

BACHELORTHESES
Henry Peters

Entwicklung und Einführung eines Vorgehensmodells zur Optimierung von Business Intelligence Prozessen

Analyse einer veralteten Auswertung und Erneuerung als
interaktives Dashboard in einem medizinischen Unternehmen

FAKULTÄT DESIGN, MEDIEN UND INFORMATION
Department Medientechnik

Faculty of Design, Media and Information
Department of Media Technology

Henry Peters

Entwicklung und Einführung eines Vorgehensmodells zur Optimierung von Business Intelligence Prozessen

Analyse einer veralteten Auswertung und Erneuerung als interaktives Dashboard
in einem medizinischen Unternehmen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Media Systems*
am Department Medientechnik
der Fakultät Design, Medien und Information
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Larissa Putzar
Zweitgutachter: Prof. Dr. Andreas Pläß

Eingereicht am: 08. April 2022

Henry Peters

Thema der Arbeit

Entwicklung und Einführung eines Vorgehensmodells zur Optimierung von Business Intelligence Prozessen

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Entwicklung eines Vorgehensmodells, zur Optimierung und Ablösung von veralteten Business Intelligence Prozessen. Als Basis für die Konzeption des Vorgehensmodells dienen verschiedene Modelle aus dem Data-Mining und der Software- und BI-Entwicklung, da sowohl Datenaufbereitungs- als auch iterative Entwicklungsschritte als wichtig angesehen werden. Nach der konzeptionellen Ausarbeitung wird das Modell in einem medizinischen Unternehmen eingesetzt, um eine veraltete wöchentliche Auswertung zu durchgeführten Behandlungen mit einem interaktiven und umfangreicheren Dashboard zu ersetzen.

Henry Peters

Title of Thesis

Development and Implementation of a Process Model for the Optimization of Business Intelligence Processes

Abstract

This thesis deals with the development of a process model for the optimization and replacement of outdated business intelligence processes. Various models from data mining as well as software and BI development serve as the basis for the conceptual design of the process model, since both data preparation and iterative development steps are considered important. After the conceptual elaboration, the model is used in a medical company to replace an outdated weekly report on performed treatments with an interactive and more comprehensive dashboard.

Inhaltsverzeichnis

Abbildungsverzeichnis	VI
Auflistungen	VII
Abkürzungen	VIII
1 Einleitung	1
2 Business Intelligence Grundlagen	3
2.1 Business Intelligence	3
2.2 BI Reifegradmodell	6
3 Methodisches Vorgehen	8
4 Konzeption des Vorgehensmodells	9
4.1 Anforderungen	9
4.2 Bestehende Vorgehensmodelle	11
4.2.1 Knowledge Discovery in Databases (KDD)	12
4.2.2 Cross Industry Standard Process for Data Mining (CRISP-DM)	14
4.2.3 Iterativ evolutionäres Vorgehensmodell	18
4.2.4 Phasenorientiertes BI-Vorgehensmodell	20
4.3 Entwicklung des Vorgehensmodells	22
4.3.1 Vorbereitung	23
4.3.2 Datenaufbereitung	26
4.3.3 Entwicklung	27
4.3.4 Zusammenfassender Aufbau des Vorgehensmodells	29
4.3.5 Grafische Umsetzung	30
4.4 Zwischenfazit	32

5	Praktische Umsetzung	34
5.1	Verständnis der Domäne	34
5.2	Anforderungsanalyse	36
5.2.1	Funktionale Anforderungen	38
5.2.2	Nichtfunktionale Anforderungen	38
5.3	Analyse und Dokumentation der bisherigen Auswertung	39
5.4	Wahl einer Auswertungssoftware	42
5.5	Vorbereiten der Datenquellen	43
5.6	Datenaufbereitung	45
5.6.1	Auswahl	46
5.6.2	Bereinigung und Konstruktion	47
5.6.3	Integration und Formatierung	53
5.7	Entwicklung der Auswertung	55
5.7.1	Laden der Daten im Dateneditor	55
5.7.2	Layout des Dashboards	57
5.7.3	Implementierung der Filter und Auswahlmöglichkeiten	59
5.7.4	Implementierung der Visualisierungen	61
5.8	Testen der Auswertung	69
5.8.1	Funktionale Anforderungen	69
5.8.2	Nichtfunktionale Anforderungen	70
5.9	Abschluss und interne Veröffentlichung	71
6	Fazit und Ausblick	72
	Literatur	75
A	Auszug aus einem Wochenbericht	77
B	Auszug aus den wöchentlichen Behandlungszahlen	79
C	Erklärung der Datenbankattribute	80
D	Finales Dashboard	81
	Glossar	84
	Selbstständigkeitserklärung	85

Abbildungsverzeichnis

2.1	Funktionen eines Business Intelligence Competency Center	5
2.2	BI and Analytics Maturity Model	6
4.1	Phasen des KDD Prozesses	12
4.2	CRISP-DM Vorgehensmodell	15
4.3	Evolutionäres Vorgehensmodell	19
4.4	Phasenorientiertes BI-Vorgehensmodell	20
4.5	Grundlegende auf CRISP-DM basierte Phasen	23
4.6	Erster Entwurf des MOROP-BI Vorgehensmodells	30
4.7	MOROP-BI Vorgehensmodell	31
5.1	Auszug aus dem veralteten Wochenbericht - durchschnittliche Behand- lungszahlen und Trends	40
5.2	Auszug aus dem veralteten Wochenbericht - Vorjahresvergleich	41
5.3	Auszug aus der <i>TreatmentCTE</i> Abfrage	47
5.4	Auszug aus der <i>TreatmentBillingItem</i> Tabelle	48
5.5	Auszug aus der <i>TreatmentClass</i> Tabelle	50
5.6	Auszug aus der finalen Abfrage	54
5.7	Layoutsentwurf des Dashboards	58
5.8	KPIs im oberen Bereich des Dashboards	62
5.9	Grafische Entwicklung der Patienten- und Behandlungszahlen	65
5.10	Nachstellung des ehemaligen Wochenberichts	66
5.11	Patientenentwicklung als Wasserfalldiagramm	68

