Integration (CI/CD) zunehmend an Bedeutung. Diese Praktiken setzen ein fundiertes und stets aktuelles Configuration Management voraus.

Ein zentrales Konzept ist „Infrastructure as Code“ (IaC). Hierbei werden Infrastrukturen, Abhängigkeiten und Komponenten auf Codebasis definiert und mittels Automatisierungstools wie Ansible, Puppet oder Terraform implementiert und orchestriert. Gleichzeitig führt die zunehmende Dynamik moderner IT-Landschaften zu einer potenziellen Informationslücke im Configuration Management: Neue Informationen stehen in immer kürzeren Intervallen zur Verfügung.

Damit automatisierte Prozesse effizient funktionieren, müssen aktuelle Konfigurationsdaten verfügbar sein. Umgekehrt können solche Prozesse Informationslücken verursachen, wenn sie nicht adäquat rückgekoppelt werden. Daraus ergibt sich die Notwendigkeit von Mechanismen, welche sowohl die kontinuierliche Aktualisierung der Konfigurationsdaten gewährleisten als auch geeignete Schnittstellen zur Datenbereitstellung bieten.

2.10 Zusammenfassung

Die Realisierung betrieblicher Geschäftsziele erfordert die Bereitstellung geeigneter Services. IT-gestützte Services sind hochkomplex und bedürfen einer umfassenden Verwaltung. Die hierfür etablierten Methoden und Praktiken sind im IT Service Management verortet, welches sich in diverse Teilgebiete gliedert.

¹³Vgl. [ITI19, S. 124-127].

Eine zentrale Rolle spielt hierbei das Configuration Management, dessen Informationen in einer Configuration Management Database gespeichert und gepflegt werden. Die CMDB fungiert als zentrales Repositorium und ermöglicht die notwendige Transparenz und Struktur zur effizienten Steuerung der IT-Landschaft.

Configuration Management steht in enger Beziehung zu sämtlichen Teilgebieten des ITSM und ist insbesondere für den operativen Betrieb von kritischer Bedeutung. Daher liegt ein besonderer Fokus dieser Arbeit auf der Ausgestaltung und Umsetzung eines funktionalen Configuration Managements in Verbindung mit einer CMDB.

Um die kontinuierliche Pflege der CMDB sicherzustellen, sind automatisierte Mechanismen unabdingbar. Ebenso bedarf es klarer Verantwortlichkeiten hinsichtlich Services, Teilservices und zugehöriger Komponenten. Der Erfolg eines ITSM-Ansatzes hängt maßgeblich von der Akzeptanz geeigneter Werkzeuge und der unternehmensweiten Anerkennung der Bedeutung von Dokumentation und Informationspflege ab.

2.11 Anpassung in der Nomenklatur

Für den weiteren Verlauf der Arbeit werden einige Begriffe verwendet, die als Alternativen zu in den Bereichen ITSM und CMDB üblichen Bezeichnungen stehen. Die Entscheidung für diese alternativen Begriffe ergibt sich daraus, dass bereits die vorliegende CMDB bestimmte Gegenstände anders betitelt als etwa ITIL. So werden Configuration Items hier beispielsweise schlicht als Objekte bezeichnet; die CI-Typen, auf denen sie beruhen, erscheinen entsprechend als Objekt-Typen. Zudem wird im Verlauf der Arbeit wiederholt auf Klassendiagramme und Grafen Bezug genommen. Vor diesem Hintergrund werden die eingeführten CIs als Objekte und die ihnen zugrunde liegenden CI-Typen als Klassen bezeichnet. Die vorliegende CMDB stellt darüber hinaus einen visuellen Editor bereit, der einen Grafen zur Veranschaulichung der Beziehungen zwischen Objekten erzeugt. Im Kontext dieser Visualisierung wird daher ebenfalls auf die Begriffe Knoten (Objekt) und Kante (Beziehung) zurückgegriffen.

Kapitel 3

Analyse der bestehenden IT-Landschaft

Im folgenden Kapitel wird die bestehende IT-Landschaft der Sächsischen Landesbibliothek – Staats- und Universitätsbibliothek Dresden analysiert. Im Zentrum steht die Betrachtung der vorhandenen Systeme zur Dokumentation und Überwachung, ihrer gespeicherten und verwalteten Daten, der Beziehungen untereinander sowie ihrer Integration in die Gesamtlandschaft der IT-Systeme. Ziel ist es, ein strukturiertes Verständnis der bestehenden Komponenten und ihrer funktionalen sowie organisatorischen Einbettung zu gewinnen.

3.1 Bestehende Systeme zur Dokumentation

ITSM-System

Als integrierte Lösung für das IT Service Management, das Configuration Management sowie für die Verwaltung der Configuration-Management Database kommt an der SLUB die Softwarelösung i-doit zum Einsatz. Diese Lösung stellt den zentralen Knotenpunkt für das ITSM dar. I-doit unterstützt eine Vielzahl an Funktionen, die über klassische ITSM-Aufgaben hinausgehen. Dazu zählen unter anderem das IT Asset Management, das Erstellen von Betriebshandbüchern, das Verwalten von Stammdaten, Automatisierungsfunktionen sowie die Betrachtung von Netzwerkplänen.¹

Mittels Erweiterungen lässt sich der Funktionsumfang erhöhen. Exemplarisch sei hier die Integration des BSI-Grundschutzes genannt, welcher durch eine entsprechende Erweiterung innerhalb von i-doit umgesetzt wird. Durch die Möglichkeit, komplexe Berichte auf Basis der gespeicherten Daten zu erstellen, sowie durch die Bereitstellung einer API, wird eine weitreichende Integration mit anderen Systemen ermöglicht. Dies

¹Vgl. [ido].

macht i-doit zu einem vielseitigen und strategisch bedeutenden Werkzeug im Kontext des IT-Betriebs der SLUB.

Live-Monitoring

Die fortlaufende Überwachung der IT-Systeme ist angesichts zunehmender Komplexität unerlässlich. Potenzielle Zwischenfälle wie das Erreichen von Festplattenkapazitätsgrenzen, ablaufende Zertifikate oder unzulässig hohe Antwortzeiten von Webservern stellen nur einen Bruchteil der möglichen Störungen dar. Angesichts dessen setzt die SLUB zur Überwachung ihrer IT-Infrastruktur auf das System checkmk.

Checkmk bietet ein umfassendes Repertoire an Überwachungsfunktionen – unabhängig davon, ob die Systeme in einer extern gehosteten Cloud oder im lokalen Rechenzentrum betrieben werden. Unterstützt werden physische Server, virtuelle Maschinen und Container gleichermaßen.² Die Überwachungsfähigkeiten lassen sich durch zahlreiche Plug-ins erweitern und verfeinern. Dabei bezeichnet der Begriff „Services“ innerhalb von checkmk typischerweise einzelne Softwarekomponenten (z. B. einen einzelnen Webservice) und unterscheidet sich somit in seiner Bedeutung vom ITSM-Kontext.

Plug-ins und Services können individuell entwickelt und integriert werden. Zudem bietet checkmk eine umfangreiche API, über die sich Automatisierungen und Systemverknüpfungen realisieren lassen. Damit fungiert checkmk nicht nur als Monitoring-Plattform, sondern auch als technisches Bindeglied zwischen verschiedenen Betriebsprozessen.

Schwachstellen-Überwachung

Angesichts der konstanten Bedrohung durch Angriffe auf IT-Systeme ist eine kontinuierliche Überwachung der Infrastruktur auf Sicherheitslücken unabdingbar. Mit wachsender Komplexität einer IT-Landschaft steigt auch die Anzahl potenzieller Angriffsvektoren, was eine strukturierte und automatisierte Analyse erforderlich macht.

Die SLUB verwendet zu diesem Zweck den Greenbone Security Manager (GSM), ein System zur automatisierten Schwachstellenanalyse. Der GSM wird so konfiguriert, dass er definierte Zielgruppen, bestehend aus physischen und virtuellen Maschinen, anhand vordefinierter Zeitpläne scannt. Die Ergebnisse dieser Scans werden in Form von Berichten zusammengefasst, die eine punktuelle sowie chronologische Bewertung des Sicherheitsstatus erlauben.

²Vgl. [cmk].

Diese Berichte lassen sich zur weiteren Bearbeitung automatisiert an das zentrale Ticketsystem übermitteln. Zusätzlich besteht durch die API die Möglichkeit, Prozesse und Integrationen in angrenzende Systeme zu realisieren. Der GSM ist damit ein wesentliches Element zur Aufrechterhaltung und Verbesserung der IT-Sicherheit an der SLUB.

Ticket-System

Zur Verwaltung von Anfragen und Störungen wird bei der SLUB das Ticket-System Znuny eingesetzt. Dabei handelt es sich um eine Abspaltung der ehemals freien Software OTRS. Im Rahmen des ITSM bildet das Ticket-System den zentralen Service Desk, über den sämtliche Anfragen, Störungen und Kontaktaufnahmen zu Services erfasst und bearbeitet.

Innerhalb von Znuny werden Anfragen in sogenannten Queues organisiert, spezifischen Bearbeitern zugewiesen und deren Bearbeitungsstatus dokumentiert. Das System erlaubt eine detaillierte Nachverfolgung des Anfrageverlaufs. Auch Znuny bietet eine API, über welche sich Integrationen mit anderen Systemen ermöglichen lassen. Damit trägt das Ticket-System wesentlich zur Prozessautomatisierung und Nachvollziehbarkeit von Serviceprozessen bei.

Code-Repository

Im Rahmen moderner DevOps-Methoden, wie Software Defined Networking (SDN) oder Infrastructure as Code (IaC), wird ein wachsender Teil der IT-Infrastruktur in Form von Code abgebildet. Dieser Code ist häufig, wie im Fall der Orchestrierungslösung Ansible, selbstdokumentierend und ersetzt damit klassische Dokumentationsformen.

Zur Verwaltung dieser Codeteile nutzt die SLUB ein selbst gehostetes GitLab-System. GitLab basiert auf der Versionsverwaltung Git und erweitert diese um umfangreiche Funktionen im Bereich des Projektmanagements (z. B. Kanban), der Dokumentation, der Ticketverfolgung sowie um DevOps-Komponenten wie CI/CD und Release-Management. Auch GitLab verfügt über eine umfassende API, womit es als zentrales Element für dokumentationsbasierte Infrastrukturentwicklung an der SLUB dient.

Virtualisierungsumgebung

Obwohl Virtualisierungsumgebungen (VE) nicht primär als Dokumentations- oder Monitoringsysteme konzipiert sind, liefern sie dennoch wertvolle Informationen für das Configuration Management. Die an der SLUB eingesetzte Lösung VMware stellt

mindestens rudimentäre Monitoring-Daten zu den gehosteten virtuellen Maschinen bereit.

Ferner sind insbesondere bei der Erstellung neuer virtueller Maschinen (VM) oder Container relevante Konfigurationsdaten verfügbar, die in der CMDB genutzt werden können. Der Virtualisierungsgrad an der SLUB ist mit rund 89 % (Stand: 30.06.2025) sehr hoch. Auch VMware stellt eine API bereit, die eine systematische Integration in andere Management- und Dokumentationssysteme ermöglicht.

Intranet

Ergänzend zu den bisher genannten Systemen spielt auch das Intranet eine bedeutende Rolle. Die SLUB setzt hierbei auf ein selbst gehostetes Confluence-System. Es handelt sich um eine unternehmensweite Intranet-Lösung mit einem stark ausgeprägten sozialen Fokus. Der Mensch als Nutzer steht im Zentrum, wodurch sich das System als Social Intranet klassifizieren lässt.

Das Intranet ist nicht als klassisches Dokumentations- oder Monitoringsystem zu verstehen, da es Inhalte im Freitext verwaltet und damit keine strukturierten Daten liefert. Dennoch ist es aufgrund seiner hohen Nutzerakzeptanz und seiner breiten Anwendungsbasis in der täglichen Arbeit aller Abteilungen von strategischer Bedeutung. Es dient als Plattform für Informationen, die in den anderen Systemen schwer oder gar nicht abbildbar sind, beispielsweise für Nutzerhandbücher oder Projektdokumentationen. Über eine bestehende API werden bereits heute zahlreiche Inhalte automatisiert, basierend auf Daten aus anderen Systemen, generiert und veröffentlicht.

3.2 Bestehendes ITSM-Konzept

Zentrales Element des ITSM der SLUB ist das bereits erwähnte i-doit-System, das zugleich als CMDB dient. Die aktuelle Konzeption zur Darstellung von Services und deren Konfiguration unterscheidet zwischen vier Klassen von Services: Business Service IT, Infrastructure Service, Technical Service IT und Technical Service IT extern. Weitere Knoten innerhalb der CMDB sind u. a. Geschäftsprozesse, virtuelle Server, physische Server und Netzlaufwerke.

Beziehungen werden primär über den Abhängigkeitstyp *Service-Komponente* modelliert. Die Services bilden den Kern des ITSM und sind entsprechend definiert und strukturiert:

Business Service

Business Services (BS) stellen für interne oder externe Kunden erbrachte Leistungen dar, die namentlich benannt werden können. Sie sind mindestens einem Geschäftsprozess (GP) zugeordnet und dienen dessen Zielerfüllung. BS können ausschließlich untergeordnete Knoten vom Typ Technical Service (TS) aufweisen. Ein Beispiel hierfür ist das Datenbankinformationssystem DBoD, welches E-Ressourcen mit Rechteprüfung bereitstellt. Ein weiteres Beispiel ist der Authentifizierungsservice auf Basis von Shibboleth, der als SAML-Identity Provider fungiert und unter anderem DBoD als Backend dient.

Technical Service

Technical Services fungieren als Bindeglieder zwischen BS, weiteren TS sowie technischen Ressourcen wie Servern oder virtuellen Maschinen. Sie ermöglichen die technische Realisierung der BS und sind hierarchisch immer unter BS oder anderen TS eingeordnet.

Technical Service extern

Die Technical Service extern (TSe) werden von externen Anbietern bereitgestellt und unterstützen die Erbringung eines BS. Sie sind funktional analog zu internen TS zu betrachten, dürfen jedoch keine Endgeräte untergeordnet haben.

Infrastructure Service

Infrastructure Services (IS) werden von nahezu allen Geschäftsprozessen genutzt und ihre Existenz gilt als bekannt. Die tatsächliche Nutzung wird im CM der SLUB jedoch nicht explizit abgebildet. Beispiele für IS sind das zentrale Speichersystem (z. B. SAN, NFS-Shares, SMB-Shares) sowie das Intranet, das trotz intuitiver Abgrenzung als Infrastrukturservice klassifiziert ist.

3.3 Zusammenwirken der bestehenden Systeme

Innerhalb der SLUB Dresden steht eine Vielzahl von Systemen zur Verfügung, um im Sinne des IT Service Managements sowie des Configuration Managements Daten zu erschließen, zu verarbeiten und vorzuhalten. Hierzu zählen insbesondere die Configuration Management Database, das Intranet, verschiedene Monitoringlösungen, die Virtualisierungsumgebung, ein Schwachstellenscanner, das zentrale Code-Repository sowie das Ticket-System. Diese Systeme sind jedoch nicht von Haus aus integriert,

sondern müssen durch geeignete technische Maßnahmen – insbesondere über die angebotenen Programmierschnittstellen (APIs) – miteinander vernetzt werden.

Im nachfolgenden Abschnitt wird exemplarisch aufgezeigt, wie die genannten Systeme miteinander interagieren. Zu diesem Zweck wird der Lifecycle eines virtuellen Servers innerhalb der IT-Landschaft skizziert und anhand dessen die zugrunde liegenden Informationsflüsse und Mechanismen erläutert. Nachdem ein Server über das automatisierte Bereitstellungssystem GUBS konfiguriert wurde, erzeugt dieses mittels der in VMware integrierten API eine neue virtuelle Maschine (VM) und initiiert deren automatische Installation über das PXE-Verfahren [Glossar-Eintrag hinzufügen]. VMware übermittelt im Anschluss grundlegende Metadaten zur neu erstellten VM an die Bereitstellungsplattform. Diese Informationen dienen als Grundlage dafür, die VM in weiteren Systemen bekannt zu machen.

Auf dieser Basis erfolgt unter anderem die automatische Anlage eines Monitoring-Objekts in checkmk sowie eines Scanziels im Schwachstellenscanner Greenbone Security Manager. Das Monitoring ist dabei auf fünf sogenannte Sites aufgeteilt, von denen jede einem der fünf IT-Referate zugeordnet ist. Die neu erzeugten Server werden jeweils im Root-Bereich der entsprechenden Site angelegt. Im Schwachstellenscanner wiederum stehen insgesamt fünf Scan-Konfigurationen zur Verfügung; der neu erstellte Server wird dabei dem Scan desjenigen Referats zugewiesen, dem auch die anfordernde Person angehört. Sofern während der initialen Konfiguration über GUBS eine Klassifikation als Testsystem vorgenommen wurde, erfolgt darüber hinaus eine automatische Zuordnung des Servers zu einem vordefinierten Test-Service in der CMDB, der alle Testsysteme strukturiert unter sich gruppiert.

Die Interaktion der Systeme entlang dieses Lifecycles verdeutlicht, dass trotz bestehender Automatisierungsmechanismen weiterhin ein erheblicher Bedarf an Integration und Koordination besteht, um eine konsistente und umfassende Dokumentation sowie Überwachung der IT-Ressourcen sicherzustellen.

3.4 Probleme mit dem bisherigen ITSM-Konzept

Das in der SLUB implementierte ITSM-Konzept orientiert sich in weiten Teilen an den Vorgaben des IT Infrastructure Library-Rahmenwerks. Dennoch ergeben sich bei einer stringenten Umsetzung der konzeptuellen Vorgaben diverse praktische Probleme, insbesondere im Kontext der Verantwortlichkeitsstruktur sowie der systemübergreifenden Automatisierung.