Auflistungen

5.1	Basis Abfrage der Behandlungen (<i>TreatmentCTE</i>)	46
5.2	kombinierte Abfrage des Behandlungsortes nach Region, Unternehmen und Zentrum (<i>RegionLegalEntityLocationCTE</i>)	49
5.3	Abfrage der Behandlungsklassen nach <i>TreatmentClass</i> Tabelle (Teil der <i>TreatmentClassCTE</i>)	51
5.4	Abfrage der Behandlungsklassen nach <i>Treatments</i> Tabelle (Teil der <i>Treat-</i> <i>mentClassCTE</i>)	52
5.5	kombinierte Abfrage der Behandlungsklassen (<i>TreatmentClassCTE</i>)	53
5.6	Finale Abfrage der Behandlungen	54
5.7	Ladeskript im Dateneditor von Qlik Sense	56
5.8	Schalterlogiken zum Umschalten der Variablen	60
5.9	Berechnungen der Patienten- und Behandlungszahlen	62
5.10	Berechnungen der KPIs	63
5.11	Berechnungen der Patientenzahlen - Aktuelles Jahr	64
5.12	Berechnungen der Patientenzahlen - Vorjahr	65
5.13	Berechnungen des 4 vs. 12 Wochen Trends der Patienten	67
5.14	Berechnungen der Patienten der Vorjahreswoche	67

Abkürzungen

AN Apherese

APD automatisierte Peritonealdialyse

BI Business Intelligence

BICC Business Intelligence Competency Center

BSNR Betriebsstättennummer

CAPD kontinuierliche ambulante Peritonealdialyse

CRISP-DM Cross Industry Standard Process for Data Mining

CTE Common Table Expression

EBM einheitlicher Bewertungsmaßstab

GFR glomeruläre Filtrationsrate

HD Hämodialyse

IPD intermittierende Peritonealdialyse

KBV Kassenärztliche Bundesvereinigung

KDD Knowledge Discovery in Databases

KPI Key Performance Indicator

LANR lebenslange Arztnummer

MOROP-BI Model for Optimizing and Replacing Outdated Processes for Business Intelligence

PD Peritonealdialyse

QMC Qlik Management Console

SSMS SQL Server Management Studio

1 Einleitung

Aufgrund der immer weiter anwachsenden Datenmengen innerhalb eines Unternehmens entsteht zunehmend der Wunsch nach datengetriebenen Geschäftsentscheidungen basierend auf Business Intelligence (BI). Obwohl es mittlerweile viele Möglichkeiten und Anwendungsbeispiele gibt, um aus ungeordneten und noch nicht vollständig erschlossenen Daten, zum Beispiel mittels Data-Mining, Mehrwert und Verständnis zu schaffen, werden alte und dadurch oftmals unternehmenssteuernde Auswertungen weiterhin genutzt und nur verändert, um die Kompatibilität fortlaufend zu gewährleisten.

Solche veralteten Berichte resultieren unter anderem aus inkonsistenten Datenquellen und nicht standardisierten und anfangs oft temporären Auswertungen. Diese Probleme lassen sich auf viele Unternehmen übertragen, die sich in den Anfangsphasen von Business Intelligence befinden und somit noch zu wenig Wert auf strukturierte Prozesse legen.

In der Softwareentwicklung ist eine vergleichbare Art von veralteter, aber dennoch weiter zu unterstützender Software als Legacy-System bekannt. Da das vollständige Ersetzen von Legacy-Systemen einen ressourcen- und zeitintensiven Prozess darstellt, wird er oft vernachlässigt, bis das System aufgrund von generellen Defiziten zusammenzubrechen droht. Vorgehensmodelle stellen hier wichtige Werkzeuge dar, um Prozesse von Beginn an optimal zu planen und durchzuführen (vgl. Nascimento u. a. 2009).

Um die Zukunftssicherheit und Modernität eines Unternehmens in Bezug auf Business Intelligence zu gewährleisten, ist es wichtig, veraltete Auswertungen frühzeitig zu erkennen und mithilfe eines standardisierten Prozesses zu ersetzen.

Obwohl es bereits einige Vorgehensmodelle für die Entwicklung und Implementierung von neuen BI-Lösungen in Unternehmen gibt, ist keines speziell darauf ausgelegt, aus den Erkenntnissen der bestehenden, jedoch veralteten Auswertungen zu lernen (vgl. Gluchowski, Gabriel und Dittmar 2008, S. 254 ff.).

Im Rahmen dieser Arbeit soll diese Forschungslücke geschlossen und ein Vorgehensmodell entwickelt werden, das eine strukturierte, standardisierte und dokumentierte Ablösung von veralteten BI Prozessen ermöglicht.

Um den Erfolg des Modells nach der Entwicklung aufzuzeigen, wird es in einem medizinischen Unternehmen eingesetzt, um einen veralteten wöchentlichen Bericht zu durchgeführten Behandlungen durch ein interaktives und inhaltlich weitaus umfangreicheres Dashboard zu ersetzen.

Angesichts dessen werden im ersten Teil der Arbeit die Grundlagen von BI erklärt, sowie die Hintergründe von veralteten BI Prozessen erläutert. Anschließend wird mit der Konzeption des Vorgehensmodells begonnen. Dafür werden andere, bereits bestehende Vorgehensmodelle analysiert, die nachfolgend als Basis für die Ausarbeitung des Modells dienen. Nach der konzeptionellen Entwicklung wird das Modell im Unternehmen schrittweise angewendet und die Ergebnisse zusammengefasst. Abschließend wird der Erfolg des Modells auch in Hinblick auf Verbesserungen und weitere Anwendungsmöglichkeiten in einem kritischen Fazit beurteilt.

2 Business Intelligence Grundlagen

Um einleitend einen Einblick in das Feld der Business Intelligence zu geben, wird kurz auf die historische Entwicklung eingegangen und verschiedene Definitionen von BI vorgestellt. Im Anschluss wird eine für diese Arbeit relevante Definition von Business Intelligence aufgestellt und BI aus heutiger Sicht im Unternehmen eingeordnet. Abschließend folgt die Vorstellung eines Reifegradmodells, welches unterschiedliche Entwicklungsstufen von BI in einem Unternehmen aufzeigen kann und dabei hilft veraltete BI Prozesse zu verstehen.

2.1 Business Intelligence

Der Begriff „Business Intelligence“ wurde erstmals 1958 vom deutschen Informatiker Hans Peter Luhn im Artikel „A Business Intelligence System“ des „IBM Journal of Research and Development“ erwähnt. Er beschrieb ein theoretisches System, das es einem Unternehmen ermöglichen sollte, sowohl interne als auch externe, unstrukturierte Daten automatisch zu analysieren und die Analyseergebnisse zusammen mit einer Kopie des Originaldokuments abzuspeichern. Die im Dokument enthaltenen Informationen sollten anschließend in einer organisierten Art und Weise den Mitarbeitern zugänglich gemacht werden, damit auf Basis der Analyseergebnisse schnellere und genauere Entscheidungen getroffen werden können (vgl. Luhn 1958).