Zur Veranschaulichung der bestehenden Problematik wird nachfolgend eine vereinfachte Struktur des Business Service DBoD dargestellt. Der BS DBoD fungiert als

zentrales Element des entsprechenden Produkts und ist einem oder mehreren Geschäftsprozessen zugeordnet, die auf dessen Funktionalität angewiesen sind. Vom BS DBoD zweigen mehrere Technical Services ab, darunter etwa der DBoD-Webservice. Für dessen Betrieb ist wiederum der TS Shibboleth-IdP erforderlich. Weiterhin besteht eine Abhängigkeit zum übergeordneten BS Authentifizierung (Shibboleth) sowie weiterführend zum TS Libero.

Diese Struktur verdeutlicht die hierarchisch verzweigte und stark verflochtene Abhängigkeit von Services. Problematisch wird diese Darstellung jedoch in Hinblick auf die Verantwortlichkeitsgrenzen. So wird unter anderem der TS Shibboleth-IdP von einem anderen Team betreut als der BS DBoD. Gleiches gilt für den TS Libero. Damit werden innerhalb weniger Abstraktionsebenen zwei Verantwortungsbereiche überschritten – eine Information, die aus der grafischen Darstellung oder aus den zugrundeliegenden CMDB-Daten nicht eindeutig hervorgeht.

Für gewöhnlich ist einem BS ein dediziertes Team zugeordnet, das sowohl für den BS als auch für die ihm unmittelbar zugehörigen TS und Systeme verantwortlich ist. Die Nutzung von Services anderer Teams ist zwar üblich, die Verantwortung liegt in diesen Fällen jedoch außerhalb des primären Teams. Diese Verantwortlichkeitsverschiebung wird im Configuration Management (CM) der SLUB nicht explizit abgebildet.

Zudem ist eine Ableitung der Teamzugehörigkeit über die in den Objekten hinterlegten Personalinformationen nicht zuverlässig möglich. So kann ein Mitarbeiter, etwa Herr Müller, sowohl im Team DBoD als auch im Team Shibboleth Mitglied sein, was zu einer fehlerhaften Interpretation der Systemzugehörigkeit führen kann. Die einzige Information, die aktuell sicher aus dem CMDB-Modell entnommen werden kann, ist die Zugehörigkeit zu einem der fünf bestehenden IT-Referate. Da es jedoch auch teamspezifische Zuständigkeiten über Referatsgrenzen hinweg gibt, ist auch diese Information nicht immer zielführend.

Dieser Mangel an klarer Zuordnung und expliziter Verantwortlichkeitsdarstellung hat unmittelbare Auswirkungen auf die Automatisierung. So erfolgt etwa die automatisierte Integration von Servern in das Monitoring-System ausschließlich über eine Zuweisung zu einer der fünf Hauptgruppen, welche den IT-Referaten entsprechen. Die Sichtbarkeit der Systeme ist dabei auf die jeweiligen Referatsadministratoren und -leitungen beschränkt. Eine Sichtbarmachung für die zuständigen Teams erfordert manuelle Nachbearbeitung – der Automatisierungsprozess endet somit in einem manuellen Eingriff.

Eine vergleichbare Problematik zeigt sich im Kontext der Schwachstellenüberwachung. Auch hier erfolgt die Zuweisung der gescannten Systeme auf Referatsgruppenebene. Eine feinere Granularität, etwa eine servicebezogene Gruppierung, kann lediglich

durch manuelle Konfiguration erreicht werden.

Auch innerhalb der Virtualisierungsumgebung, genauer im Frontend-System vSphere, tritt dieses Problem zutage. Server werden hier automatisch dem Nutzer zugeordnet, der die VM über GUBS beantragt hat. Soll ein Team seine Systeme konsolidiert in einem gemeinsamen Verzeichnis strukturieren, beispielsweise zur Vergabe von Zugriffsrechten oder zur Nutzung von vApps, ist hierfür erneut eine manuelle Intervention durch Administratoren mit entsprechenden Rechten erforderlich.

Die CMDB stellt in diesem Zusammenhang das System mit der größten Lücke dar. Während Testsysteme automatisch einem entsprechenden Test-Service zugeordnet werden, erfolgt bei produktiven VMs lediglich eine grundlegende Erfassung im System. Eine automatische Zuordnung zu Services oder das Anlegen relevanter Abhängigkeiten findet nicht statt. Die betreffenden Objekte verbleiben daher in einem undefinierten Zustand im Configuration Management und bedürfen manueller Nachbearbeitung durch das zuständige Personal.

Die Notwendigkeit zahlreicher manueller Tätigkeiten in verschiedenen Systemen birgt nicht nur ein hohes Fehlerpotential, sondern verursacht darüber hinaus zusätzlichen personellen Aufwand. Dieser Mehraufwand muss im Sinne einer betriebswirtschaftlichen Betrachtung ebenfalls als Kostenfaktor berücksichtigt werden.

3.5 Vorgehaltene Daten

Die verschiedenen Systeme der SLUB halten eine Vielzahl von Informationen vor. Nicht alle dieser Daten sind für das IT Service Management oder das Configuration Management unmittelbar relevant. Die nachfolgende Darstellung erhebt daher keinen Anspruch auf Vollständigkeit; vielmehr werden diejenigen Informationen aufgeführt, die im Hinblick auf die zuvor beschriebenen Problemlagen als besonders bedeutsam einzuschätzen sind. Die Daten werden systemweise strukturiert dargestellt. Die Verknüpfung zwischen den Systemen erfolgt dabei stets über den jeweiligen Bezeichner – im Fall von Servern der Hostname und im Fall von Services der korrespondierende Servicename.

3.5.1 CMDB

Da innerhalb der CMDB nicht nur eine einzige Klasse von Objekten abgebildet wird, erfolgt eine getrennte Betrachtung der verschiedenen Klassen sowie ihrer jeweiligen Beziehungen untereinander. Zunächst werden die den Klassen zugeordneten Kategorien – also Sammlungen von Attributen – in Form einer Matrix aufgeführt. Anschließend

Metadatum	BS	TS	TSe	IS	VM
Allgemein	x	x	x	x	x
Externe Partner-Accounts	x			x	
IT-Grundschutz (Zielobjekte) *	x	x	x	x	x
Service *	x	x	x	x	
Service-Abschaltung		x			
Servicedokumentation TS		x			
Servicemerkmale TS		x			
Datenschutz				x	
Servicedokumentation (Intranet)	x		x	x	
Servicedokumentation (IT-Hotline)	x		x	x	
Servicemerkmale1 SLUB	x			x	
Servicemerkmale3 TUD	x				
Vertragszuweisung	x		x		
Betriebssystem					x
Black Building Test					x
Buchhaltung					x
Hostadresse					x
SLUB:Betriebsorganisation					x
SLUBKonfiguration1					x
Softwarezuweisung					x

Tabelle 3.1: CMDB-Klassen und zugehörige Kategorien

werden die einzelnen Kategorien und ihre Attribute hinsichtlich Inhalt und Zweck analysiert.

Kategorien und Attribute

Übergreifend gepflegt werden die Kategorien Allgemein und IT-Grundschutz. Aus der Kategorie Allgemein werden der Bezeichner sowie der CMDB-Status genutzt. Jedes Objekt verfügt über einen Bezeichner, der innerhalb seiner Klasse eindeutig sein sollte; technisch ist jedoch eine Mehrfachvergabe möglich. Der CMDB-Status zeigt an, in welchem Stadium sich das Objekt aus Sicht des Configuration Managements befindet – etwa „in Arbeit“ oder „in Betrieb“. Auf diese Weise wird sichtbar, wie weit die Erfassung eines Services fortgeschritten ist. Der Status wird manuell gepflegt.

Die Kategorie IT-Grundschutz (Zielobjekte) wird durch das in i-doit integrierte BSI-Grundschutz-Plug-in bereitgestellt. Da sie inhaltlich vollständig dem IT-Grundschutz zuzuordnen ist und dieser im Rahmen der vorliegenden Arbeit nicht weiter betrachtet wird, erfolgt an dieser Stelle keine vertiefte Analyse.

Die Kategorie Externe Partner-Accounts enthält eine vordefinierte Liste externer Einrichtungen, die über spezifische Nutzerkonten Zugriff auf den jeweiligen Service haben.

Die Kategorie Service-Abschaltung beinhaltet Informationen zu Abschaltanweisungen und -reihenfolgen. Hier wird unter anderem definiert, ob die Abschaltung auf Grundlage der in i-doit modellierten Abhängigkeiten automatisch zu ermitteln ist oder ob entsprechende Daten aus der Virtualisierungsplattform herangezogen werden sollen. Zudem ist ein Prüfdatum zu erfassen.

Die Kategorien Datenschutz, Servicedokumentation TS, Servicedokumentation (Intranet) und Servicedokumentation (Hotline) bestehen jeweils aus Attributen des Typs „Link“, der das Hinterlegen von URLs ermöglicht. Während Servicedokumentation TS inhaltlich der Servicedokumentation (Intranet) entspricht, verweist Servicedokumentation (Hotline) auf eine standardisiert angelegte Hotline-Hilfeseite. Unter Datenschutz befindet sich ein Link zum Verzeichnissesverzeichnis gemäß Datenschutz-Grundverordnung (DSGVO).

Die Kategorie *Service* ist eine vom System bereitgestellte Sammlung mehrerer Unterkategorien, die allgemeine Serviceattribute umfasst. Hervorzuheben sind hier insbesondere:

- *Business Unit* zur benannten Zuordnung des Services zu einer Abteilung,
- *Service-Kategorie*, *Service-Art* und *Service-Typ*, die zwar definierbar, derzeit aber ungenutzt sind,
- die Felder für den *Service Manager*,
- interne und externe Servicebeschreibungen,
- sowie die Möglichkeit, unter *Service-Komponenten* weitere Objekte mit dem Beziehungstyp „Service-Komponente“ zu verknüpfen.

Die Kategorie *Servicemerkmale*¹ SLUB umfasst darüber hinaus:

- die Service-URL (sofern es sich um einen Webservice handelt),
- eine kurze Servicebeschreibung,

- das Datum der erstmaligen Inbetriebnahme,
- sowie die fachliche Abteilungs- und Referatszuordnung.

Die Kategorie *Servicemerkmale3 TUD* enthält Informationen, die nicht für das interne ITSM benötigt werden, sondern der Übermittlung an die Technische Universität Dresden dienen. Dazu zählen u. a.:

- Ansprechpartner (BearbeiterIn),
- Betreiberangaben (für die SLUB stets „SLUB“),
- Kurz- und Langbeschreibung,
- Kapazitätsangaben,
- der Schutzbedarf gemäß BSI (ohne nähere Spezifikation hinsichtlich Integrität, Verfügbarkeit oder Vertraulichkeit),
- englischer Titel sowie englische Kurz- und Langbeschreibung,
- Katalog-Tags (Mehrfachauswahl),
- Kundengruppen (z. B. *Studierende, Beschäftigte, Gäste*),
- Informationen zur Kostenpflichtigkeit,
- Betriebs- und Supportzeiten,
- eine Service-URL (falls Webservice),
- sowie die Zuordnung der First-Level-Support-Rolle.

Die Kategorie *Vertragszuweisung* listet sämtliche Beziehungen zu zugewiesenen Vertragsobjekten auf.

Weitere Kategorien beziehen sich ausschließlich auf virtuelle Maschinen, deren Inhalte überwiegend sprechend gestaltet sind, etwa Betriebssystem oder Hostadresse. Die Kategorie *Softwarezuweisung* bildet dabei das Gegenstück zur unter *Service* aufgeführten Beziehung *Service-Komponente*.

Die Kategorie *Black Building Test* dokumentiert Daten aus einem früheren Projekt zur geregelten Ab- und Wiederinbetriebnahme aller Services; sie hält fest, ob und wann eine Maschine im Rahmen dieses Tests abgeschaltet wurde.

Die Kategorie *Buchhaltung* enthält kaufmännische Angaben wie Bestelldatum, Anschaffungskosten oder Garantiezeiträume. Da sie für den weiteren Verlauf der Arbeit nicht relevant ist, wird auf eine detaillierte Analyse verzichtet.

Unter *SLUB:Betriebsorganisation* wird gepflegt, ob es sich bei dem Objekt um ein Produktiv- oder Testsystem handelt.

Die Kategorie *SLUBKonfiguration1* enthält ausschließlich Zustandsinformationen, die in Form vordefinierter Statuswerte wie *ok*, *to-do* oder *nicht erforderlich* erfasst werden. Die Attribute dieser Kategorie werden seit geraumer Zeit nicht mehr aktiv gepflegt, weshalb eine detaillierte Betrachtung entfällt.

3.5.2 Weitere Hilfssysteme

Die weiteren Systeme, die im Kontext von ITSM, Monitoring und Dokumentation eingesetzt werden, enthalten überwiegend keine Informationen, die für das ITSM selbst erforderlich sind. Die Analyse zeigt, dass die CMDB nahezu alle relevanten Daten für ITSM, CM und die Abbildung systemischer Abhängigkeiten vorhält. Ausnahmen bestehen lediglich in geringem Umfang.

Eine solche Ausnahme bildet das System GUBS, welches für die Bereitstellung virtueller Maschinen genutzt wird. Neben der initialen Systemkonfiguration werden hier insbesondere Berechtigungen gepflegt. Diese erfolgen über frei definierbare Gruppen, denen Personen zugeordnet werden. Die zugeordneten Personen können anschließend Systemkonfigurationen – welche per PXE verteilt werden – für die ihnen zugewiesenen Server vornehmen.

3.6 Fazit

Insgesamt zeigt die Analyse, dass die vorhandenen Systeme eine breite, jedoch teilweise heterogene Datenbasis bereitstellen. Während die CMDB den zentralen Informationskern bildet, existieren in den weiteren Hilfssystemen nur wenige, teils spezialisierte Ergänzungen. Die identifizierten Lücken, Redundanzen und Inkonsistenzen verdeutlichen zugleich das Optimierungspotenzial für ein konsistentes, zweckgerichtetes IT Service Management.

Auf dieser Grundlage folgt im nächsten Kapitel die Entwicklung eines Konzepts, das die gewonnenen Erkenntnisse in ein strukturiertes Zielbild überführt und Lösungsansätze für die zuvor beschriebenen Problemlagen aufzeigt.

Kapitel 4

Konzeptionierung

In diesem Kapitel wird das neue Datenmodell der Configuration Management Database entworfen und um Prozesse sowie Mechanismen zur Erschließung der Daten ergänzt. Grundlage hierfür bilden die in Kapitel 3 erfassten Informationen, die hinsichtlich ihrer Notwendigkeit überprüft, von Redundanzen bereinigt und um fehlende Metadaten ergänzt werden. Eine Umfrage zur bisherigen CMDB-Struktur sowie zum Thema IT Service Management im Allgemeinen dient dazu, weitere Anforderungen für das Soll-Konzept zu identifizieren. Auf dieser Basis wird das eigentliche Datenmodell erstellt, wobei die Möglichkeiten und Einschränkungen der vorhandenen CMDB-Software i-doit besondere Berücksichtigung finden.

4.1 Befragung zum Umgang mit CMDBs und ITSM

4.1.1 Aufbau

Zur Präzisierung der Anforderungen an die Modellierung wurden Mitarbeiterinnen und Mitarbeiter der SLUB-IT zur bisherigen und zukünftigen Nutzung der CMDB sowie des ITSM befragt. Die befragten IT-Beschäftigten lassen sich in verschiedene Gruppen gliedern. Neben fest angestellten und projektgebunden Beschäftigten existieren innerhalb der Festangestellten unterschiedliche Rollen, etwa Führungspersonal sowie Mitarbeitende. Mitarbeitende lassen sich wiederum grob in Entwicklerinnen und Entwickler sowie Administratorinnen und Administratoren unterteilen. Unabhängig von der Zugehörigkeit zu Projekt- oder Festanstellung können mehrere Rollen gleichzeitig ausgeübt werden, beispielsweise in Form von Referatsleitung und Entwicklungsaufgaben oder als Kombination aus Entwicklung und Administration. Projektangestellte sind nahezu ausschließlich im Bereich der Entwicklung tätig und zeitlich nur begrenzt an der SLUB beschäftigt. Als Beschäftigte mit zeitlich begrenzter Anstellung haben sie nur wenig

Kontakt mit der CMDB. Daher wurden sie von der Befragung ausgeschlossen. Gleiches gilt für Entwicklerinnen und Entwickler im Allgemeinen, da sich gezeigt hat, dass die Pflege und Nutzung der CMDB wesentlich durch Führungspersonal, Administratoren und DevOps erfolgt.

Die Umfrage beschränkte sich daher auf fest angestellte Mitarbeitende der Administration sowie DevOps und auf das IT-Führungspersonal. Einzelne Projektmitarbeitende mit Tätigkeiten im Bereich der Administration und einem Vertragsverhältnis von mindestens fünf Jahren wurden als Ausnahme einbezogen. Insgesamt wurde die Umfrage an 21 Personen versandt, von denen 15 die Befragung begonnen und 11 vollständig abgeschlossen haben. Die Abbruchquote von etwa 27 % wirkt sich aufgrund der geringen Grundgesamtheit nachteilig auf die Ergebnisqualität aus. Auffällig ist, dass sämtliche abgebrochenen Befragungen an gleicher Stelle, Frage 2b, unterbrochen wurden. Diese Frage wurde ausschließlich Personen präsentiert, die die Datenqualität und den Informationsgehalt der aktuellen Modellierung nicht als optimal einschätzten. Es sollte in Freitextform erläutert werden, wie diese Mängel aussehen. Trotz der Optionalität der Frage brachen vier Personen die Befragung an dieser Stelle ab. Es wird angenommen, dass eine alternative Erhebungsmethode, etwa ein direktes Interview, geeignetere Ergebnisse geliefert hätte, da hierbei Abbrüche weniger wahrscheinlich gewesen wären und Fragen kleinteiliger hätten gestellt werden können.

Für die nachfolgende Auswertung wurden ausschließlich die elf vollständig ausgefüllten Fragebögen berücksichtigt. Die Umfrage umfasste 15 Fragen, wobei eine dieser Fragen nur in Abhängigkeit zur Antwort eine vorangegangene Frage gestellt wurde. Unter diesen 15 Fragen waren acht verpflichtende Multiple-Choice-Fragen sowie sieben optionale Freitextfragen. Die Entscheidung, offene Fragen fakultativ zu gestalten, sollte verhindern, dass Befragungen aufgrund von Unsicherheit oder Zeitmangel abgebrochen werden. Die Durchführung erfolgte vollständig anonym. Für die Erhebung wurde eine lokal gehostete Umfragesoftware installiert, um eine moderne, ansprechende und direkt am Arbeitsplatz nutzbare Befragung zu ermöglichen. Die erhobenen Daten verließen zu keinem Zeitpunkt das SLUB-Netzwerk; Der eingesetzte Server wurde nach Abschluss der Diplomarbeit entfernt. Während der Befragung wurde anstelle des Begriffs CMDB durchgängig der Produktname der intern eingesetzten Software i-doit verwendet.

4.1.2 Auswertung

Für die Auswertung wurden bestimmte Fragen und deren Antworten gemeinsam betrachtet. Hierzu wurden Filter gesetzt, um die Anzahl der Antworten zu reduzieren, diese gruppiert auszuwerten und anschließend weitere Fragen und Antworten der

gefilterten Gruppen zu betrachten.

Besonders interessant stellten sich die Antworten heraus, welche unter Berücksichtigung der Antworten auf die Frage *Wie bewerten Sie den aktuellen Informationsgehalt und die Datenqualität im i-doit?* gegeben wurden. Als Auswahl standen *Sehr gut, Gut, Ausreichend, Mangelhaft* und *Unbrauchbar* zur Verfügung. Nachfolgend wird die Befragung mit den gefilterten Ergebnissen *Sehr gut* und *Gut* betrachtet.

Nur drei der elf Befragten bewerteten die in der CMDB vorliegenden Daten hinsichtlich Qualität und Quantität als gut oder sehr gut und sahen damit keine wesentlichen Probleme auf Datenebene. Im Umkehrschluss stuften acht von elf Personen die Daten als *ausreichend* bis *unbrauchbar* ein. Bemerkenswert ist die Verteilung der Antworten auf Frage elf, die nach der Verantwortlichkeit für die Pflege der CMDB fragt. Der Personenkreis, der die Datenqualität als *gut* oder *sehr gut* bewertet, spricht sich durchgängig für eine zentrale Verantwortlichkeit aus. Eine solche zentrale Zuordnung entspricht dem derzeitigen Vorgehen an der SLUB, bei dem jeweils eine Person pro Referat alle Services der Kolleginnen und Kollegen pflegt.

Im Gegensatz dazu stimmte die Gruppe, welche die Datenqualität als mangelhaft beurteilt, überwiegend dafür, dass die Daten von den jeweiligen Serviceverantwortlichen selbst gepflegt werden sollten. Zwei Personen plädierten für eine gemeinsame Verantwortlichkeit, bei der Serviceverantwortliche für die Pflege zuständig sind, während in jedem Referat zusätzlich eine Ansprechperson als Multiplikator fungiert. Die Erwartung hinsichtlich dieses Fragepaars war entgegengesetzt zu den tatsächlich erhobenen Ergebnissen. Auch nach eingehender Betrachtung konnten bislang keine eindeutigen Ursachen für diese vermeintliche Unstimmigkeit identifiziert werden.

Im weiteren Verlauf wurde der Filter umgekehrt, sodass die Antworten derjenigen betrachtet wurden, die die Datenqualität als ausreichend bis unbrauchbar eingestuft hatten. Als Mängel wurden unter anderem benannt:

- fehlende Kategorien und Attribute, beispielsweise zu Berechtigungen wie SSH-Zugängen, Netzwerkspeichern oder Firewall-Freigaben zu Servern und Services,
- mangelnde Klarheit über das bestehende Datenmodell, insbesondere hinsichtlich der Frage, welche Attribute und Kategorien für welche Klassen vorhanden sind und wie diese zu pflegen sind,
- fehlerhafte oder unvollständige Daten infolge fehlender Automatisierung,
- Daten, die überflüssig oder trivial sind, etwa automatisch erzeugte Informationen aus der Virtualisierungsumgebung wie die Betriebssystemangabe *GNU Linux 64*

bit, oder Identifier wie VM-IDs in der Form *vm-19283*, die im Betrieb kaum nutzbar sind,

- eine unzureichende Befähigung zur Nutzung der CMDB, da für ein komplexes IT-System Schulungen, Handbücher und Fachpersonal notwendig seien,
- fehlende Klarheit über Verantwortungsbereiche und deren Sichtbarkeit in der CMDB,
- unklare Verantwortlichkeiten und Prozesse für die Pflege der CMDB und
- eine unzureichende visuelle Darstellung von Abhängigkeiten, wodurch Auswirkungen von Ausfällen schwer erkennbar seien.

Für eine bessere Nutzbarkeit der CMDB wurden folgende Bedingungen formuliert:

- deutliche Steigerung der automatisierten Erfassung und Pflege von Attributen und Kategorien,
- Schulungen zum Umgang mit dem System,
- Verbesserung der Visualisierung von Abhängigkeiten und Strukturen,
- stärkere Nutzung der Reporting-Funktion, nicht nur zur Identifikation von Pflege-Rückständen, sondern auch als Darstellungsebene für Informationen im Sinne von Views aus SQL-Datenbanken,
- Nutzung von Versionierung und Änderungshistorien,
- Integration bestehender Systeme und Prozesse als Datenquellen,
- Beseitigung redundanter Daten und
- regelmäßige Datenbereinigungen.

4.1.3 Fazit

Aus der Mitarbeiterbefragung ergeben sich mehrere Erkenntnisse für die Modellierung. Im Bereich der bestehenden Daten sollen Redundanzen aufgelöst, der Nutzen vorhandener Informationen überprüft und gegebenenfalls Attribute entfernt werden. Auch die bestehenden Klassen und deren Definitionen sollen hinterfragt werden. Für die Pflege und Erfassung von Daten wird empfohlen, Automatismen zu etablieren, bestehende Systeme und Prozesse zu integrieren sowie bislang nicht erfasste Daten zu

evaluieren und gegebenenfalls in das Modell aufzunehmen. Im Umgang mit der CMDB und dem ITSM sollten Schulungen angeboten, Handbücher erstellt, Multiplikatoren für Rückfragen bestimmt und Verantwortlichkeiten klarer kommuniziert werden.

Viele der geäußerten Antworten decken sich mit den im Kapitel 1 beschriebenen Problemen und Zielstellungen. Dies bestätigt die zuvor getroffenen Annahmen über bestehende Mängel und die daraus abzuleitenden Aufgaben.

4.2 Modellierung der CMDB

4.2.1 Klassen

Die bestehenden Klassen werden hinsichtlich ihres Nutzens überprüft und bei Bedarf neu gestaltet. Aus der IST-Analyse ist bekannt, dass fünf essenzielle Klassen für die CMDB-Modellierung auf ITSM-Ebene relevant sind. ITSM-Ebene bedeutet hierbei, dass nur Klassen und Objekte unterhalb der Geschäftsprozesse betrachtet werden. Objekte wie Arbeitsplatztechnik oder Verträge werden ebenfalls nicht berücksichtigt. Aus dem Bereich der zentralen Technik (Server, VMs, Netzwerkkomponenten) werden ausschließlich virtuelle Maschinen betrachtet, da diese etwa 89 % der Serverlandschaft ausmachen. Aus den bestehenden Service-Klassen werden im Wesentlichen der Business Service und der Technical Service beibehalten. Die Klasse des Infrastructure Service geht in die Technical Services über. Technische Services externer Anbieter, bisher als Technical Service extern geführt, werden zu einem späteren Zeitpunkt darauf geprüft, ob eine separate Klassifizierung notwendig ist. Diese Entscheidung erfolgte in Abstimmung mit der IT-Leitung, da externe Dienste zahlreiche zusätzliche Aspekte aufweisen (Mietgeräte, Wartungsverträge, Lizenzen), welche einer ausführlichen Betrachtung bedürfen und den Umfang dieser Arbeit übersteigen. Die weitere Modellierung konzentriert sich daher auf Business Services, Technical Services und virtuelle Maschinen, die definiert und in einem gemeinsamen Klassendiagramm zusammengeführt werden.

Business Service

Business Services dienen als Bindeglied zwischen Geschäftsprozessen und dem übrigen ITSM. Während Services allgemein einen Wert für Konsumierende bereitstellen, richtet sich der Nutzen eines Business Service ausschließlich an eine definierte Untergruppe, die als Customer bzw. Endnutzer bezeichnet wird. Hierzu zählen interne und externe Endnutzer, etwa Studierende, Mitarbeitende der Universität und der SLUB, studentische Hilfskräfte sowie nicht universitäre Bibliotheksmitglieder und Gäste. Business

Services sollen keine direkte Abhängigkeit von zentraler Technik wie Servern oder Netzwerktechnik aufweisen. Sie bilden die Schnittstelle zwischen Geschäftsprozessen und Technical Services und halten insbesondere die für Endnutzer relevanten Metadaten vor (Zielgruppen, Zugriffszeiten, Produktinformationen). Die Darstellung technischer Abhängigkeiten erfolgt ausschließlich über Technical Services.

Definition: Business Service

Business Services erbringen einen Nutzen für interne und externe Endnutzer. Sie dienen als Bindeglied zwischen Geschäftsprozessen und allen zur Leistungserbringung notwendigen Services und technischen Komponenten. Business Services besitzen keine direkte Abhängigkeit zu technischen Objekten (Server, VMs, Endgeräte oder Cluster).

Technical Service

Technical Services bilden die am häufigsten genutzte Serviceklasse. Sie geben Auskunft über die Art eines Service (z. B. Web- oder Applikationsserver), die zugehörigen Geräte sowie alle Informationen für den operativen Betrieb, etwa Verantwortlichkeiten und Serverzuordnungen. Auch Abhängigkeiten zu anderen Technical Services werden hier erfasst. Die Inhalte technischer Services decken ein breites Spektrum ab, weshalb nicht für jeden Zweck eine eigene Klasse erstellt wird. Die Klassifizierung erfolgt über Attribute wie Service-Typ, Service-Art und Service-Kategorie, die in i-doit frei konfigurierbar sind.

Definition: Technical Service

Technical Services beschreiben einen expliziten IT-Dienst. Ihnen werden andere Services und Technik (Cluster, physische Server, VMs) als Abhängigkeit zugeordnet. Ein Technical Service muss mindestens eine Service-Komponente besitzen.

Infrastructure Service

Im Rahmen der Konzeptionierung zeigte sich, dass keine Kategorien und Metadaten existieren, die ausschließlich einem Infrastructure Service zugeordnet wären und sich nicht gleichzeitig im Technical Service wiederfinden. Nach dem Prinzip der Datensparsamkeit wurde daher entschieden, Infrastruktur-Dienste als Technical Services zu pflegen. Sollte eine spätere visuelle Separierung notwendig werden, kann ein identischer Infrastructure Service per API eingeführt und Dienste migriert werden. Der Aufwand hierfür wird als gering eingeschätzt.

4.2.2 Kategorien und Attribute

Erläuterungen zu Visualisierungen

Die in diesem Kapitel verwendeten Klassendiagramme weichen von gängigen UML-Darstellungen ab. Attribute werden ohne Sichtbarkeit aufgeführt und Methoden werden nicht dargestellt. In i-doit sind Attribute stets Kategorien zugeordnet, nicht jedoch direkt Klassen. Kategorien enthalten beliebig viele Attribute sowie ein verpflichtendes Attribut Beschreibung. Kategorien können einer Klasse einzeln oder als Sammlung zugeordnet werden. Zur Vereinfachung werden verschachtelte Kategorien, sofern sinnvoll, aufgelöst dargestellt. Attribute werden mit einem vorangestellten Punkt gelistet. Kategorien können als Listen-Kategorien definiert werden, wodurch eine 1:n-Beziehung ermöglicht wird. Im Diagramm wird dies durch eine eigene Klasse mit entsprechender Kardinalität dargestellt.

Service-übergreifend

Zur Modellierung wird zunächst eine generische Klasse *Meta-Service* definiert, deren Bezeichnung zur Vermeidung von Verwechslungen mit der in i-doit vorhandenen Kategorie *Service* um den Zusatz *Meta* erweitert wurde. Diese Klasse existiert ausschließlich konzeptionell, da i-doit keine abstrakten Klassen erlaubt. Die *Meta*-Klasse wird inklusive ihrer Kategorien und Attribute in einem Klassendiagramm dargestellt (siehe Abbildung 4.1).

Durch die neue Kategorie *Links* können mehrere bisher getrennt gepflegte Kategorien aus Business und Technical Services zusammengeführt werden. Die bisherigen Kategorien *Servicedokumentation (Intranet)*, *Servicedokumentation (IT-Hotline)*, *Servicedokumentation TS* sowie *Datenschutz* enthielten ausschließlich URL und wurden vollständig überführt. Das Attribut *Typ* erlaubt die Klassifikation der URLs und ermöglicht eine automatisierte Qualitätskontrolle mittels Reports.

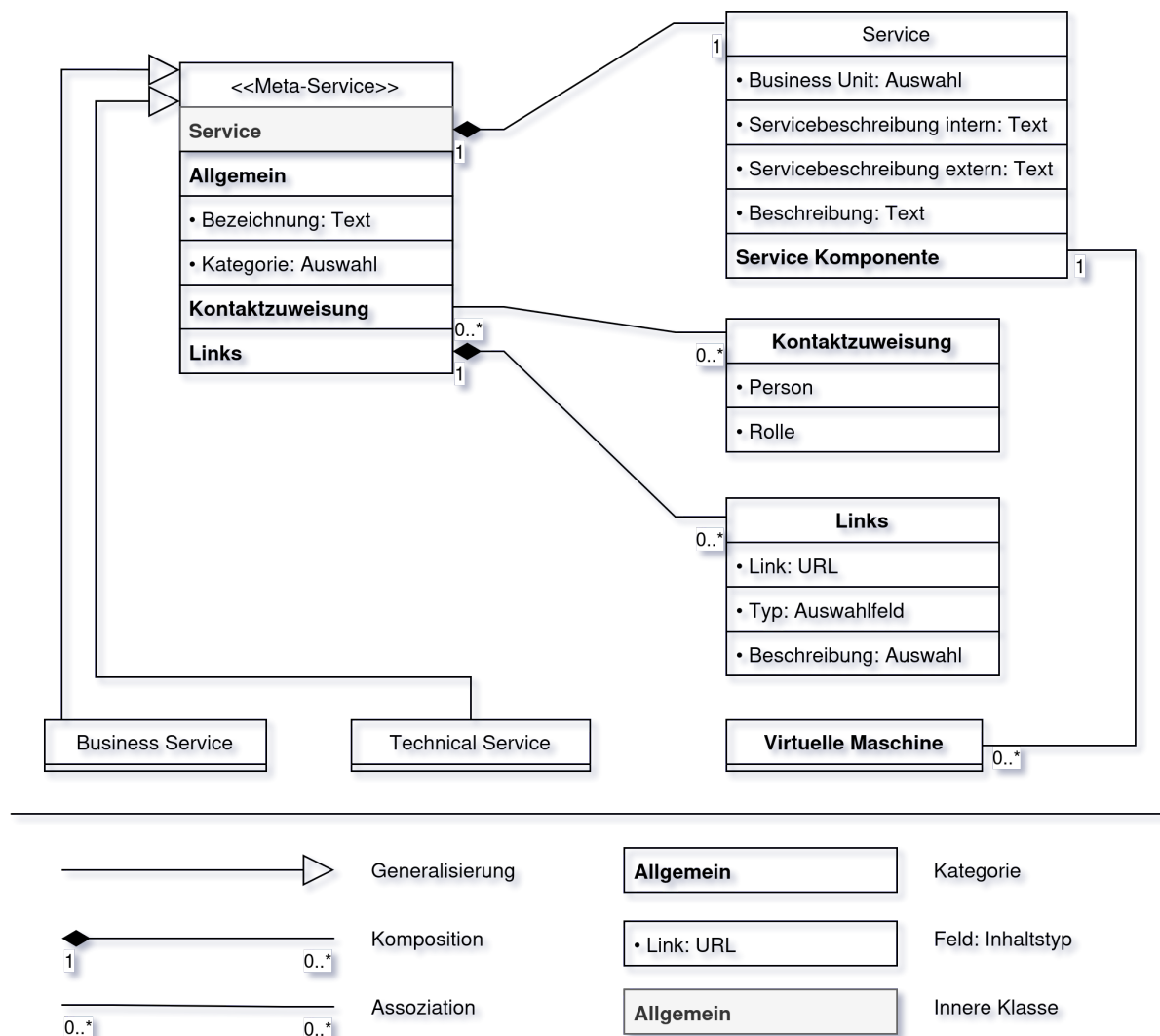


Abbildung 4.1: Klassendiagramm: *Meta*-Klasse und Abhängigkeiten

Business Services

Die Klasse Business Service (siehe Abbildung 4.2) bleibt weitgehend unverändert. Kategorien, welche für Endnutzer relevanten Informationen vorhalten, wie etwa *Externe Partner-Accounts* und *PR-Mitbestimmung*, bleiben bestehen. Vertragszuweisungen werden ebenfalls analog zum Bestand berücksichtigt. Die Kategorie *Servicemerkmale1 SLUB* entfällt, da deren Inhalte teils redundant und teils nicht länger notwendig sind. Das Attribut *URL* zum *Service* wird in die neue Kategorie *Links* überführt; weitere Attribute sind bereits in bestehenden Kategorien enthalten oder lassen sich über *Kontaktzuweisungen* abbilden.