Aus dieser ersten Idee eines Systems zur Analyse und Auswertung von Daten entwickelte sich ein für heutige Unternehmen essenzielles Werkzeug in der Unternehmensführung. Bis heute existiert jedoch noch keine allgemeingültige Definition für Business Intelligence. Die Fachgruppe „Business Intelligence und Analytics“ der Gesellschaft für Informatik fasst die häufigsten Definitionen in folgende Punkte zusammen (Baars u. a. 2011):

- BI als Sammlung von IT-Systemen
- BI als übergeordnete IT-Architektur
- BI als Sammelbegriff für diverse Technologien und Konzepte zur Entscheidungsunterstützung
- BI als IT-basierter Gesamtansatz zur Entscheidungsunterstützung
- BI als Wissensgenerierungsprozess

Trotz der inhaltlichen Unterschiede der Definitionen, hat sich laut Baars u. a. (2011) mittlerweile sowohl in der Literatur als auch in den Ansätzen der führenden BI-Systemanbieter ein grundsätzliches Verständnis von BI ergeben. Eine solche allgemeingültige Definition liefern Gluchowski, Gabriel und Dittmar (vgl. 2008, S. 93). Sie beschreiben BI als eine Verschmelzung verschiedener entscheidungsunterstützender Technologien, die eine Sammlung, Aufbereitung und Präsentation von geschäftsrelevanten Informationen zum Zwecke einer besseren Analyse, Planung und Kontrolle ermöglichen.

Daraus resultierend wird Business Intelligence für den Zweck dieser Arbeit wie folgt definiert:

Business Intelligence dient der Analyse und Präsentation von geschäftsrelevanten Informationen und lässt sich in die Bereitstellung und Aufbereitung von Daten, sowie die Wissensgewinnung und abschließende Darstellung dieser unterteilen.

Die Ausführlichkeit der einzelnen Punkte ist dabei abhängig von der spezifischen Anwendung und den damit verbundenen Anforderungen an die Ergebnisse des Business Intelligence Prozesses.

Dabei wird BI heutzutage sowohl von Führungskräften als auch vom entsprechenden Fachpersonal benötigt, um Planungs- und Kontrollprozesse zu unterstützen (vgl. Gluchowski, Gabriel und Dittmar 2008, S. 15). Die Einbindung ins Unternehmen gliedert sich dabei nach Totok (vgl. 2016, S. 45) oftmals noch in zwei getrennte Bereiche. Die BI-Infrastruktur, welche auf IT-Ebene entwickelt, verwaltet und zur Verfügung gestellt wird und darauf aufbauend die jeweiligen Fachbereiche des Unternehmens, welche durch die Auswertungen Erkenntnisse gewinnen oder Entscheidungen treffen. Typische Unterkategorien der IT-seitigen BI können dabei die Datenbankadministration oder die Entwicklung und Architektur der Auswertungen sein, wobei das Controlling, der Vertrieb und das Marketing zu den Benutzergruppen zählen können.

Totok (vgl. 2016, S. 45 f.) führt weiter aus, dass die Zusammenlegung der beiden Bereiche BI-Projekte erfolgreicher und flexibler macht, da alle Kompetenzen gebündelt werden und die Kommunikation zwischen den einzelnen Rollen vereinfacht wird. Da eine solche Integration oftmals mit dem organisatorischen Aufbau eines Unternehmens kollidiert, kann die Gründung eines Business Intelligence Competency Center (BICC) sinnvoll sein. Das BICC kann als zentraler Ansprechpartner sowohl für die Fachbereiche als auch für die IT des Unternehmens dienen und berät beide Seiten konzeptionell (vgl. Totok 2016, S. 45 f.). Typische Funktionen eines BICC werden in Abbildung 2.1 gezeigt.

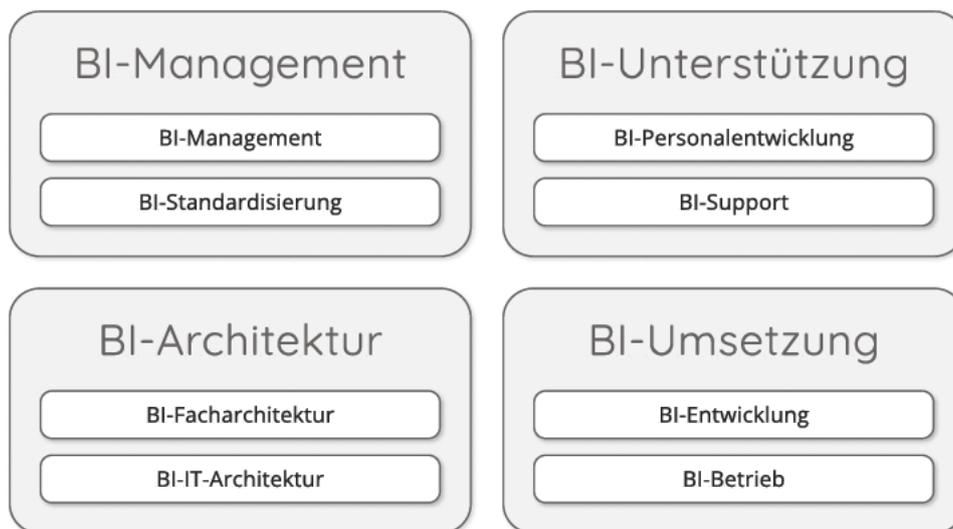


Abbildung 2.1: Funktionen eines Business Intelligence Competency Center (vgl. Totok 2016, S. 46)

2.2 BI Reifegradmodell

Um ein tieferes Verständnis für die Entstehung von veralteten BI Prozessen zu erlangen, kann das Reifegradmodell des IT-Beratungs- und Marktforschungsunternehmens Gartner herangezogen werden. Ein Reifegradmodell ordnet im Generellen einen bestimmten Entwicklungsstatus auf einer für das Fach spezifischen Skala ein und gibt so Auskunft darüber, welche zukünftigen Entwicklungsstufen existieren und wie diese erreicht werden können. Das hier angesprochene Reifegradmodell bezieht sich im Speziellen auf die Entwicklung der BI-Infrastruktur innerhalb eines Unternehmens.

Das „BI and Analytics Maturity Model“ der Gartner Analysten Howson und Duncan (2015) teilt den Business Intelligence Prozess in fünf Entwicklungsstufen ein: *Unaware*, *Opportunistic*, *Standards*, *Enterprise* und *Transformative*, zu sehen in Abbildung 2.2.