Die Kategorie *Servicemerkmale3 TUD* wird teilweise übernommen und künftig als *Servicekatalog TUD* geführt. Redundante Informationen entfallen, stattdessen werden

nur nicht anderweitig gepflegte Daten erfasst. Ein Report aggregiert anschließend alle für den Servicekatalog der TU Dresden notwendigen Informationen.

Zur Abbildung endnutzerrelevanter Angaben wird die neue Kategorie *Service User Information* erstellt, die Attribute wie *Katalog-Tags*, *Kundengruppe*, *Bereitstellungszeit*, *Supportzeit*, *Kosten* und *zubuchbare Optionen* enthält.

Technical Services

Die Kategorie *Servicemerkmale TS* wird entfernt, da ihr einziger Inhalt, die Zuordnung zu einem Referat, entweder über Kontaktzuweisungen abbildbar oder inhaltlich nicht eindeutig ist. Gleiches gilt für die Kategorie *Serviceabschaltung*. Die hier adressierten Verfahren werden künftig ausschließlich in der Virtualisierungsumgebung gepflegt, da sie im SLUB-Alltag nur selten benötigt werden. Die Entscheidung erfolgte in Abstimmung mit der IT-Leitung. Damit entspricht der Technical Service inhaltlich der *Meta*-Klasse.

Virtuelle Maschinen

Mehrere Kategorien der virtuellen Maschinen entfallen, darunter *Buchhaltung*, *Black Building Test*, *Servicezuweisung* und *SLUB:Betriebsorganisation*. Die Kategorie *SLUBKonfiguration1* wird vollständig entfernt, da ihre Inhalte überwiegend redundante oder unpräzise Zustandsangaben enthalten. Die hier erfassten Kontrollinformationen sollen künftig durch Reports abgebildet werden. Die Kategorie *Softwarezuweisung* bleibt erhalten, da sie für Schwachstellen- und Störungsanalysen relevant sein kann; eine automatisierte Überführung der CMK-Inventare in i-doit wäre hierbei sinnvoll.

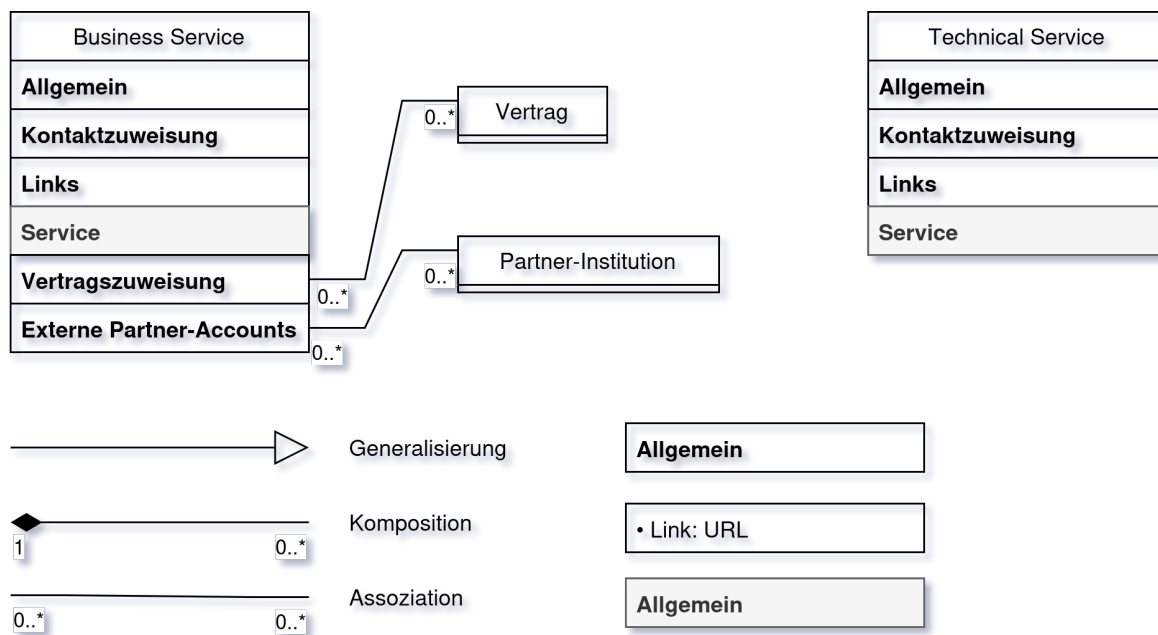
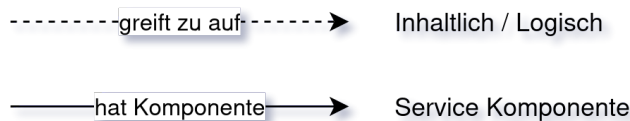
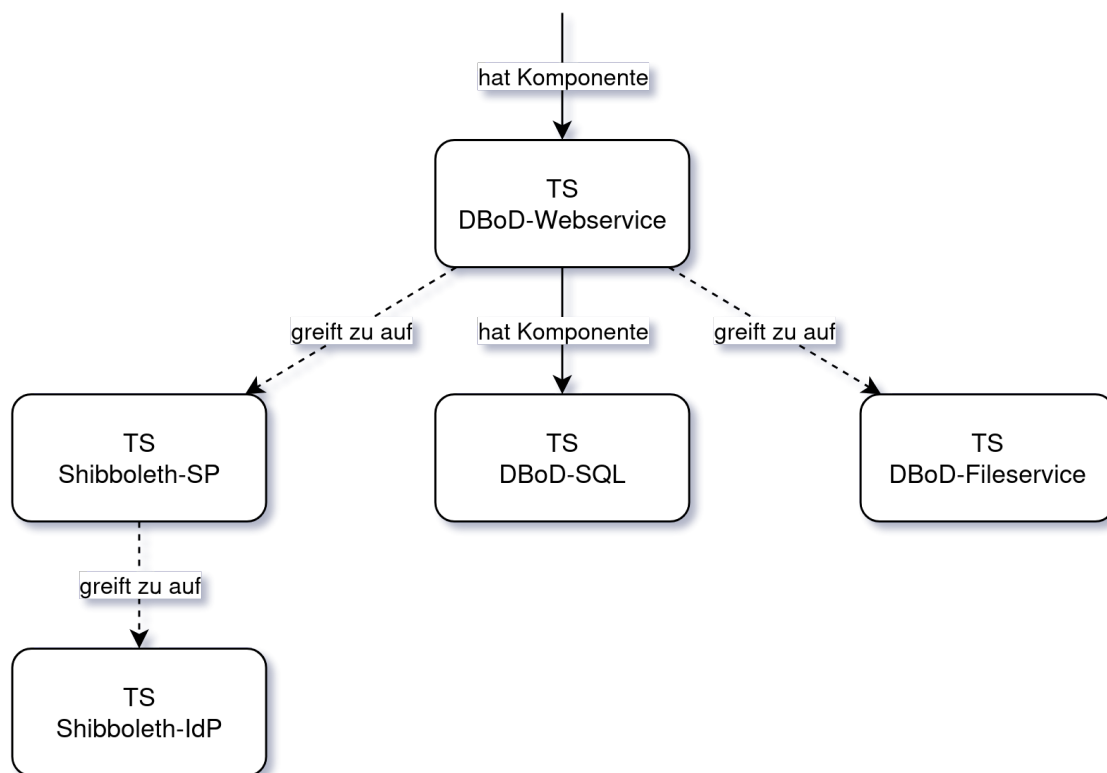


Abbildung 4.2: Klassendiagramm: Business und Technical Services

4.2.3 Beziehungsarten

Bislang wurden sämtliche Beziehungen zwischen Services und Maschinen über die von i-doit vordefinierte Beziehungsart *Service-Komponente* abgebildet. Dies führte zu Analyseproblemen, etwa im Störfall, da technische und nicht-technische Abhängigkeiten nicht getrennt dargestellt werden konnten. Künftig sollen daher unterschiedliche Beziehungsarten eingeführt werden. Die *Service-Komponente* bleibt als technische Abhängigkeit bestehen und beschreibt notwendige Beziehungen nach dem Prinzip der Komposition. Nicht notwendige Abhängigkeiten werden hingegen als *Inhaltlich / Logisch* definiert. Diese Beziehungsart umfasst sowohl inhaltliche als auch logisch bedingte Abhängigkeiten.

Am Beispiel des Webdienstes DBoD (siehe Abbildung 4.3) wird deutlich, dass technische Abhängigkeiten, etwa zu einer Datenbank, von rein inhaltlichen oder logischen Beziehungen zu unterscheiden sind, wie der Authentifizierungsdienst oder ein Fileservice. Diese Differenzierung erleichtert die Analyse und Behebung von Störungen. i-doit erlaubt zudem eine Gewichtung der Beziehungen, wodurch auch nichttechnische Abhängigkeiten angemessen dokumentiert werden können. Über Filter lassen sich in der CMDB getrennte Sichten definieren, um rein technische oder nicht technische Abhängigkeitsstrukturen zu visualisieren.


Abbildung 4.3: Beziehungsarten: Service-Komponente und *Inhalt / Logisch*

4.2.4 Verantwortungsgrenzen

Vorüberlegung

Neben Klassen, Kategorien, Attributen und Beziehungsarten fehlte bislang die in Kapitel 3 identifizierte und durch die Mitarbeiterbefragung bestätigte Verantwortungsgrenze zwischen Services. Um die Verantwortung für eine Gruppe von Services eindeutig bestimmen zu können, wurde daher eine neue Klasse definiert und modelliert: die Service Domain. Ziel war eine möglichst intuitive Bezeichnung, die den intendierten Inhalt klar widerspiegelt.

Der Begriff Domain bzw. Domäne besitzt laut Duden verschiedene, in ihrer Grundbedeutung ähnliche Konnotationen, etwa „Staatsgut“ oder „landesherrlicher Besitz“.¹

¹Vgl. [dudb].

In Anlehnung an das verwandte Wort *Dominium*² kann darunter auch ein Herrschaftsbereich verstanden werden. Entsprechend lässt sich der Begriff auf einen Verantwortungsbereich übertragen. Ferner stimmt diese Bedeutungsrichtung mit den in der Informationstechnik etablierten Verwendungen des Begriffs überein, beispielsweise in der Netzwerktechnik (Kollisionsdomäne) oder im Kontext von Internet-Domains.

Die Zusammensetzung aus Service und Domäne kann somit als Menge definierter Services interpretiert werden, die einem gemeinsamen Herrschafts- bzw. Verantwortungsbereich zugeordnet sind. Die Service Domain wird im Rahmen dieser Arbeit wie folgt definiert:

Definition: Service Domain

„Jeder Service muss genau einer Servicedomäne zugeordnet sein. Eine Servicedomäne ist die Gruppe aller Services, die dem gleichen primären Zweck dienen und im gleichen Verantwortungsbereich (Gruppe von Personen) liegen.“

Um zu prüfen, ob die gewählte Bezeichnung intuitiv mit dem definierten Inhalt übereinstimmt, wurden mehrere Kolleginnen und Kollegen der IT befragt. Zunächst wurde lediglich der Begriff präsentiert und erfragt, welche Erwartungen er auslöst. Anschließend wurden diese Erwartungen mit der oben genannten Definition abgeglichen. Die geäußerten Vorstellungen stimmten nahezu vollständig mit der intendierten Definition überein.

Modellierung

Die Klasse Service Domain wird entsprechend modelliert (siehe Abbildung 4.4). In i-doit können Klassen einer sogenannten *spezifischen Kategorie* zugewiesen werden. Erfolgt diese Zuordnung, wird die neue Klasse in den Dialogen und Mechanismen der CMDB so behandelt, als sei sie selbst eine Klasse der zugeordneten spezifischen Kategorie. Dies entspricht funktional einer Spezialisierung im Sinne eines Klassendiagramms. Die Klasse Service Domain wird der spezifischen Kategorie *Personengruppe* zugeordnet. Dadurch kann sie als *Kontaktzuweisung* gepflegt werden, was wiederum ermöglicht, die Service Domain als *primären Kontakt* zu hinterlegen.

In der visuellen Darstellung der CMDB kann die Service Domain somit direkt in der Übersicht angezeigt werden, sodass Verantwortungsgrenzen unmittelbar erkennbar werden. Die Sammelkategorie *Personengruppe* ist für Klassen dieser spezifischen Kategorie bereits systemseitig vordefiniert. Sie umfasst unter anderem die Kategorie

²Vgl. [duda].

Personengruppenmitglieder, in der Personen hinzugefügt und mit Rollen versehen werden können. Zudem können hier alle zugewiesenen Services auf einen Blick angezeigt werden.

Ergänzend werden zwei benutzerdefinierte Kategorien eingeführt: *Service* und *Recovery*. Die Kategorie *Service* enthält sowohl die im Ticketsystem verwendete Queue-Bezeichnung als auch die dieser Queue zugeordnete E-Mail-Adresse. In der Kategorie *Recovery* sollen künftig Informationen hinterlegt werden, die beschreiben, wie die Service Domain im Falle einer Havarie wiederhergestellt werden kann. Der konkrete Inhalt dieser Kategorie wird in einem separaten Folgeprojekt festgelegt.

Die Entscheidung, die Service Domain als eigenständige Klasse statt als benutzerdefinierte Kategorie umzusetzen, bietet mehrere Vorteile. Die Service Domain kann als *Kontaktzuweisung* definiert werden, ist in der Lage, komplexere Datenstrukturen aufzunehmen als eine einzelne Kategorie und kann als eigener Knoten in der Visualisierung der CMDB dargestellt werden.

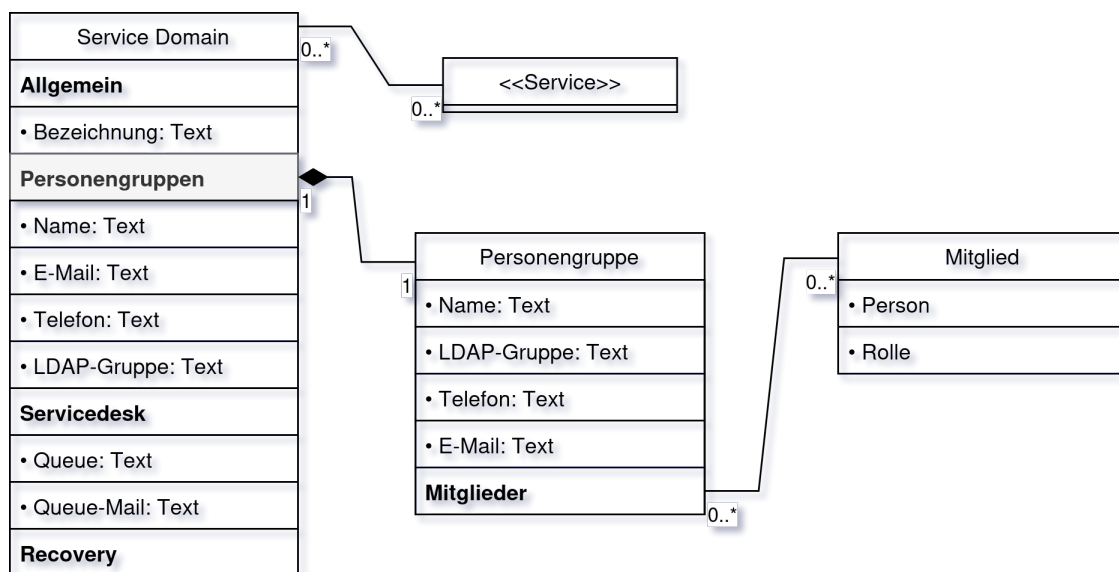


Abbildung 4.4: Klassendiagramm: Service Domain

4.3 Bereitstellungsumgebungen

Im operativen Betrieb spielen nicht nur die Betreuung der produktiven Systeme eine zentrale Rolle, sondern ebenso deren fortlaufende Weiterentwicklung sowie anstehende Migrationsprozesse. Diese Tätigkeiten erfolgen in unterschiedlichen Bereitstellungsumgebungen, etwa in Test-, Entwicklungs- oder Staging-Umgebungen. Auch die in diesen Umgebungen betriebenen Services müssen konsistent erfasst und gepflegt werden. Für die Einbindung solcher Umgebungen existieren verschiedene konzeptionelle Ansätze. Die Auswahl einer konkreten Lösung ist jedoch nicht Gegenstand dieser Arbeit. Stattdessen sollen die im Rahmen dieser Untersuchung gewonnenen Erkenntnisse dazu dienen, in einem Folgeprojekt eine geeignete und tragfähige Lösung zu entwickeln.

4.4 Verschränkung der Hilfssysteme

Ein wesentliches Ziel dieser Arbeit besteht darin, die Grundlage für eine automatisierte Pflege der CMDB und der weiteren Hilfssysteme auf Basis der Daten der CMDB zu schaffen. In diesem Abschnitt wird die konzeptionelle Anbindung des Schwachstellenscanners erläutert.

4.4.1 System-Logik

Alle Server, unabhängig davon, ob es sich um virtuelle Maschinen oder physische Systeme handelt, müssen im Schwachstellenscanner erfasst und regelmäßig überprüft werden. Das verwendete System GSM ist wie folgt konfiguriert (siehe Abbildung 4.5).