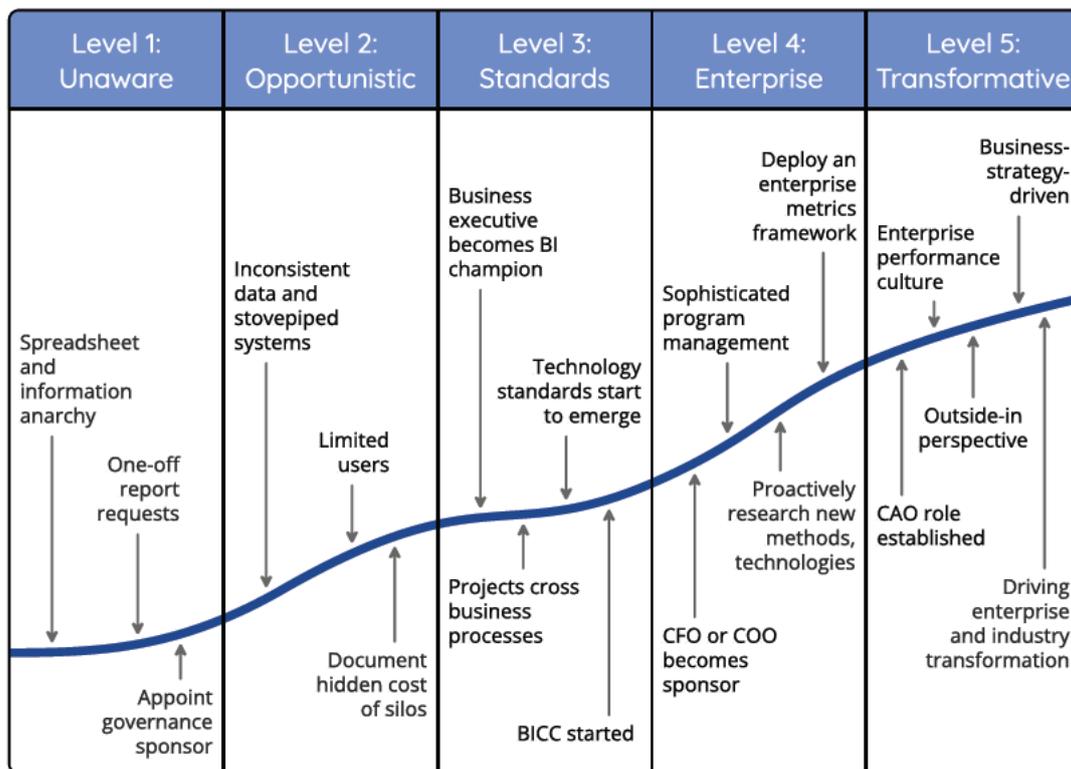


Abbildung 2.2: BI and Analytics Maturity Model (Howson und Duncan 2015, vgl.)

Im ersten Level herrsche laut Gartner eine *Informationsanarchie*, es gäbe keine formellen Entscheidungsprozesse und Berichte würden aus unterschiedlichsten Systemen und Datenquellen zusammengetragen und dann inkonsistent ausgewertet werden.

Unternehmen auf der zweiten Ebene fangen an, Auswertungen auf Abteilungsebene zu erstellen und zu nutzen. Zu diesem Zeitpunkt existieren aber noch keine übergreifenden Metriken und die Auswertungen werden mithilfe von unangepasster Standardsoftware erstellt.

Der dritte Schritt des Modells sieht eine anfangende Standardisierung sowohl der Prozesse als auch der unterliegenden Daten vor. Auch etabliert sich BI zunehmend in den Geschäftsprozessen, wobei oftmals ein leitender Angestellter zum Unternehmensbeauftragten für BI und Analysen wird. Zusätzlich dazu kann es vorkommen, dass ein BICC gegründet wird, um zentrale Ansprechpartner für BI Projekte zu etablieren.

Im vierten Schritt ist die BI Infrastruktur so weit ausgebaut, dass die Auswertungen die meisten Entscheidungsprozesse im Unternehmen maßgeblich unterstützen. Dies geschieht durch die Definition von unternehmensweiten Key Performance Indicators (KPIs) und der gemeinsamen Nutzung von Datensätzen und BI- und Analysesystemen.

Das letzte zu bewältigende Level des Modells sieht eine voll umfassende Nutzung von Business Intelligence im Unternehmen vor. Es werden bei allen anfallenden Prozessen standardisierte Auswertungen und Modelle verwendet, die durch umfassende und vollständige KPIs gestützt werden. Der gesamte BI Geschäftsprozess ist hierbei darauf ausgerichtet, effizient und fehlerfrei zu arbeiten und somit eine durchgehende Wertsteigerung des Unternehmens zu gewährleisten.

Die Analysten Howson und Duncan beschreiben weiter, dass ihr Modell von Unternehmen genutzt werden kann, um die Reife ihrer BI Infrastruktur nachhaltig zu verbessern. Gerade die Identifikation von Engpässen, also einzelnen Abteilungen, die sich im Vergleich zum Rest des Unternehmens auf einem niedrigeren Level befinden, ist dafür essenziell.

Das Problem der Entstehung von veralteten BI Prozessen kann in diesem Modell in die ersten beiden Phasen eingeordnet werden. Da noch keine klaren Strukturen für die Entwicklung und Instandhaltung von Auswertungen existieren, ergeben sich auf diesen Stufen inkonsistente und schnell veraltende Prozesse. Ein Beispiel hierfür wäre ein nicht dokumentiertes und durch mehrere Ordnerumstrukturierungen nun fehlerhaftes Spreadsheet, welches jedoch eine Berechnung für eine spezielle und wichtige Kennzahl enthält.

3 Methodisches Vorgehen

Das zu entwickelnde Vorgehensmodell wird auf der Grundlage bereits vorhandener Modelle und zuvor definierter Anforderungen konzipiert. Um es anschließend praxisnah testen und bewerten zu können, folgt nach der Konzeption und Ausarbeitung eine spezifische Anwendung in einem Unternehmen.

Im Detail soll ein automatisiertes und interaktives Dashboard entwickelt werden, welches wöchentliche Patienten- und Behandlungszahlen darstellt und sobald eingesetzt den bisherigen mittlerweile veralteten Report ablöst.

Das dafür gewählte Unternehmen ist in der Nephrologie tätig und betreibt mehrere Dialysezentren in Deutschland. In diesen Zentren werden, meist dreimal pro Woche, Patienten behandelt, die an einem chronischen Nierenversagen leiden. Da die Nieren unter anderem für die Filtration des Blutes zuständig sind, reichern sich bei einem Versagen Schadstoffe im Körper des Patienten an. Um dies zu verhindern, kann das Blut des Patienten aus dem Körper geleitet werden, um es in einer Dialysemaschine zu filtern und dem Patienten gereinigt wieder zuzuführen.

Die Anwender der neuen Auswertung werden hauptsächlich die leitenden Ärzte der jeweiligen Zentren, sowie das regional übergeordnete Management bestehend aus dem Controlling und der Geschäftsführung sein, die das Dashboard als Kontrolle der wichtigsten Kennzahlen nutzen werden. Vorbereitend auf die Entwicklung der Auswertung fanden mehrere Gespräche mit den Anwendern statt, um allgemeine Rahmenbedingungen festzulegen und erste Anforderungen zu definieren. Die technische Entwicklung hingegen wird ohne weitere Unterstützung durchgeführt, wobei einige der Erkenntnisse und Praktiken auf der Vorarbeit des restlichen internen BI Teams aufbauen. Der organisatorische Aufbau der BI im Unternehmen gliedert sich dabei in die in Kapitel 2 bereits angesprochene BI-Infrastruktur und die anwendenden Fachbereiche, sodass die typischen Rollen eines BICCs nicht zum Einsatz kommen.

4 Konzeption des Vorgehensmodells

Um ein ganzheitliches Vorgehensmodell zu erstellen, müssen viele unterschiedliche Aspekte des zu bewältigenden Prozesses analysiert und abstrahiert werden. In diesem Kapitel werden zuerst einige Anforderungen an das zu entwickelnde Modell erarbeitet und zusammengefasst, um dann verschiedene bestehende Vorgehensmodelle und Praktiken vorzustellen und zu vergleichen, die in die Entwicklung einfließen sollen. Aus den Anforderungen und den bereits bestehenden Prinzipien wird dann abschließend das finale Modell entwickelt, mit dem es möglich sein soll, bestehende BI Prozesse zu analysieren, optimieren und abzulösen.

4.1 Anforderungen

Das zu entwickelnde Vorgehensmodell sollte einige vorher definierte Anforderungen erfüllen, um einen erfolgreichen und nachhaltigen Ablauf des BI Prozesses zu ermöglichen. Die Anforderungen wurden konzeptionell aus Erfahrungswerten, unternehmensinternen Gesprächen und verschiedener Fachliteratur erstellt und werden im weiteren Verlauf, mit Referenz zu bestehenden Vorgehensmodellen weiter belegt.

Zunächst soll eine klar strukturierte und verständliche grafische Übersicht entwickelt werden, die auf visueller Ebene überzeugt und inhaltlich als Leitfaden für den Ablauf des Modells genutzt werden kann. Zusätzlich sollte eine Abstraktionsebene gewählt werden, die dem Modell eine gewisse Allgemeingültigkeit zuspricht, damit auch ähnliche Prozesse an das Modell angepasst und dann erfolgreich bearbeitet werden können. Hierbei sollte aber beachtet werden, dass die Formulierungen nicht zu viel Interpretationsspielraum bieten, um keine Zweideutigkeit oder Unverständnis hervorzurufen und eine klare Umsetzung der einzelnen Schritte zu ermöglichen. Eine klare Ein- und Unterteilung der einzelnen Teilschritte des Modells ist dabei ausschlaggebend. Auch soll das Modell iterative Prozesse unterstützen und nutzen, um starre Kausalketten zu vermeiden und eine

flexible und individuelle Entwicklung zu fördern. Dabei ist jedoch zu beachten, dass klar definierte Ziele formuliert werden, um effizient und ergebnisorientiert zu arbeiten.

Aus den genannten Punkten lassen sich die folgenden nichtfunktionalen Anforderungen zusammenfassen:

1. Klar strukturierte und verständliche grafische Übersicht
2. Anpassbarkeit an ähnliche Prozesse
3. Klare Ein- und Unterteilungen der Teilschritte
4. iterative Prozesse
5. klare Ziele

Zu den funktionalen Anforderungen gehört unter anderem, dass Langlebigkeit und Beständigkeit der Auswertung im Fokus der Entwicklung liegen. Da das Modell eine Ablösung von alten Prozessen vorsieht, sollte besonderer Wert darauf gelegt werden, dass die finale, den alten Prozess ablösende, Auswertung in naher Zukunft nicht erneut das Vorgehensmodell durchlaufen muss. Andererseits sollte jeder Entwicklungsprozess die spätere Erweiterbarkeit und Anpassbarkeit der resultierenden Auswertung berücksichtigen, da BI Auswertungen, ähnlich wie andere Softwareentwicklungen, nie als vollständig abgeschlossen angesehen werden können (vgl. Kemper, Mehanna und Unger 2006, S. 147 ff.). Um die Dauerhaftigkeit weiter zu gewährleisten, sollte die neue Auswertung außerdem robust und zukunftsorientiert sein. Die Stabilität ist wichtig, da andernfalls das Vertrauen in den neuen Prozess schwinden könnte und die Auswertung möglicherweise nicht mehr genutzt wird.

Ein weiterer wichtiger Punkt ist die Datenqualität. Sowohl Bange (vgl. 2016, S. 99 f.) als auch Gluchowski, Gabriel und Dittmar (vgl. 2008, S. 257) schreiben, dass eine unzureichende und mangelnde Datenqualität ein großes Problem in Informationssystemen darstellen. Bange (vgl. 2016, S. 99 f.) führt weiter aus, dass ein separater Datenaufbereitungsschritt zu den wesentlichen Aufgabenbereichen eines BI-Systems gehört. Um den Erfolg des Projektes weiter zu sichern und das Vertrauensverhältnis zwischen Anwender und Anwendung zu stärken, ist es deshalb entscheidend, dass die verwendeten Daten frei von Qualitätsmängeln wie Fehlern oder Lücken sind. Sollte die zugrunde liegende Qualität der Rohdaten nicht für fehlerfreie und stabile Auswertungen reichen, so soll ein Prozess gefunden werden, der diese Daten bereinigt oder neue Datenquellen erschließt und konsolidiert.

Ein abschließender und zentraler Bestandteil in der gesamten Entwicklung des Vorgehensmodells sollte der Fokus auf die Abwesenheit von logischen Fehlern oder Trugschlüssen in der finalen Auswertung sein, um nachhaltig das Vertrauen und den Mehrwert der Analyse aufrechtzuerhalten.