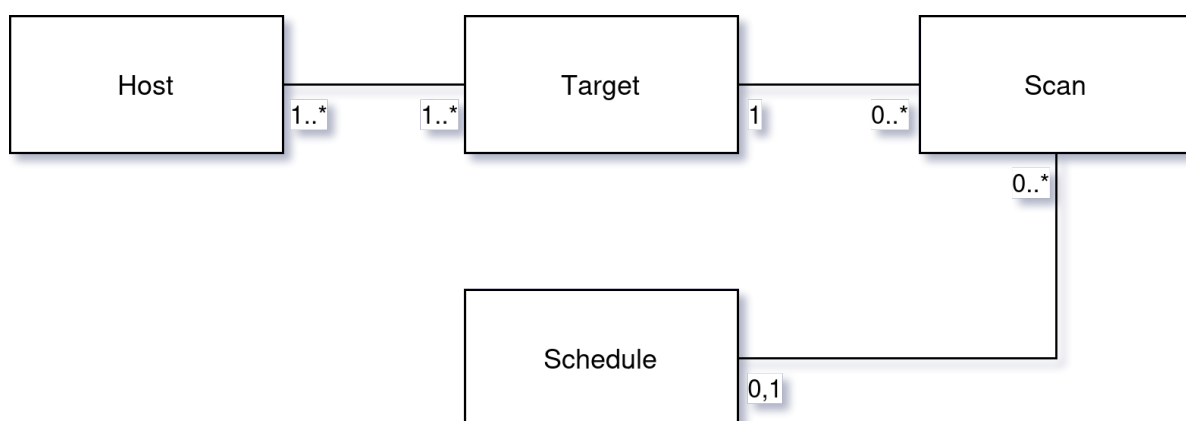


Abbildung 4.5: Schwachstellenscanner-Logik

Zu scannende Maschinen werden als *Hosts* in sogenannten *Targets* zusammengefasst. Ein Host kann hierbei in beliebig vielen Targets enthalten sein. Leere Targets sind nicht

zulässig. Um ein Target scannen zu können, muss es einem *Scan* zugeordnet werden. Ein Target kann von keinem, einem oder mehreren Scans verwendet werden. Scans wiederum müssen zwingend ein Target besitzen. Optional kann einem Scan ein *Schedule* zugewiesen werden, welcher der automatischen Ausführung dient. Ohne Schedule erfolgt ein Scan ausschließlich durch manuelle Aktivierung.

4.4.2 Problem

Aktuell wird der Schwachstellenscanner der SLUB so betrieben, dass alle einem Referat zugeordneten Server in einem referatsweiten Target gesammelt werden. Dieses wird anschließend über einen ebenfalls referatsweiten Scan auf Sicherheitslücken geprüft. Da in solchen Sammel-Scans sehr viele Hosts und damit noch mehr Ergebnisse enthalten sind, werden die erzeugten Berichte schnell unübersichtlich. Zur Auswertung müssen die verantwortlichen Administratorinnen und Administratoren manuell gepflegte Filter einsetzen.

In einem der fünf IT-Referate der SLUB werden die Targets bereits granularer definiert. Diese feineren Targets entsprechen im weiteren Sinne den zuvor vorgestellten Service Domains. Da Service Domains jedoch im bisherigen CMDB-Konzept nicht enthalten sind und entsprechende Informationen fehlen, erfolgt diese Pflege vollständig manuell. Der Prozess ist dadurch zeitaufwendig und fehleranfällig.

4.4.3 Angestrebte Lösung

Auf Grundlage der neu eingeführten Klasse Service Domain soll künftig ein Report erzeugt werden, der alle Hosts (physische und virtuelle), die ihnen zugeordneten Technical Services sowie die jeweils übergeordnete Service Domain ausgibt. In einem ersten Schritt sollen anschließend Targets auf Basis der Service Domain automatisch erzeugt und mit den zugehörigen Hosts befüllt werden. Darauf aufbauend kann für jede Service Domain ein Scan erstellt und das entsprechende Target zugeordnet werden.

Durch diese Vorgehensweise entfällt die manuelle Pflege von Filtern. Die Ergebnisberichte werden schlanker, übersichtlicher und laden schneller. Da einzelne Scans weniger umfangreich sind, können sie zudem besser verteilt werden, was die Last auf dem Schwachstellenscanner reduziert. Darüber hinaus lässt sich die Sicherheit einzelner Service Domains über längere Zeiträume hinweg präziser auswerten.

Der Schwachstellenscanner ermöglicht verschiedene konfigurierbare Automatismen, etwa das Erstellen von Delta-Berichten sowie die darauf basierende automatisierte Generierung von Tickets oder das Versenden von E-Mails. Bei großen Sammel-Scans können diese Funktionen jedoch nicht effektiv eingerichtet oder genutzt werden.

Während die CMDB die benötigten Informationen, etwa Hostname, Technical Service und Service Domain, vollständig bereitstellt, existiert auf Seiten des Scanners keine native Integrationsmöglichkeit. Daher soll ein Host als Middleware etabliert werden, der regelmäßig CMDB-Reports abfragt und auf deren Grundlage die Konfiguration der Hilfssysteme über deren APIs vornimmt (siehe Abbildung 4.6).

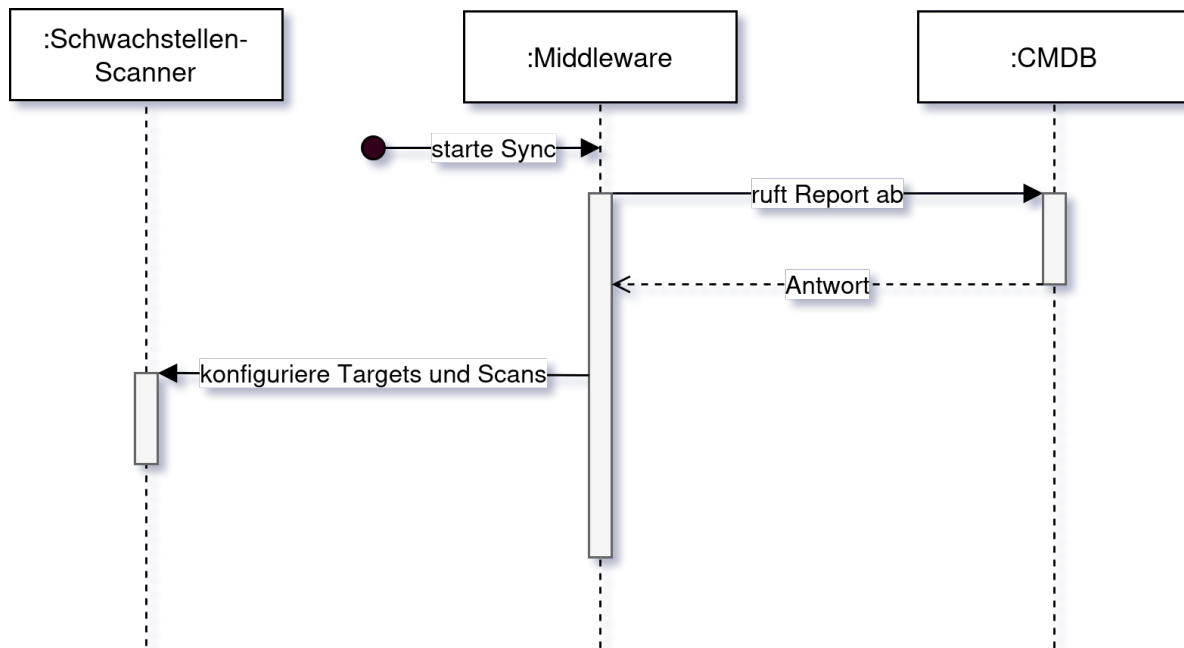


Abbildung 4.6: Sequenzdiagramm: Anlage Scans im Schwachstellenscanner

Die CMDB ermöglicht das Erstellen von Reports über direkte SQL-Abfragen an die zugehörige Datenbank. Die Konfiguration des Schwachstellenscanners erfolgt über die integrierte API. Vorgesehen ist eine tägliche Synchronisation der relevanten Daten.

Kapitel 5

Umsetzung

Die Umsetzung des in Kapitel 4 entwickelten Konzepts lässt sich in zwei zentrale Abschnitte gliedern. Zum einen erfolgte die manuelle Pflege der Service-Struktur in der CMDB auf Basis der zuvor definierten Modellierung. Zum anderen wurde eine Synchronisierungsschnittstelle zwischen der CMDB und dem Schwachstellenscanner implementiert, um exemplarisch die Verschränkung beider Systeme nachzuweisen.

5.1 Pflege der CMDB

Vorbereitung

Für die Modellierung der CMDB wäre eine vollständig leere Instanz der eingesetzten Software i-doit die optimale Ausgangsbasis gewesen. Da jedoch auch für Testsysteme entsprechende Lizenzen erforderlich sind, die bereits durch das bestehende Testsystem belegt waren, konnte keine separate Instanz eingerichtet werden. Ebenso war eine Neuinstallation der vorhandenen Testinstanz nicht möglich, da hierbei umfangreiche Datenbestände – wie Stammdaten, Geschäftsprozesse und technische Informationen – hätten manuell rekonstruiert werden müssen. Um dennoch ohne bestehende Altlasten arbeiten zu können, wurden sämtliche vorhandenen Daten in den Bereichen Services, Server (virtuell und physisch), Beziehungstypen sowie nutzerdefinierte Kategorien vollständig bereinigt. Dadurch konnten die im Folgenden beschriebenen Modellierungsmaßnahmen auf einer möglichst neutralen Grundlage durchgeführt werden, ohne relevante Datenbestände des Testsystems zu verlieren.

Konfiguration

Bevor inhaltliche Daten gepflegt werden konnten, mussten zunächst die in Kapitel 4 spezifizierten Klassen in der CMDB umgesetzt werden. Zu diesem Zweck wurden die Klassen Business Service, Technical Service und Service Domain angelegt. Die daraus abgeleiteten Kategorien und Attribute ließen sich wie geplant und ohne nennenswerte Schwierigkeiten konfigurieren. Die bereits vorhandene Klasse Virtuelle Maschine musste nicht neu implementiert werden, jedoch wurden überflüssige Kategorien und Attribute entfernt, um eine klare und konsistente Modellstruktur zu gewährleisten.

Nach Abschluss der Klassenkonfiguration sowie der Vorbereitung der Dialogoptionen für Kategorien und Attribute wurden die Service Domains DBoD und Shibboleth eingepflegt. Auch diese Modellierung konnte wie vorgesehen realisiert werden. Zur Qualitätssicherung wurde anschließend der integrierte CMDB-Explorer herangezogen, der eine grafische Visualisierung der Daten bereitstellt. Die Darstellung im Explorer basiert auf zwei Einstellungsbereichen: Profilen, die das Erscheinungsbild von Knoten und Beziehungstypen, einschließlich der angezeigten Attribute, definieren, sowie Filtern, mit denen Knoten oder Kanten vollständig aus der Darstellung ausgeschlossen werden können.

Um eine möglichst hohe Vergleichbarkeit zur bestehenden Modellierung im Produktivsystem zu erzielen, wurden sowohl Profile als auch Filter so weit wie möglich analog zu den produktiven Einstellungen konfiguriert. Die Analyse der resultierenden Visualisierung führte zu mehreren Erkenntnissen.

Erkenntnisse

Positiv hervorzuheben ist, dass die hierarchische Darstellung der Modellierung die bestehenden Abhängigkeiten wie vorgesehen transparent macht. Allerdings zeigte sich, dass der CMDB-Explorer nicht in der Lage ist, unterschiedliche Abhängigkeitstypen in der Visualisierung voneinander zu unterscheiden. Die Trennung zwischen technischen und inhaltlich-logischen Beziehungen ist damit zwar technisch abbildbar, jedoch nicht visuell erfassbar. Eine mögliche Abhilfe besteht darin, separate Filter für technische beziehungsweise inhaltlich-logische Beziehungen zu definieren; eine gemeinsame Darstellung beider Beziehungstypen innerhalb eines einzigen Graphen ist jedoch weiterhin nicht möglich.

Ein weiteres Problem ergab sich in Bezug auf die geplante Darstellung der Service Domain am Knoten der Business und Technical Services. Profile können nur vordefinierte Attribute im Explorer anzeigen, unter anderem den „Primären Kontakt“. Um die Service Domain direkt visualisieren zu können, wurde sie als „Personengruppe“

modelliert, wodurch sie grundsätzlich als primärer Kontakt zuweisbar ist – ein Verhalten, das in der Konfiguration erfolgreich getestet wurde. In der grafischen Darstellung des CMDB-Explorers bleibt das entsprechende Feld jedoch leer, sobald es sich beim primären Kontakt um eine Service Domain handelt. Da auch die Darstellung anderer vordefinierter Personengruppen fehlerhaft ist, deutet dies auf einen Fehler im CMDB-Explorer hin. Obwohl die Zuordnung in Listenansichten korrekt erscheint, ist somit eine grafische Visualisierung der Beziehung zwischen Services und Service Domain nicht möglich.

Zur Umgehung dieses Problems wurde den Bezeichnungen der Services ein Kürzel, etwa *DBoD:*, vorangestellt. Diese Lösung ist jedoch fehleranfällig, da sie manuelle Pflege erfordert. Da die Einschränkung ausschließlich die grafische Darstellung betrifft und die technische Auswertung weiterhin uneingeschränkt möglich ist, wird das Problem insgesamt als nicht kritisch bewertet.

5.2 Verschränkung der Systeme

Der im vorangegangenen Kapitel 4 entwickelte Ablauf verdeutlicht, dass die Verschränkung der beteiligten Systeme – zumindest in unidirektionaler Form – aus drei zentralen Schritten besteht. Erstens müssen die erforderlichen Daten aus der CMDB abgefragt werden. Zweitens ist eine Auswertung dieser Daten erforderlich, sofern sie nicht unmittelbar in ihrer vorliegenden Struktur weiterverarbeitet werden können. Drittens erfolgt auf Basis der aufbereiteten Informationen die Pflege der angeschlossenen Hilfssysteme. Diese drei Schritte bilden den Kernprozess der systemübergreifenden Nutzung der in der CMDB gepflegten Daten.

Datenabruf

Die Grundlage für eine weitergehende Nutzung der CMDB-Daten bildet die Fähigkeit, komplexe Reports zu erzeugen und diese über eine API automatisiert auszulesen. Für die exemplarische Umsetzung wurde versucht, ein Report zu erstellen, der sämtliche virtuelle Server einschließlich zugeordneter Technical Services und implizit verknüpfter Service Domains umfasst. Die CMDB i-doit stellt hierfür einen umfangreichen grafischen Abfrage-Editor zur Verfügung, der prinzipiell das komfortable Zusammenstellen von Report-Abfragen ermöglicht.

Da die Service Domain jedoch nicht direkt, sondern ausschließlich über die Technical Services mit den virtuellen Servern verbunden ist, ergibt sich eine verschachtelte Abfragestruktur. Die Komplexität dieser Struktur überschreitet die Möglichkeiten des

grafischen Editors, sodass die gewünschte Darstellung darüber nicht realisierbar war. i-doit erlaubt neben dem grafischen Editor jedoch auch die Erstellung direkter SQL-Abfragen, weshalb für die vorliegende Aufgabe auf dieses Instrument zurückgegriffen wurde.

```
SELECT
  obj_main.isys_obj__id AS '__id__',
  obj_main.isys_obj__title AS 'Host',
  j8.isys_obj__title AS 'Technical Service',
  job9.isys_obj__title AS 'Service Domain'
FROM
  isys_obj AS obj_main
  INNER JOIN isys_cmdb_status AS obj_main_status ON
    → obj_main_status.isys_cmdb_status__id =
    → obj_main.isys_obj__isys_cmdb_status__id
  INNER JOIN isys_obj_type AS j14 ON j14.isys_obj_type__id =
    → obj_main.isys_obj__isys_obj_type__id
  LEFT JOIN isys_connection AS conn_main ON
    → conn_main.isys_connection__isys_obj__id = obj_main.isys_obj__id
  LEFT JOIN isys_catg_its_components_list AS j4 ON
    → j4.isys_catg_its_components_list__isys_connection__id =
    → conn_main.isys_connection__id
  LEFT JOIN isys_obj AS j8 ON j8.isys_obj__id =
    → j4.isys_catg_its_components_list__isys_obj__id
  INNER JOIN isys_catg_contact_list AS j10 ON
    → j10.isys_catg_contact_list__isys_obj__id =
    → j4.isys_catg_its_components_list__isys_obj__id
  AND j10.isys_catg_contact_list__primary_contact = '1'
  INNER JOIN isys_connection AS j9 ON j9.isys_connection__id =
    → j10.isys_catg_contact_list__isys_connection__id
  LEFT JOIN isys_obj AS job9 ON j9.isys_connection__isys_obj__id =
    → job9.isys_obj__id
WHERE
  j14.isys_obj_type__id = '59'
ORDER BY
  job9.isys_obj__title ASC,
  obj_main.isys_obj__title ASC;
```

Abbildung 5.1: Abfrage für Hosts, Technical Service und Service Domain

Die gezeigte SQL-Abfrage erzeugt einen Report, der eine vollständige Auflistung aller virtuellen Server inklusive ihrer jeweils zugeordneten Technical Services sowie der entsprechenden Service Domains ermöglicht.

Aufgrund der Modellierung können virtuelle Server im Ergebnis mehrfach erscheinen, sofern ein einzelner Server mehreren Technical Services zugeordnet ist. Überdies ist gemäß den Modellierungsvorgaben davon auszugehen, dass kein Server unterschiedlichen Service Domains zugeordnet ist. Der Report kann daher genutzt werden, um die Einhaltung dieser Vorgabe systematisch zu überprüfen und gegebenenfalls korrigierend einzugreifen. (siehe Abbildung 5.2)

Report-Vorschau			×
Host	Technical Service	Service Domain	
sdvdbodfile	DBoD: File	DBoD	
sdvdbodbranch01	DBoD: Citrix-VDC	DBoD	
sdvdbodbranch02	DBoD: Citrix-VDC	DBoD	
sdvdbodts01	DBoD: IB-Stat	DBoD	
sdvdbodts02	DBoD: IB-Stat	DBoD	
sdvdbodts03	DBoD: IB-Stat	DBoD	
sdvdbodts04	DBoD: IB-Stat	DBoD	
sdvdbodts05	DBoD: IB-Stat	DBoD	
sdvdbodts06	DBoD: IB-Stat	DBoD	

Abbildung 5.2: Report-Vorschau für Service Domain-Hosts

Datenaufbereitung und Pflege der Hilfssysteme

Die im Rahmen der Diplomarbeit vorgesehene automatisierte Pflege von Targets und Scans im Schwachstellenscanner auf Basis der aus der CMDB abgerufenen Daten konnte aus zeitlichen Gründen nicht realisiert werden. Da jedoch bereits bestehende Automatisierungsprozesse zur Pflege dieses sowie weiterer Hilfssysteme vorhanden sind, die auf alternativen Datenquellen operieren, kann davon ausgegangen werden, dass die geplante Integration der Systeme umsetzbar ist. Die hierfür erforderlichen Schritte entsprechen in ihrer Struktur weitgehend den etablierten Verfahren, sodass die Integration der CMDB-Daten in diese Prozesse voraussichtlich ohne grundlegende konzeptionelle Anpassungen möglich ist. Die Realisierung dieser Funktionalitäten sowie weiterer noch ausstehender Arbeitspakete ist für zukünftige Projekte vorgesehen.