Die funktionalen Anforderungen werden daher wie folgt definiert:

1. Langlebigkeit
2. Stabilität
3. Qualität der verwendeten Daten
4. fehlerfreie Analysen

4.2 Bestehende Vorgehensmodelle

Um die aufgestellten Anforderungen zu erfüllen, werden im folgenden Abschnitt verschiedene Vorgehensmodelle vorgestellt und verglichen, die einige der angesprochenen Anforderungen behandeln und somit in die Entwicklung des Modells einfließen sollten. Aufgrund der Relevanz der Datenqualität werden neben allgemeinen prozessorientierten Vorgehensmodellen auch Modelle betrachtet, bei denen die Datenqualität ein entscheidendes Kriterium darstellt.

Demzufolge wird zuerst auf den „Knowledge Discovery in Databases“ (KDD) Prozess aus dem Data-Mining Bereich eingegangen, um dann das Vorgehensmodell „Cross Industry Standard Process for Data Mining“ (CRISP-DM) vorzustellen, das auf diesen Prozess aufbaut.

Im Anschluss wird aufgrund des geforderten iterativen Prozessablaufs des finalen Modells sowohl ein evolutionäres Vorgehensmodell als auch ein phasenorientiertes Vorgehensmodell vorgestellt. Ersteres stammt vorwiegend aus der Softwareentwicklung und wird als Beispiel für rein iterative Prozesse herangezogen. Das phasenbasierte Vorgehensmodell wurde für die Entwicklung von Business Intelligence Lösungen entworfen und soll ebenfalls als Anhaltspunkt für die noch folgende Konzeption des Vorgehensmodells dienen.

Es wurde bewusst auf die Vorstellung von sequenziellen Vorgehensmodellen verzichtet, da schon in den Anforderungen überlegt wurde, dass iterative Modelle für den vorgestellten Anwendungsfall besser geeignet sind.

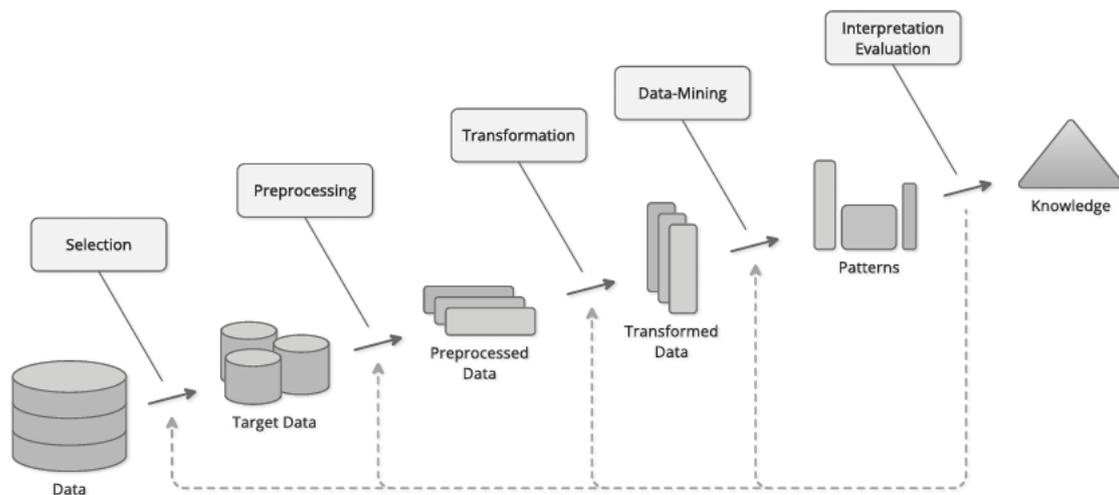


Abbildung 4.1: Phasen des KDD Prozesses (vgl. Fayyad, Piatetsky-Shapiro und Smyth 1996a, S. 41)

4.2.1 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases beschreibt sinngemäß den Prozess des Findens oder Erkennens von Wissen aus Datensätzen. Dieser Wissensfindungs- bzw. Wissensgenerierungsprozess wird mithilfe von gezielter Vorbereitung und anschließender automatischer Verarbeitung – dem Data-Mining – der Daten erreicht. Data-Mining wird dabei oft als Synonym für KDD benutzt, obwohl es nur einen Teilschritt im gesamten KDD Prozess ausmacht (vgl. Fayyad, Piatetsky-Shapiro und Smyth 1996a, S. 39). Fayyad, Piatetsky-Shapiro und Smyth, die an der Prägung und Definition dieses Begriffes maßgeblich beteiligt waren, beschreiben den KDD Prozess wie folgt:

“Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.” (Fayyad, Piatetsky-Shapiro und Smyth 1996b)

Um diesen nicht trivialen Prozess der Mustererkennung erfolgreich durchführen zu können, wurde ein idealer Prozessablauf entworfen, welcher in Abbildung 4.1 gezeigt und nachfolgend kurz erläutert wird (vgl. Fayyad, Piatetsky-Shapiro und Smyth 1996a, S. 42).

In der ersten Phase des KDD Prozesses müsse demnach ein Grundverständnis der Domäne erreicht und das finale Ziel aus Sicht des Kunden erfasst werden, um im zweiten Schritt einen für die angestrebte Wissensfindung geeigneten Zieldatensatz auszuwählen.

Anschließend müssen die Daten vorbereitet und bereinigt werden, dazu kann unter anderem das Entfernen oder Ersetzen von fehlerhaften Daten gehören. Der vierte Schritt besteht aus der Reduktion und Transformation der nun bereinigten Daten, um diese für die nachfolgenden Schritte weiter vorzubereiten und an das Ziel der Wissensfindung anzupassen. In den Schritten fünf und sechs wird schließlich eine zielführende Data-Mining-Methode (wie z. B. Regression oder Clustering) ausgewählt, um dann die Konfigurationen und allgemeinen Rahmenbedingungen für diese Methode festzulegen. Der siebte Schritt besteht aus der Anwendung der ausgewählten Methode. Hier wird nach neuen, besonderen Mustern in den Daten gesucht, die in Schritt acht interpretiert, bewertet oder auch zum besseren Verständnis visualisiert werden. Es wird außerdem erwähnt, dass gerade die Erkenntnisse dieses Schrittes einen Rücksprung auf eine der vorherigen Phasen bewirken. Im letzten Schritt des KDD Prozesses wird das neu gewonnene Wissen eingesetzt und dokumentiert. Der Prozess ist an diesem Punkt abgeschlossen, kann aber Ausgangspunkt für weitere Wissensgenerierungsprozesse sein.