5.3 Fazit

Abschließend lässt sich festhalten, dass die im Rahmen dieses Kapitels dargestellten Schritte die praktische Umsetzbarkeit des in Kapitel 4 entwickelten Konzepts verdeutlichen. Sowohl die Modellierung und Pflege der CMDB als auch die exemplarische Verschränkung mit einem Hilfssystem zeigen, dass die grundlegenden Anforderungen erfüllt werden konnten, auch wenn einzelne Funktionalitäten aufgrund technischer Einschränkungen oder begrenzter Ressourcen nur teilweise umgesetzt wurden. Die gewonnenen Erkenntnisse bilden zugleich eine wesentliche Grundlage für weiterführende Arbeiten in zukünftigen Projekten. Kapitel 6 zieht daraus ein übergeordnetes Fazit und reflektiert die Ergebnisse der Arbeit im Hinblick auf ihre Zielsetzung und ihren Beitrag für zukünftige Entwicklungen.

Kapitel 6

Fazit

In Kapitel 1 wurden zunächst die zentralen Probleme des bestehenden ITSM-Systems beschrieben und daraus die Ziele dieser Arbeit abgeleitet. Im vorliegenden Fazit werden diese Problemstellungen rekapituliert und daraufhin überprüft, ob sie zum aktuellen Stand weiterhin bestehen. Gleiches gilt für die abgeleiteten Ziele und wie weit diese erfüllt wurden. Darüber hinaus wird auf die Probleme eingegangen, die sich erst im Verlauf der Arbeit ergeben haben. Weiterhin wird erläutert, wie diese gelöst werden konnten oder wie deren Auftreten hätte vermieden werden können. Abschließend erfolgt ein Ausblick auf potenzielle Folgeprojekte, die sich aus den Ergebnissen dieser Arbeit ableiten.

6.1 Ergebnisse

Ein zentrales Problem der Ausgangssituation bestand in der unzureichenden Datenqualität der CMDB. Diese konnte im Rahmen der Arbeit deutlich verbessert werden, insbesondere durch die Entfernung redundanter und obsoleter Datensätze. Zusätzlich führten die Einführung neuer Kategorien und Attribute zu einer spezifischeren und kontrollierbareren Datenlage. Als exemplarisches Ergebnis ist die neu eingeführte Kategorie „Links“ hervorzuheben, die zuvor verteilte Attribute und Kategorien konsolidiert und darüber hinaus, durch die Typisierung der vorgehaltenen URLs, eine systematische Überprüfung ermöglicht.

Ein weiteres identifiziertes Problem betraf die einseitige Betrachtung der Beziehungen zwischen Services und Servern. Die Einführung des Beziehungstyps „Inhalt/Logik“ stellt einen ersten Schritt dar, diese Beziehungen nach ihrer inhaltlichen Natur zu gruppieren und damit die Analyse komplexer Strukturen, beispielsweise im Incident- oder Change-Management, zu unterstützen.

Gleichzeitig muss berücksichtigt werden, dass die im Rahmen dieser Arbeit konzipierten Datenmodelle und Strukturen nicht zwangsläufig alle zukünftigen Anforderungen abdecken werden. Ein ITSM-Konzept ist stets als dynamisch zu verstehen und muss regelmäßig evaluiert und an neue Gegebenheiten angepasst werden muss.

Das Ziel der Erstellung eines umfassenden Konzepts konnte vollständig erreicht werden. Die Umsetzung weist jedoch einige Schwächen auf, sodass dieses Ziel insgesamt nur als teilweise erfüllt gelten kann. Trotz dieser Einschränkungen stehen die erzeugten Daten dem produktiven System, insbesondere in Bezug auf den System-Komplex DBoD, sowohl in Qualität als auch Quantität in nichts nach. Darüber hinaus sind sie strukturierter und leichter auffindbar. Verantwortlichkeiten sind insbesondere durch die Einführung der Service Domain klar erkennbar.

Die in Kapitel 5 beschriebenen Programmfehler sowie fehlenden Funktionen der eingesetzten CMDB behindern allerdings die vollständige Nutzung der Daten mittels des CMDB-Explorers. Gleichwohl funktioniert die maschinelle Auswertung, basierend auf dem in Kapitel 5 entwickelten Report, wie vorgesehen. Die abschließende Integration des Schwachstellenscanners hätte Teil der Umsetzung sein sollen, um die Einbettung der Lösung in die bestehende Systemlandschaft exemplarisch nachzuweisen. Aufgrund zeitlicher Begrenzungen konnte dieser Schritt nicht mehr realisiert werden, weshalb die praktische Umsetzung insgesamt nur teilweise als erfolgreich zu bewerten ist.

Unabhängig von den definierten Zielen und Problemen ist außerdem die durchgeführte Befragung zu erwähnen. Im Rückblick erweist sich die gewählte Methode als nicht optimal, da ein hoher Anteil an Freitextfeldern und ein zu kleiner Teilnehmerkreis zu einer geringen Ausbeute verwertbarer Ergebnisse führte. Für vergleichbare zukünftige Projekte sollte daher eine alternative Befragungsmethode, wie etwa persönliche Interviews, gewählt werden.

6.2 Ausblick

Bereits vor Beginn der Arbeit war absehbar, dass die umfassende Neukonzeption des ITSM an der SLUB nicht vollständig in einem einzelnen Projekt bearbeitet werden kann. Folglich werden dieser Arbeit mehrere Folgeprojekte angeschlossen sein. Zunächst betrifft dies die Evaluierung der erzielten Ergebnisse und deren Überführung in die Produktivumgebung der SLUB. Hierfür ist die Einbindung aller Mitarbeitenden der IT-Abteilung notwendig und ein entsprechendes Projekt muss initiiert werden.

Ein weiteres zukünftiges Projekt umfasst die vollständige Verschränkung aller Hilfsysteme. Dabei sind insbesondere Automatisierungsmechanismen zur Pflege der CMDB zu analysieren und zu etablieren. Neben der Entwicklung neuer Konzepte und techni-

scher Mechanismen ist es essenziell, Prozesse im Kontext des ITSM zu erstellen oder anzupassen. Hierzu müssen bestehende Abläufe evaluiert, überarbeitet und ergänzt werden.

Ein Teil dieser Prozesse, insbesondere die regelmäßige Evaluierung des Datenmodells, kann inhaltlich bereits definiert werden. Die in dieser Arbeit nicht vertieft behandelten Bereiche, etwa Bereitstellungsumgebungen, extern erbrachte Services und physische Server, sind zeitnah zu untersuchen und in das Datenmodell zu integrieren.

Um sowohl die in dieser Arbeit erstellten Konzepte sowie zukünftige Änderungen am ITSM angemessen umzusetzen und die CMDB allen Beteiligten näherzubringen, sollen Nutzerhandbücher für die CMDB erstellt und Schulungen durchgeführt werden.

Abschließend ist geplant, die identifizierten Programmfehler der eingesetzten CMDB sowie die aus Sicht der SLUB fehlenden Funktionen des CMDB-Explorers an den Hersteller zu kommunizieren, um langfristig eine Verbesserung der CMDB zu erreichen.

Anhang A

Verwendung von LLMs zur Formulierung

Nachfolgend findet sich die Eingabe und Ausgabe für das Kapitel 3. Die Ausgabe wurde anschließend überarbeitet.

A.1 Eingabe

Kapitel 3 - Analyse der bestehenden IT-Landschaft

- * Es wird betrachtet, welche Systeme zur Dokumentation und Überwachung
 - der IT-Landschaft bestehen, welche Daten durch diese vorgehalten und
 - verwaltet werden, wie diese zusammenhängen und wie sie sich in die
 - restliche IT-Landschaft einfügen

Bestehende Systeme zur Dokumentation

CMDB

- * Als integrierte Lösung für ITSM und CM, kommt i-doit als CMDB zum
 - Einsatz
- * Es ist der zentrale Knotenpunkt für das ITSM an der SLUB
- * Idoit ist eine Softwarelösung für, unter anderem, ITSM, ITAM, CM, das
 - Erstellen von Betriebshandbüchern, verwalten von Stammdaten,
 - Automatisierung, betrachtung von Netzwerkplänen [zitat-stelle mit
 - schlüssel ``idoit``]

- * Viele weitere Funktionen stehen, nicht zuletzt durch Erweiterungen, zur
→ Verfügung
- * So wird etwa der BSI-Grundschutz für die SLUB mittels einer solchen
→ Erweiterung verwaltet
- * Die Möglichkeit zum Erstellen komplexer Berichte über die vorgehaltenen
→ Daten, in Kombination mit der vorhanden API, lassen viel Spielraum
→ zur integration weiterer Systeme und Mechanismen

Live-Monitoring

- * Festplattenspeicher kann durch eine Fehlkonfiguration oder
→ unvorhergesehene Ereignisse ausgehen, Zertifikate ablaufen, die
→ Antwortzeit eines Webservers zu hoch ausfallen
- * Diese Ereignisse bilden nicht mal einen Bruchteil der möglichen
→ Vorkommnisse ab
- * Komplexe IT-Landschaften müssen daher stetig auf ihren
→ Gesundheitszustand überwacht werden
- * An der SLUB kommt hierfür das System checkmk
- * Checkmk bietet viele Optionen die eigene IT-Landschaft zu überwachen.
→ Ganz gleich ob diese in einer extern gehosteten Cloud liegt oder dem
→ eigenen Rechenzentrum. Egal ob physische Server, virtuelle Maschinen
→ oder Container [zitat-stelle mit schlüssel `cmk`]
- * Zahlreiche Plugins [Glossar-Eintrag ergänzen] dienen dazu, die
→ Überwachungsfähigkeiten zu erweitern und zu präzisieren
- * In checkmk ist häufig von services die rede. Im Vergleich zu ITSM sind
→ hierbei jedoch einzelne Softwarekomponenten, wie etwa ein einzelner
→ Webservice, gemeint
- * Sowohl Plugins als auch Services, welche häufig hand-in-hand bestehen,
→ können selbst entwickelt und integriert werden
- * Darüber hinaus verfügt checkmk über eine umfängliche API, über welche
→ sich Automatisierungen und Verknüpfungen zu anderen Systemen
→ realisieren lassen

Schwachstellen-Überwachung

- * Angriffe auf IT-Systeme sind, sowohl im privaten als auch im
→ geschäftlichen Umfeld, eine permanente Bedrohung

- * Je größer eine IT-Landschaft wird, desto komplexer fällt diese meist
 - aus
- * Mit einem hohen Grad an Komplexität sinkt der Überblick und
 - gleichzeitig häufen sich die Angriffsvektoren und damit die Risiken
 - im Betrieb
- * Um den Überblick über diese Angriffsvektoren zu behalten und die Risiken zu minimieren, muss die IT-Landschaft auf Sicherheitslücken überwacht werden
- * An der SLUB kommt hierfür der automatische Schwachstellenscanner
 - Greenbone Security Manager (GSM) zum Einsatz
- * Der GSM wird für definierte Zielgruppen (Sammlungen von physischen und virtuellen Maschinen) konfiguriert und scannt diese nach vorgegebenen Zeitplänen auf bekannte Sicherheitslücken
- * Aus diesen Scans ergeben sich Berichte, welche eine punktuelle und chronologische Auswertung des Sicherheitszustands eines oder mehrere Systeme erlauben
- * Über Berichte, automatische Benachrichtigungen sowie die integrierte API, lassen sich Prozesse und Integrationen in andere Systeme realisieren
- * Unter anderem werden solche Berichte an ein Ticket-System geschickt, um diese dem verantwortlichen Team zur Bearbeitung zu überlassen

Ticket-System

- * Als Ticket-System kommt Znuny zum Einsatz
- * Es handelt sich hierbei um einen Fork [Glossareintrag ergänzen] der vormals freien Software OTRS
- * Im Kontext von ITSM, bildet das Ticket-System den Service Desk ab, welcher zur Erfassung und Bearbeitung von Anfragen (Störungen, Zwischenfällen, Kontaktanfragen) zu Services Dient [zitat-stelle mit schlüssel ``idoit`` - Seite 149]
- * Im Ticket-System werden Anfragen in sogenannten Queues gepflegt, Bearbeitern zugewiesen, Handlungen dokumentiert und der Status der jeweiligen Anfrage erfasst
- * Auch Znuny bietet eine API zur Integration in andere Systeme

Code-Repository

- * In modernen DevOPS-Methoden, wie etwa Software Defined Networking (SDN)
 - oder Infrastructure as Code (IaC) [Glossar-Einträge ergänzen] liegen
 - teile der IT-Landschaft als Code vor
- * Oft ist dieser Code, etwa bei der Orchestrierungslösung
 - [Glossar-Eintrag ergänzen] Ansible selbst-dokumentierend und bedarf
 - keiner weiteren ausführlichen Dokumentation
- * In diesem Zusammenhang ist ein Code-Repository, zur Speicherung und
 - Verwaltung dieser Lösungen, als Dokumentations-System zu verstehen
- * In der SLUB kommt ein selbst gehotetes GitLab zum einsatz
- * GitLab ist eine integrierte Lösung zur Softwareentwicklung, welche auf
 - der freien Versionierungssoftware Git aufsetzt
- * Gitlab ergänzt Git hierbei um zahlreiche Funktionen im Bereich
 - Projektmanagement (etwa Kanban), Ticketbearbeitung, Dokumentation
 - sowie um DevOps-Funktionalitäten wie etwa CI/CD [Glossar-Eintrag
 - ergänzen] und Release-Management
- * Wie schon die vorhergehenden Systeme, bietet auch GitLab eine
 - umfangreiche API

Virtualisierungsumgebung (VE)

- * Eine Virtualisierungsumgebung, im Falle der SLUB VMware, ist weder ein
 - dediziertes Monitoring- noch Dokumentationssystem
- * Gleichzeitig hält eine VE mindestens rudimentäre Monitoring-Daten zu
 - den gehosteten VMs vor
- * Noch wichtiger ist der Aspekt, dass eine VE besonders bei der Erzeugung
 - neuer VMs oder Containern, einige Informationen bereithält, die in
 - einem Configuration Management von relevanz sind
- * An der SLUB ist die überwiegende Mehrheit der Server virtualisiert.
 - Etwa 89 \% (Stand 30.06.2025)
- * Auch hier, die Erwähnung ist fast obligatorisch, existiert eine API

Intranet (nach hinten verschieben)

- * Abschließend soll auch das Intranet genannt werden
- * Es kommt Confluence im self-hosting zum Einsatz

- * Es handelt sich um eine enterprise intranet Lösung, die nicht nur, aber
 - besonders durch die Integration des Menschen hervorsteicht. Das
 - Stichwort lautet social Intranet
- * Nicht zuletzt durch die sozialen Aspekte, hat das Intranet an der SLUB
 - einen äußerst hohen Nutzungsgrad erreicht
- * Praktisch jede Abteilung und alle Angestellten, verwenden das Intranet
 - in ihrer täglichen Arbeit
- * Es handelt sich bei einem Intranet jedoch weniger um ein klassisches
 - Dokumentations- oder Monitoring-System
- * Der Umstand, dass Inhalte im Freitext erstellt und verwaltet werden,
 - also keine zwangsläufig strukturierten Daten vorliegen, gestaltet die
 - Verwendung in einem automatisierten Umfeld als problematisch
- * Trotzdem erlaubt ein Intranet, gerade durch seinen Freitext-Charakter,
 - Informationen vorzuhalten, die auf die Verwendung für Menschen
 - abzielen
- * Das Intranet findet vor allem ergänzende Anwendung. Es werden
 - Informationen dargestellt die in den anderen Systemen schlecht oder
 - gar nicht dargestellt werden können. Beispiele wären hierfür
 - Nutzerhandbücher, Projekt-Dokumentationen
- * Über die bestehende API des Intranet werden schon heute zahlreiche
 - Berichte automatisch, auf Grundlage von Informationen aus anderen
 - Systemen, erstellt

VM Provisioning

- * Eine an der SLUB selbst entwickelte WebApp dient dazu, neue VMs zu
 - erzeugen, genannt grant unified boot server (GUBS)
- * GUBS setzt hierbei vorgegebene Parameter und Konfigurationen durch
- * Weiterhin etabliert GUBS automatisierte Prozesse, um die Dokumentations-
 - und Monitoringsysteme mit den neuen VMs zu befüllen
- * GUBS selbst ist weder Dokumentations- noch Monitoringssystem, bildet
 - jedoch den Einstieg des Lifecycles eines virtuellen Servers und muss
 - daher mitbetrachtet werden

Bestehendes ITSM-Konzept

Zugrundeliegende Definitionen

- * Das i-doit dient als Kernelement des ITSM und als CMDB
- * Das bisherige Konzept zur Abbildung von Services und ihrer
 - Konfiguration sieht vier verschiedene Service-Arten vor: Business
 - Service IT, Infrastructure Service, Technical Service IT und
 - Technical Service IT extern
- * Weitere Knoten-typen sind der Geschäftsprozess, virtuelle Server,
 - physische Server, Netzlaufwerke und einige seltener vorkommende, die
 - hier nicht näher betrachtet werden sollen
- * Als Abhängigkeitstyp kommt, abseits von Beziehungen zu personellen oder
 - anderen Stammdaten, fast ausschließlich "Service-Komponente" vor
- * Die Services bilden den Kern des ITSM und sind wie folgt definiert und
 - verknüpft
- * Business Service (BS):
 - * Für Kunden (intern und extern) erbrachte Leistung, die Namentlich von
 - diesen benannt werden kann
 - * Sind mindestens einem Geschäftsprozess (GP) zugeordnet. Dienen zur
 - Zielerfüllung des GP
 - * Können nur Technical Service als untergeordnete Knoten aufweisen -->
 - Umkehrschluss: BS können nicht auf andere BS zeigen
 - * Als Beispiele und zur späteren Referenz seien die folgenden BS der
 - SLUB erwähnt:
 - * Datenbankinformationssystem DBoD: System zur ortsunabhängigen
 - Präsentation von E-Ressourcen (E-Books, Anwendungen,
 - Online-Datenbanken) mit Rechtedurchsetzung
 - * Authentifizierung (Shibboleth): Stellt einen SAML-Identity Provider
 - basierend auf der Serverlösung Shibboleth zur Verfügung. Dient
 - für viele Services der SLUB als Authentifikations-Backend. So
 - auch für DBoD
- * Technical Service (TS):
 - * Bindeglied zwischen BS, weiteren TS und Geräten (etwa Server und
 - virtuelle Maschinen)
 - * Erbringt technische Funktionalitäten um den BS zu ermöglichen
 - * TS sind als Knoten immer unter BS oder anderen TS (intern und extern)
 - angeordnet
 - * Ts können als Kanten nur andere TS und Geräte untergeordnet haben
- * Technical Service extern (TSe):

- * Durch externe Anbieter erbrachte technische Funktionalitäten, die zur
 - Erbringung eines BS dienen
- * Sind analog zu TS zu betrachten
- * Können keine Endgeräte untergeordnet haben
- * Infrastructure Service (IS):
 - * Services, die von fast allen Geschäftsprozessen genutzt werden
 - * Nutzung wird als allgemein bekannt angenommen
 - * Vorhande Verbindungen zwischen IS und anderen Services werden im CM
 - der SLUB meist nicht eingetragen. Es wird angenommen, dass fast
 - alle den IS nutzen. Die Tatsächliche Nutzung bleibt daher aus ITSM-
 - und CMDB-Sicht unsichtbar
 - * Beispiele:
 - * Zentraler Speicher: Hier verbirgt sich das Disk-Speichersystem der
 - SLUB, über welches das SAN, NFS-Shares sowie SMB-Shares verwaltet
 - und angeboten werden
 - * Intranet: Wird in der SLUB unter Infrastruktur gelistet, was mit
 - der Definition, dass fast alle GP diese nutzen, zwar konform ist,
 - jedoch intuitiv nicht als Infrastruktur verstanden wird

Zusammenwirken der bestehenden Dokumentations- und Monitoringsysteme

- * Mit CMDB, Intranet, Monitoring, VE, Schwachstellenscanner,
 - Code-Repository und Ticket-System, steht eine Vielzahl von Systemen
 - bereit, um Daten im Sinne des ITSM und CM zu erschließen, zu
 - verarbeiten und vorzuhalten
- * Die Systeme sind nicht von Haus aus integriert und müssen durch
 - geeignete Maßnahmen, wie etwa Anbindungen über APIs, vernetzt werden
- * Nachfolgend wird aufgezeigt, wie die Systeme miteinander Interagieren.
 - Hierfür wird der Lifecycle eines virtuellen Servers innerhalb der
 - IT-Landschaft skizziert und anhand dessen die Informationsflüsse und
 - Mechanismen erläutert

[Grafik einfügen]

- * Nachdem ein Server über GUBS konfiguriert ist, erzeugt dieser über die
 - in VMware integrierte API eine neue VM und startet die automatische
 - Installation per PXE [glossar-eintrag hinzufügen]

- * VMware meldet grundlegende Informationen bezüglich der VM zurück an die
 - Bereitstellungsplattform, die genutzt werden, um die neue VM den
 - anderen Systemen gegenüber bekannt zu machen
- * Die neue VM wird im Monitoring sowie im Schwachstellenscanner angelegt
- * Das Monitoring wird auf fünf sogenannten Sites aufgeteilt, wobei jede
 - Site für jeweils eines der fünf IT-Referate vorgesehen ist
- * Die Server werden hier direkt im Root-Bereich der Site angelegt
- * Für den Schwachstellenscanner stehen fünf Scan-Konfigurationen zur
 - Verfügung
- * Der neu erstellte Server wird dem Scan des Referats zugeordnet, dem der
 - Ersteller des SServers angehört
- * Sofern während der initialen Konfiguration via GUBS der Server als
 - Testsystem erstellt wurde, wird der Server in der CMDB automatisch
 - einem Test-Service zugewiesen, der alle Testsysteme unter sich
 - gruppiert
- *

Probleme mit dem bisherigen ITSM-Konzept

- * Das bestehende Konzept ist weitestgehend ITIL-konform
- * Gleichzeitig entstehen Probleme, wenn das Konzept so umgesetzt wird,
 - wie durch die Definitionen vorgesehen
- * Nachfolgend [Bild einfügen] findet sich eine stark vereinfachte
 - Darstellung des Business Service DBoD, inklusive der angrenzenden
 - Systeme. Es soll als Anschauung dienen, um die bestehenden Probleme
 - aufzuzeigen
- * Als zentraler Punkt steht der Business Service DBoD. Es ist der
 - zentrale Knotenpunkt des Produkts DBoD
- * In oberster Reihe, der Vollständigkeit halber, sind die
 - Geschäftsprozesse zu sehen, welche den BS DBoD zur Erfüllung
 - benötigen
- * Vom BS abzweigend, finden sich mehrere Technical Services. Einer davon
 - wird detaillierter visualisiert, der DBoD-Webservice
- * Damit der Webservice funktionieren kann, ist der Technical Service
 - Shibboleth-IdP notwendig. Die Abhängigkeit ist entsprechend zu sehen
- * Die Überordnung des BS Authentifizierung (Shibboleth) ist hier von der
 - Seite angedeutet

- * Weiterhin braucht der TS Shibboleth-IdP den TS Libero, um zu
 - funktionieren
- * Es ergibt sich ein nach unten verzweigter Abhängigkeitsbaum
- * Das Problem mit dieser Darstellung ist, dass hier innerhalb weniger
 - Schritte bereits zwei Verantwortlichkeitsgrenzen überschritten werden,
 - ohne dass diese Information dem Graphen eindeutig entnommen werden
 - kann
- * So wird etwa der Shibboleth-IdP von einem anderen Team betrieben
- * Mit dem TS Libero besteht die gleiche Problematik, auch hier ist ein
 - gänzlich anderes Team verantwortlich
- * Für einen BS und die direkt damit einhergehenden Maschinen und Services,
 - ist für gewöhnlich ein festes Team zuständig
- * Es wird zwar auf Services anderer Teams zurückgegriffen, jedoch ist die
 - Verantwortlichkeit hier entsprechend bei anderen Teams gelagert
- * Das aktuelle Configuration Management liefert keine Informationen, die
 - es einem ohne Vorwissen erlauben, die Services und Maschinen eines BS
 - zu erfassen, die vom gleichen Team bearbeitet werden
- * Auch kann diese Information nicht aus den den Services zugeordneten
 - Personal abgelesen werden. So findet sich etwa der Kollege Müller
 - sowohl im Team DBoD, als auch dem Team Shibboleth wieder
- * Es kann ohne weitere Informationen schnell angenommen werden, die
 - Services gehörten eng zusammen
- * Das einzige Wissen, das sicher aus den Informationen im CM abgeleitet
 - werden kann, ist die Zugehörigkeit zu einem Referat
- * Da es aber auch Services und Teams gibt, die Referatsübergreifend
 - verwaltet werden, ist auch dieses Wissen nur bedingt von Nutzen
- * Aus dem Informations- und Wissensmangel leiten sich einige Nachteile in
 - der aktuellen Automatisierung ab
- * Es werden Server etwa automatisiert im Monitoring angelegt, jedoch
 - ausschließlich einem von 5 Hauptgruppen zugeordnet
- * Diese Gruppen entsprechen den fünf an der SLUB existierenden
 - IT-Referaten
- * Die Sichtbarkeit der Server, sofern an diesen Hauptgruppen verankert,
 - ist stark eingeschränkt. Nur der Referats-Admin sowie die
 - Referats-Leitung können diese Server sehen und überwachen

- * Um die Server für die Verantwortlichen Teams sichtbar zu machen, ist
 - ein manuelles Eingreifen notwendig. Die Automatisierung mündet also
 - in manueller Arbeit
- * Ähnlich gelagert ist es auch bei den Schwachstellenscans. Auch hier
 - werden die Server der entsprechenden Referats-Gruppe zugeordnet
- * Zwar ist hier die Sichtbarkeit für alle Mitglieder des Referats
 - gewährleistet, will jedoch ein Team die eigenen Server granularer
 - verwaltet, etwa durch eine Bündelung in einer dem Service
 - entsprechenden Gruppe, muss dies händisch vorgenommen werden
- * im VMware, präziser dem Frontend-System VSphere, über das VMs auf der
 - VE verwaltet werden können, setzt sich die Problematik fort
- * Server werden hier dem Nutzer, welcher den Server per GUBS angefordert
 - hat, direkt zugeordnet
- * Besteht der Anspruch eines Teams, die eignen Server gebündelt in einem
 - Verzeichnis zu sammeln, um etwa Rechte auf diesen besser zu
 - definieren oder Server zusammenzufassen (VMware vApps), muss dies
 - erst händisch durch einen VE-Admin mit entsprechenden Rechten
 - eingerichtet werden
- * In der CMDB sind die Auswirkungen am gravierendsten. Während Testsysteme
 - einem entsprechenden Service automatisch zugeordnet werden, werden
 - produktive VMs nur grundlegend im System angelegt
- * Es wird keinerlei automatische Abhängigkeitszuordnung vorgenommen. Der
 - entsprechende Knoten im Configuration management, hängt in der Luft
- * Die Verantwortlichen müssen hier wieder manuell eingreifen
- * Durch die häufigen manuellen Tätigkeiten ergibt sich ein hohes
 - Fehlerpotential in der Pflege der Systeme
- * Weiterhin muss auch der zeitliche Aufwand als Personelle Kosten
 - betrachtet werden

Vorgehaltene Daten

- * Die verschiedenen Systeme halten eine Vielzahl von Informationen vor
- * Nicht alle davon sind für das ITSM und CM von belang. Die nachfolgende
 - Auflistung ist daher nicht als komplett zu betrachten
- * Es werden die Daten angegeben, die als besonders relevant zur Lösung
 - der beschriebenen Probleme angesehen werden

- * Nachfolgend sind die Daten, nach Hilffssystemen aufgeschlüsselt, zu
 - finden
- * Die Verknüpfung zwischen den Systemen erfolgt immer über den Bezeichner,
 - welcher für Server der hostname ist und für Services der
 - entsprechende Name

CMDB

- * Da innerhalb der CMDB nicht nur eine Klasse von Objekten abgebildet
 - wird, werden diese einzeln betrachtet und die etwaigen Verbindungen
 - beschrieben
- * Zuerst werden die den Klassen zugeordneten Kategorien (Sammlungen von
 - Attributen) mittels einer Matrix aufgelistet
- * Anschließend werden die einzelnen Kategorien und deren Attribute auf
 - Ihren Inhalt und Zweck hin analysiert

Kategorien und Attribute

- * Übergreifend gepflegt werden die Kategorien Allgemein und
 - IT-Grundschutz
- * Aus der Kategorie Allgemein der Bezeichner sowie der CMDB-Status
 - gepflegt und nachgenutzt. Jedes Objekt hat einen Bezeichner, welcher
 - innerhalb der Klasse eindeutig sein soll. Technisch erlauben die
 - genutzten Systeme eine Mehrfachvergabe
- * Der CMDB-Status soll anzeigen, in welchem Zustand sich das Objekt aus
 - CMDB-Sicht befindet. Etwa "in Arbeit" oder "in Betrieb"
- * Dieser Status dient damit der Überprüfung, wie weit die Erfassung des
 - Service fortgeschritten ist
- * Der Status wird manuell gepflegt
- * Kategorie "IT-Grundschutz (Zielobjekte) * ist eine Kategorie, welche
 - vom installierten Plug-in für die BSI-Durchsetzung etabliert wird
- * Da diese Kategorie Ihrem Inhalt nach vollständig in den IT-Grundschutz
 - fällt, welcher in dieser Arbeit nicht betrachtet wird, soll hier nicht
 - im Detail auf diese Kategorie eingegangen werden
- * Die Kategorie "Externe Partner-Accounts" enthält eine Auflistung der
 - Einrichtungen, welche Zugriff auf den Service über spezifische
 - Nutzerkonten haben

- * Die Liste wird vordefiniert
- * Die Kategorie "Service-Abschaltung" beinhaltet Informationen zu
 - Abschaltanweisungen und-Reihenfolge. Etwa ob die Abschaltung auf
 - Grundlage der i-doit-Abhängigkeiten automatisch ermittelt oder über
 - die Vrtualisierungsplattform abgefragt werden soll. Weiterhin ist ein
 - Prüfdatum zu erfassen
- * Die Kategorien "Datenschutz", "Servicedokumentation TS",
 - "Servicedokumentation (Intranet)" und "Servicedokumentation
 - (Hotline)" beinhalten alle Attribute vom Typ "Link", welcher das
 - hinterlegen von URLs gestattet. "Servicedokumentation TS" entspricht
 - hierbei inhaltlich der "Servicedokumentation (Intranet)"
- * "Servicedokumentation (Hotline)" beinhaltet einen Link zu einer nach
 - einem festen Schema angelegten Hotline-Hilfsseite
- * Unter "Datenschutz" findet sich ein weiterer Link zum
 - Verfahrensverzeichnis nach DSGVO
- * Die Kategorie "Service" ist eine vom System mitgelieferte Sammlung
 - mehrerer Kategorien, welche allgemeine Attribute für Services vorhält.
 - Die wichtigsten Attribute sind hierbei
 - * "Business Unit", für die Abteilung, der der Service zugeordnet wird
 - * "Service-Kategorie", "Service Art" sowie "Service Typ", welche Ihrer
 - Natur nach alle dientisch sind. Es kann jeweils ein Auswahldialog
 - mit einfachem Eintrag gepflegt werden. Die Auswahl kann selbst
 - definiert werden. Aktuell wird keines dieser Attribute genutzt
 - * Weiterhin wird, sofern gepflegt, der Service Manager angezeigt sowie
 - * Attribute für eine Servicebeschreibung extern und intern geboten
 - * Darüber hinaus können unter "Service Komponenten" Objekte dem Service
 - untergeordnet werden. Diese werden dann mit der System-spezifischen
 - Beziehungsart "Serive Komponente" verbunden
- * "Servicemerkmale1 SLUB" enthält
 - * Einen Link zur Service-URL, falls es sich um einen Web-Service
 - handelt
 - * Eine kurze Service-Beschreibung
 - * Ein Startdatum, wann der Service erstmals in Betrieb genommen wurde
 - * Eine fachliche Abteilungen und Referats-Zuordnung

- * "Servicemerkmale3 TUD" enthält Informationen, welche nicht für das
 - eigene ITSM benötigt werden. Es handelt sich um eine Aufstellung von
 - Attributen, welche der Technischen universität Dresden übermittelt
 - werden, um in deren Servicekatalog abgebildet zu werden
- * "BearbeiterIn": Eine Personen-Verknüpfung für Ansprechpartner
- * "Betreiber": Für die SLUB immer mit "SLUB" befüllt
- * "Langbeschreibung"
- * "Kurzbeschreibung"
- * "Kapazität"
- * "Schutzbedarf": Der nach BSI festgelegte Schutzbedarf. Ob für die
 - Integrität, Verfügbarkeit oder Vertraulichkeit ist nicht näher
 - definiert
- * Englischer Titel, englische Kurz- und Langbeschreibung
- * "Katalog-Tags": Eine Mehrfachauswahl vordefinierter Schlagwort,
 - welche den Service beschreiben
- * "Kundengruppe": Eine Mehrfachauswahl vordefinierter Werte, etwa
 - "Studenten", "Beschäftigte" oder "Gäste"
- * Informationen über die Kostenpflichtigkeit und falls gegeben die Höhe
 - der Kosten
- * Eine Service-URL, falls es sich um einen Webservice handelt
- * Betriebszeit
- * Supportzeit
- * Verfügbarkeit
- * Eine Personenzuordnung für den First-Level-Support
- * Die "Vertragszuweisung" ist eine Kategorie, welche alle Beziehungen zu
 - zugewiesenen Vertragsobjekten auflistet und hier gepflegt werden
 - können
- * Die nachfolgenden Kategorien sind ausschließlich für VMs definiert.
 - Viele davon sind sprechend und enthalten Informationen, die ihrer
 - Benennung entsprechen
- * Etwa Betriebssystem und Hostadresse
- * Darüber hinaus gibt es die Kategorie "Softwarezuweisung", welche
 - reziprok zur unter der Kategorie "Service" gelisteten "Service
 - Komponente" ist