Die Autoren beschreiben weiter, dass ein wesentlicher Teil des KDD Prozesses, wie auch im achten Schritt oder in Abbildung 4.1 angedeutet, der iterative Charakter des gesamten Vorgehens ist. Hervorgerufen durch die neuen Erkenntnisse der einzelnen Schritte ist es so jederzeit möglich und teilweise auch nötig zu bereits durchlaufenen Phasen zurückzuspringen. Dass iterative Prozesse für den erfolgreichen Ablauf eines Projektes von Vorteil sind, wurde schon in den Anforderungen an das zu entwickelnde Vorgehensmodell im vorherigen Abschnitt 4.1 angesprochen.

Abschließend schreiben die Autoren, dass für den erfolgreichen Ablauf des KDD Prozesses die vorbereitenden Phasen (Datenbereinigung, -reduktion, -transformation) oftmals sogar wichtiger als das eigentliche Data-Mining sind (vgl. Fayyad, Piatetsky-Shapiro und Smyth 1996a, S. 42). Ein ähnlicher Punkt findet sich bereits in den schon definierten Anforderungen im vorherigen Abschnitt, welcher die Notwendigkeit von fehlerfreien und qualitativ hochwertigen Daten hervorhebt (Punkt 3 der funktionalen Anforderungen).

4.2.2 Cross Industry Standard Process for Data Mining (CRISP-DM)

Der Cross Industry Standard Process for Data Mining ist ein Vorgehensmodell, welches von fünf Unternehmen¹ im Zeitraum von 1996 bis 2000 im Rahmen des CRISP-DM Konsortiums entwickelt wurde. Durch das gemeinsame Interesse und erste Erfahrungen im Data-Mining Bereich wollten die Mitglieder des Konsortiums ein universelles, frei zugängliches und nicht proprietäres Modell schaffen, um Data-Mining Projekte standardisiert und erfolgreich durchführen zu können (vgl. Chapman u. a. 2000, S. 1 f.). Das Vorhaben wurde zudem von der Europäischen Union im Rahmen der ESPRIT-Finanzierungsinitiative über einen Zeitraum von eineinhalb Jahren gefördert (vgl. European Union 1998). Im August 2000 wurde schließlich die finale Version des CRISP-DM Benutzerhandbuchs unter dem Titel „CRISP-DM 1.0: Step-by-step data mining guide“ veröffentlicht, welches das Vorgehensmodell, sowie ausführliche Erklärungen zu den einzelnen enthaltenen Schritten aufführt (vgl. Chapman u. a. 2000).

CRISP-DM wird in sechs Phasen unterteilt, welche in Abbildung 4.2 abgebildet sind. Die Autoren des Modells schließen sich dabei der Auffassung von Fayyad, Piatetsky-Shapiro und Smyth an und heben die Wichtigkeit von zyklischen und iterativen Prozessabläufen hervor. So weisen Chapman u. a. (2000, S. 10) weiter darauf hin, dass sich die Pfeile zwischen den Phasen des Modells nur auf die wichtigsten und häufigsten Prozessübergänge und nicht auf festgeschriebene Abläufe beziehen. Es solle am Ende einer Phase basierend auf dem Ergebnis dieser entschieden werden, welche Phase im Anschluss durchlaufen werden muss, wobei auch das Ende des Modells erneut Ausgangspunkt für weitere Data-Mining Projekte sein kann.

Im folgenden Teil werden die sechs Phasen des CRISP-DM Modells erläutert, um der Entwicklung des Vorgehensmodells eine fachlich fundierte Basis zu bieten. Da die letzten drei Phasen jedoch fast ausschließlich das eigentliche Data-Mining betreffen, werden diese nur kurz aufgeführt. Die für diese Arbeit relevantesten ersten drei Schritte gliedern sich in die Phasen *Business Understanding*, *Data Understanding* und *Data Preparation*.

¹NCR Systems Engineering Copenhagen (USA und Dänemark), DaimlerChrysler AG (Deutschland), SPSS Inc. (USA) und OHRA Verzekeringen en Bank Groep B.V. (Niederlande) (Chapman u. a. 2000, vgl.)

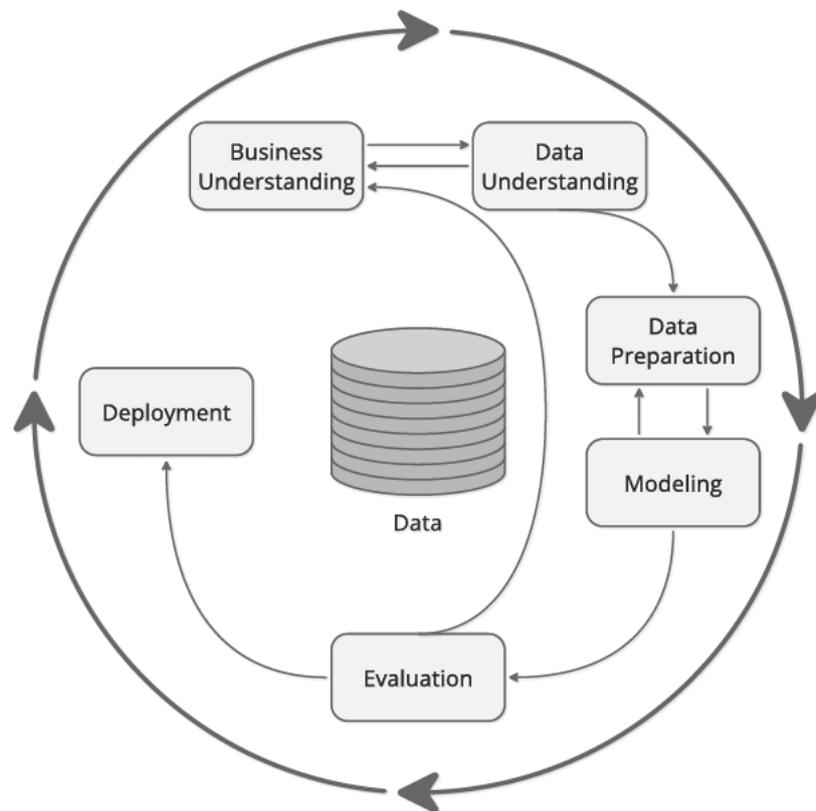


Abbildung 4.2: CRISP-DM Vorgehensmodell (vgl. Chapman u. a. 2000, S. 10)

1. Business Understanding

Die erste Phase des CRISP-DM Vorgehensmodells setzt sich aus vier grundlegenden Aufgaben zusammen. Zunächst müssen konkrete, durch das Data-Mining erreichbare Geschäftsziele bestimmt und ergebnisbeeinflussende Faktoren aufgedeckt werden. Dieser Schritt stellt die Ausgangsebene für einen erfolgreichen Abschluss des Projektes dar, da bei Vernachlässigung nur „die richtigen Antworten auf die falschen Fragen“ (Chapman u. a. 2000, S. 14) gefunden werden. Auch muss die Gesamtsituation detailliert erfasst und alle damit einhergehenden Informationen zusammengetragen und gesichert werden, um die Basis für einen einheitlichen Wissensstand zu schaffen, auf den in der zweiten Phase weiter aufgebaut wird. Eine weitere Aufgabe besteht darin, zusätzlich zu den übergeordneten wirtschaftlich ausgerichteten Geschäftszielen die konkreten technischen Ziele des Data-Minings zu bestimmen. Zuletzt wird ein zielorientierter Projektplan erstellt, der die weiteren Schritte des Projektes definiert.