- * Black Building Test beinhaltet Informationen zu einem vergangenen
 - Projekt, bei welchem alle Services in vorgesehener Reihenfolge ab und
 - wiederangeschaltet werden sollten. Es wird erfasst ob und wann die
 - Maschine abgeschaltet wurde
- * Die Kategorie "Buchhaltung" enthält allgemeine buchhalterische
 - Informationen, etwa Bestelldatum, Anschaffungskosten oder
 - Garantiezeiträume. Es wird nicht im Detail auf diese Kategorie
 - eingegangen, da sie für den weiteren Verlauf der Arbeit von
 - nachgelagertem Interesse ist
- * Unter "SLUB:Betriebsorganisation" wird der Status als Produktiv
 - oder-Testsystem gepflegt
- * "SLUBKonfiguration1" beinhaltet ausschließlich Zustandsinformationen zu
 - verschiedenen Attributen. Als Status vordefiniert ist jeweils "ok",
 - "to-do" und "nicht erforderlich". Vereinzelt gibt es Abwandlungen des
 - Status "ok", welche jedoch inhaltlich identisch sind. Die Attribute
 - dieser Kategorie werden nicht länger gepflegt und daher nicht im
 - Detail aufgelistet

Weitere Hilfssysteme

- * Die weiteren Systeme, welche im Kontext des ITSM, Monitoring und
 - Dokumentation der Services anwendung finden, enthalten keine für das
 - ITSM selbst erforderlichen Informationen
 - * Die Analyse der Systeme hat ergeben, dass das CMDB nahezu alle
 - Informationen im Kontext von ITSM, CM und Abhängigkeiten enthält
 - * Nachfolgend sind die Ausnahmen aufgelistet
 - * Einzige Ausnahme bildet aktuell das System GUBS, welches für die
 - Bereitstellung von VMs genutzt wird
 - * Hier werden nebst der initialen Systemkonfiguration auch Berechtigungen
 - gepflegt
 - * Dies erfolgt in Form frei definierbarer Gruppen, welchen Personen
 - zugeordnet werden
 - * Diese können anschließend die Systemkonfigurationen, welche über PXE
 - verteilt werden, für die zugewiesenen Server vornehmen
-

A.2 Ausgabe

Kapitel 2 - Theoretische Grundlagen

Moderne IT-Landschaften zeichnen sich durch eine hohe Komplexität aus,

- sowohl hinsichtlich der Vielzahl beteiligter Komponenten als auch
- ihrer technologischen Qualität. Ein erfolgreicher und stabiler
- Betrieb solcher Systeme setzt daher Transparenz, strukturierte
- Organisation, kontinuierliche Überwachung und gezielte Steuerung
- sämtlicher Elemente voraus. Zentrale Konzepte, welche zur
- Realisierung dieser Anforderungen beitragen, sind das IT Service
- Management (ITSM) sowie das Configuration Management (CM). In enger
- Verbindung hierzu stehen weitere Begriffe und Werkzeuge, wie etwa
- ITIL als Implementierungs-Framework für ITSM, die Configuration
- Management Database (CMDB), das Configuration Management System (CMS)
- und der Servicedesk. Das vorliegende Kapitel widmet sich der
- Definition, Einordnung und Darstellung dieser Begriffe sowie ihrer
- wechselseitigen Zusammenhänge. Darüber hinaus werden angrenzende
- Disziplinen wie das IT Asset Management (ITAM) sowie moderne
- Praktiken, insbesondere DevOps, betrachtet.

IT Service Management (ITSM)

ITSM beschreibt die Gesamtheit aller Methoden, Praktiken und Prozesse,

- die notwendig sind, um IT-gestützte Services zu planen, zu steuern,
- zu überwachen und letztlich bereitzustellen. Die im ITSM betrachteten
- Services sind immaterielle Leistungen, die darauf abzielen, einen
- geschäftlichen Mehrwert für interne oder externe Kunden zu schaffen.
- Zu den zentralen Teilgebieten des ITSM zählen das Incident Management,
- Problem Management, Change Management, Release Management sowie das
- Configuration Management.

Diese Teilgebiete stehen nicht isoliert nebeneinander, sondern

- profitieren wechselseitig voneinander. Ein besonders prägnantes
- Beispiel für die Synergien innerhalb des ITSM ergibt sich im
- Zusammenhang mit einem gepflegten Configuration Management: Im Fall
- eines Incidents ermöglicht eine detaillierte Dokumentation der
- Systemkonfiguration eine schnellere Lokalisierung des Fehlers und
- damit eine beschleunigte Fehlerdiagnose. Auch im Rahmen des Change
- Managements lässt sich durch das Configuration Management präzise
- analysieren, welche Systemkomponenten von einer geplanten Änderung
- betroffen wären, was zur Minimierung potenzieller Fehlerquellen
- beiträgt und eine fundierte Risikobewertung erlaubt.

Services im Detail

Ein Service lässt sich im Kontext des ITSM als eine Dienstleistung

- beschreiben, die einen spezifischen Wert für einen Kunden generieren
- soll. Dieser Wertbeitrag dient unmittelbar oder mittelbar der
- Zielerreichung innerhalb eines Geschäftsprozesses. Kunden können
- sowohl innerhalb der eigenen Organisation als auch außerhalb verortet
- sein.

Greift ein Nutzer über IT-Systeme auf einen Service zu, handelt es sich

- um einen IT-Service. Im weiteren Verlauf dieser Arbeit wird der
- Begriff „Service“ stellvertretend für IT-Services verwendet. Jeder
- Service wird durch einen sogenannten Service Provider bereitgestellt,
- während ein Service Consumer diesen Service nutzt, um daraus Nutzen
- zu ziehen. Dieser Nutzen kann entweder in der direkten Verwertung des
- bereitgestellten Services bestehen - beispielsweise beim Zugriff auf
- Internet oder Telefonie - oder in der Nutzung zur Bereitstellung
- eines weiteren Services, etwa bei der Verwendung eines
- Datenbank-Services zur Entwicklung einer Webapplikation.

ITIL

ITIL ist ein Framework zur Umsetzung von IT Service Management und hat

- sich als De-facto-Standard etabliert. Die aktuell gültige Version,
- ITIL 4 Edition, bezeichnet das Configuration Management als Service
- Configuration Management und betont damit explizit die
- Serviceorientierung dieses Teilbereichs.

Das Ziel von ITIL besteht in der Definition von Best Practices, welche es

- Organisationen ermöglichen sollen, ein standardkonformes ITSM -
- einschließlich eines konsistenten Configuration Managements -
- umzusetzen. Dabei definiert ITIL bewusst keine konkreten Werkzeuge
- oder Prozesse, sondern fungiert als übergeordneter Rahmen. Neben den
- allgemeinen ITSM-Zielen wie Planung, Steuerung, Überwachung und
- Betrieb von IT-Services hebt ITIL insbesondere die Unterstützung bei
- der Entscheidungsfindung, das Risikomanagement sowie die
- Effizienzsteigerung hervor.

Configuration Management

Das Configuration Management ist ein integraler Bestandteil des ITSM. Es

- befasst sich mit der systematischen Erfassung, Pflege und Überwachung
- aller Komponenten und Ressourcen, die zur Erbringung eines IT-Service
- erforderlich sind. Ziel ist eine vollständige und nachvollziehbare
- Darstellung der gesamten IT-Infrastruktur, wodurch Transparenz
- geschaffen, Fehlerquellen reduziert und eine effiziente
- Prozessgestaltung ermöglicht werden.

Die zugrunde liegende Infrastruktur setzt sich aus sogenannten

- Configuration Items (CIs) sowie deren Beziehungen untereinander
- zusammen. Diese Struktur lässt sich als Graph modellieren, wobei die
- CIs als Knoten und die zwischen ihnen bestehenden Beziehungen als
- Kanten fungieren.

Configuration Items (CIs) und deren Beziehungen

Configuration Items bilden die elementaren Bausteine des Configuration

- Managements. Dabei handelt es sich um physische oder logische
- Einheiten, Dokumente oder auch Services, die zur Erbringung eines
- bestimmten IT-Services notwendig sind. Typische Beispiele umfassen
- Server, Netzwerkinfrastrukturen, virtuelle Maschinen, Betriebssysteme,
- Anwendungen, Datenbanken, SLAs oder Webservices.

Jedes CI verfügt über mindestens einen eindeutigen Bezeichner sowie eine

- Vielzahl definierter Attribute wie etwa Status oder Version. Die CIs
- stehen in unterschiedlichen Beziehungstypen zueinander,
- beispielsweise „läuft auf“, „gehört zu“ oder „abhängig von“. Diese
- Beziehungen sind in der Regel gerichtet, können jedoch bidirektional
- interpretiert werden.

Das Modell zur Beschreibung der CIs folgt einem objektorientierten

- Paradigma: CI-Typen entsprechen dabei den Klassen aus der
- objektorientierten Programmierung, während konkrete CIs den Objekten
- zugeordnet werden. Die CI-Typen legen fest, welche Attribute für die
- jeweiligen CIs verfügbar sind.

Configuration Management Database (CMDB)

Die CMDB ist das zentrale Repository für sämtliche im Configuration

- Management erfassten Informationen. Sie dient sowohl der Speicherung
- als auch der Pflege (Erfassung, Änderung, Löschung) und Darstellung
- von CIs, deren Attributen und deren Beziehungen. Neben objekt- und
- dokumentenbasierten Ansichten bietet die CMDB häufig auch
- graphbasierte Visualisierungen mit weitreichenden Filter- und
- Anzeigoptionen.

Besonders im operativen IT-Betrieb ist eine aktuelle und konsistente

- Datenbasis in der CMDB von essenzieller Bedeutung. Angesichts der
- hohen Komplexität moderner IT-Infrastrukturen ist eine manuelle
- Pflege jedoch kaum praktikabel und mit erheblichen Risiken
- hinsichtlich Fehleranfälligkeit und Inkonsistenz verbunden. Daher
- kommen zunehmend automatisierte Mechanismen zum Einsatz, welche
- relevante Informationen identifizieren, verarbeiten und in die CMDB
- integrieren.

Configuration Management System (CMS)

Das CMS umfasst alle Werkzeuge, Schnittstellen und Datenbanken, die zur

- Erfassung, Pflege, Analyse und Darstellung von
- Konfigurationsinformationen notwendig sind. Es integriert die CMDB
- als zentralen Datenbankbestandteil und stellt darüber hinaus
- zusätzliche Funktionen bereit, etwa zur automatisierten
- Informationsgewinnung aus Monitoring- oder Ticket-Systemen.

Ziel ist es, sämtliche für den Betrieb einer IT-Landschaft relevanten

- Informationen und Aktivitäten zentral zu bündeln. Das CMS unterstützt
- die Umsetzung der im ITSM und CM formulierten Ziele, wie etwa
- Schwachstellen- und Risikoanalyse, Changemanagement sowie
- Ursachenanalyse bei Störungen. Ergänzt wird das CMS durch
- Präsentationsmöglichkeiten wie Dashboards und Berichtsfunktionen.

IT Asset Management (ITAM)

ITAM stellt ein weiteres Teilgebiet des ITSM dar und fokussiert sich auf

- die Verwaltung materieller und immaterieller IT-Ressourcen. Dazu
- zählen unter anderem Lizenzen, Hardware, Software, Wartungsverträge
- oder Garantien. Diese sogenannten IT-Assets durchlaufen innerhalb des
- ITAM ein vollständiges Lifecycle Management, das von der Beschaffung
- über den operativen Einsatz bis hin zur Ausmusterung reicht.

IT-Assets werden ebenfalls in der CMDB dokumentiert. Während das ITSM den

- Fokus auf die Service-Erbringung richtet, konzentriert sich das ITAM
- auf die Verwaltung der hierfür eingesetzten Ressourcen.

Bezug zu DevOps und agiler IT-Organisation

Im Zuge der fortschreitenden Automatisierung und Agilisierung von

- IT-Organisationen gewinnen Verfahren wie DevOps, Continuous Delivery
- und Continuous Integration (CI/CD) zunehmend an Bedeutung. Diese
- Praktiken setzen ein fundiertes und stets aktuelles Configuration
- Management voraus.

Ein zentrales Konzept ist hierbei „Infrastructure as Code“ (IaC). Hierbei

- werden Infrastrukturen, Abhängigkeiten und Komponenten auf Codebasis
- definiert und mittels Automatisierungstools wie Ansible, Puppet oder
- Terraform implementiert und orchestriert. Gleichzeitig führt die
- zunehmende Dynamik moderner IT-Landschaften zu einer potenziellen
- Informationslücke im Configuration Management: Neue Informationen
- entstehen in immer kürzeren Intervallen.

Damit automatisierte Prozesse effizient funktionieren, müssen aktuelle

- Konfigurationsdaten verfügbar sein. Umgekehrt können solche Prozesse
- Informationslücken verursachen, wenn sie nicht adäquat rückgekoppelt
- werden. Daraus ergibt sich die Notwendigkeit von Mechanismen, welche
- sowohl die kontinuierliche Aktualisierung der Konfigurationsdaten
- gewährleisten als auch geeignete Schnittstellen zur
- Datenbereitstellung bieten.

Zusammenfassung

Die Realisierung betrieblicher Geschäftsziele erfordert die

- Bereitstellung geeigneter Services. IT-gestützte Services sind
- hochkomplex und bedürfen einer umfassenden Verwaltung. Die hierfür
- etablierten Methoden und Praktiken sind im IT Service Management
- (ITSM) verortet, welches sich in diverse Teilgebiete gliedert.

Eine zentrale Rolle spielt hierbei das Configuration Management, dessen

- Informationen in einer Configuration Management Database (CMDB)
- gespeichert und gepflegt werden. Die CMDB fungiert als zentrales
- Repositorium und ermöglicht die notwendige Transparenz und Struktur
- zur effizienten Steuerung der IT-Landschaft.

Configuration Management steht in enger Beziehung zu sämtlichen

- Teilgebieten des ITSM und ist insbesondere für den operativen Betrieb
- von kritischer Bedeutung. Daher liegt ein besonderer Fokus dieser
- Arbeit auf der Ausgestaltung und Umsetzung eines funktionalen
- Configuration Managements in Verbindung mit einer CMDB.

Um die kontinuierliche Pflege der CMDB sicherzustellen, sind

- automatisierte Mechanismen unabdingbar. Darüber hinaus bedarf es
 - klarer Verantwortlichkeiten hinsichtlich Services, Teilservices und
 - zugehöriger Komponenten. Der Erfolg eines ITSM-Ansatzes hängt
 - maßgeblich von der Akzeptanz geeigneter Werkzeuge und der
 - unternehmensweiten Anerkennung der Bedeutung von Dokumentation und
 - Informationspflege ab.
-

Literaturverzeichnis

- [cmk] *Infrastruktur- und Applikations-Monitoring mit Checkmk*. <https://checkmk.com/de>
- [duda] *Dominium Rechtschreibung, Bedeutung, Definition, Herkunft Duden*. <https://www.duden.de/rechtschreibung/Dominium>
- [dudb] *Domäne Rechtschreibung, Bedeutung, Definition, Herkunft Duden*. <https://www.duden.de/rechtschreibung/Domaene>
- [IBM21a] *Configuration Management Database | IBM Documentation*. <https://www.ibm.com/docs/en/control-desk/7.6.0?topic=overview-configuration-management-database>. Version: März 2021
- [IBM21b] *Was ist die IT Infrastructure Library (ITIL)? | IBM*. <https://www.ibm.com/de-de/topics/it-infrastructure-library>. Version: September 2021
- [IBM21c] *Was ist IT-Service-Management (ITSM)? | IBM*. <https://www.ibm.com/de-de/topics/it-service-management>. Version: Oktober 2021
- [IBM23] *Configuration Items | IBM Dokumentation*. <https://www.ibm.com/docs/de/control-desk/7.6.1.2?topic=overview-configuration-items>. Version: Januar 2023
- [IBM24] *Was ist Konfigurationsmanagement? | IBM*. <https://www.ibm.com/de-de/topics/configuration-management>. Version: Dezember 2024
- [ido] *i-doit – IT-Dokumentation, CMDB, IT-Service Management*. <https://www.i-doit.com>
- [ITI19] *ITIL foundation : ITIL 4 edition*. Norwich : Stationery Office Books (TSO), 2019. – ISBN 978–0–11–331607–6. – ITIL4



Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit in allen Teilen selbstständig angefertigt und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt habe, und dass die Arbeit in gleicher oder ähnlicher Form in noch keiner anderen Prüfung vorgelegen hat. Mir ist bewusst, dass ich Autor/in der vorliegenden Arbeit bin und volle Verantwortung für den Text trage. ☒

Ich erkläre, dass ich wörtlich oder sinngemäß aus anderen Werken – dazu gehören auch Internetquellen – übernommene Inhalte als solche kenntlich gemacht und die entsprechenden Quellen angegeben habe. ☒

Mir ist bewusst, dass meine Arbeit auf Plagiate überprüft werden kann. Mir ist bekannt, dass es sich bei der Abgabe eines Plagiats um ein schweres akademisches Fehlverhalten handelt und dass Täuschungen nach der für mich gültigen Prüfungsordnung geahndet werden. ☒

Zusätzlich versichere ich, dass ich auf künstlicher Intelligenz (KI) basierende Werkzeuge nur in Absprache mit den Prüfern verwendet habe. Dabei stand meine eigene geistige Leistung im Vordergrund, und ich habe jederzeit den Prozess steuernd bearbeitet. ☒

Diese Werkzeuge habe ich im Quellenverzeichnis in der Rubrik „Übersicht verwendeter Hilfsmittel“ mit ihrem Produktnamen und einer Übersicht des im Rahmen dieser Prüfungs-/Studienarbeit genutzten Funktionsumfangs unter Angabe der Textstelle in der Arbeit vollständig aufgeführt. ☒

Ich versichere, dass ich keine KI-basierten Tools verwendet habe, deren Nutzung die Prüfer explizit schriftlich ausgeschlossen haben. Ich bin mir bewusst, dass die Verwendung von Texten oder anderen Inhalten und Produkten, die durch KI-basierte Tools generiert wurden, keine Garantie für deren Qualität darstellt. ☒

Ich verantworte die Übernahme jeglicher von mir verwendeter maschinell generierter Passagen vollumfänglich selbst und trage die Verantwortung für eventuell durch die KI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate. ☒

Dresden, 03.12.2025

Ort, Datum

Unterschrift