2. Data Understanding

In der zweiten Phase des Modells wird ein umfassendes Datenverständnis angestrebt. Dafür müssen die Daten aus den, in der ersten Phase zusammengetragenen Projektressourcen beschafft und, wenn für das weitere Vorgehen nötig, geladen werden. Ein Laden der Datensätze kann jedoch der dritten Phase vorgehen und zu ersten erforderlichen Aufbereitungsmaßnahmen führen. Um anschließend ein tieferes Verständnis für die Besonderheiten und Zusammenhänge der Daten zu erlangen, müssen diese erkundet, analysiert und die Ergebnisse dokumentiert werden. Vorbereitend auf die dritte Phase, wird zuletzt die Datenqualität geprüft. Die Vollständigkeit und Korrektheit der Daten, sowie die Häufigkeit der Fehler sind hier entscheidende Faktoren für das weitere Vorgehen.

3. Data Preparation

Die dritte Phase des Modells beschäftigt sich aufbauend auf der Datenqualitätsanalyse mit der Datenaufbereitung. Ziel dieser Phase ist es, einen oder mehrere Datensätze zu erzeugen, die eine stabile und hochqualitative Grundlage für das Data-Mining bieten. Um dieses Ziel zu erreichen, können fünf Schritte durchlaufen werden. Basierend auf der bereits erzielten und noch angestrebten Datenqualität ist ein Rücksprung auf einen vorherigen Schritt dabei jederzeit möglich.

Zuerst wird eine Vorauswahl getroffen, um einerseits das Volumen der Daten zu verringern und andererseits die Qualität zu erhöhen. Dabei können sowohl einzelne Attribute als auch ganze Datensätze abgewählt werden, die für das Erreichen der Data-Mining-Ziele nicht relevant sind.

Der zweite Schritt liegt in der Datenbereinigung. Die Datenqualität wird dabei auf ein einheitliches, für das Data-Mining benötigtes Niveau angehoben. Fehlende Werte können beispielsweise durch Standardwerte ersetzt oder durch eine Schätzung angenähert werden.

Im dritten Schritt werden neue zusätzliche Daten konstruiert, die für das Erreichen der Data-Mining-Ziele von Bedeutung sind. So können durch Transformationen oder Ableitungen der Quelldaten neue Attribute oder Datensätze erstellt werden. Ein Beispiel hierfür wäre die Berechnung des aktuellen Alters, durch die Angabe eines Geburtsdatums.

Mithilfe der Datenintegration, dem vierten Schritt der Datenaufbereitungsphase, werden mehrere Datensätze miteinander kombiniert und zusammengefügt. Dieser Schritt kann

ebenfalls Aggregationen enthalten, welche die Komplexität der Daten weiter reduzieren. Im letzten abschließenden Schritt werden die Daten für das Modellierungswerkzeug passend formatiert und wenn nötig randomisiert.

4. Modeling

Die vierte Phase nutzt die nun vorbereiteten Daten, um ein Data-Mining-Modell zu erstellen und zu testen. Dazu müssen zuerst eine oder mehrere Data-Mining-Methoden (wie z. B. Klassifikation oder Decision Trees) ausgewählt und dann die vorbereiteten Daten in Trainings- und Testdaten aufgeteilt werden. Anschließend werden, mithilfe unterschiedlicher Konfigurationen der Data-Mining-Verfahren, aussagekräftige Modelle erstellt und in Bezug auf die technischen Ziele bewertet.

5. Evaluation

Es folgt die Evaluation und erneute Prüfung der erstellten Modelle in Bezug auf die definierten Geschäftsziele. Basierend auf den Ergebnissen dieser Evaluation werden die nächsten Schritte des Projektes geplant, wobei es zu einem Abschluss oder einem erneuten Durchlauf der letzten Phasen kommen kann.

6. Deployment

Im letzten Schritt wird der Abschluss des Projektes realisiert. Es werden Punkte wie die Nutzung, Überwachung und Wartung der Data-Mining-Modelle und -Ergebnisse geplant und ein Abschlussbericht erstellt sowie der Projektverlauf rückblickend bewertet.

Allgemein lassen sich viele grundsätzliche Parallelen zum KDD Prozess herstellen, wobei CRISP-DM ein weitaus umfangreicheres und standardisiert dokumentiertes Vorgehensmodell ist, welches auch heute noch von Branchenführenden wie z. B. IBM genutzt wird (vgl. IBM 2021).

Sowohl die *Business* und *Data Understanding* Phasen als auch die *Data Preparation* Phase stellen wichtige vorbereitende Schritte dar, die im zu konzipierenden Vorgehensmodell als Prozess für die Sicherung der Datenqualität aufgenommen werden sollten. Die Schritte

helfen dabei, ein standardisiertes und konsistentes Datenmodell zu erstellen, welches potenziell vom ganzen Unternehmen genutzt werden kann. Im „BI and Analytics Maturity Model“ kann ein solches Vorgehen einen Sprung von den ersten beiden Entwicklungsstufen auf die dritte oder vierte Stufe darstellen und stimmt so mit dem angestrebten Ziel des zu entwickelnden Vorgehensmodells überein.

4.2.3 Iterativ evolutionäres Vorgehensmodell

Nach Kemper, Mehanna und Unger (vgl. 2006, S. 142) können iterative Vorgehensmodelle einige der Probleme von rein sequenziellen Modellen lösen. Obwohl keine sequenziellen Modelle vorgestellt wurden, können die Probleme, die bei der Anwendung solcher Prozessabläufe auftreten, dazu beitragen, die Idee der iterativen Gestaltung des Vorgehensmodells zu stärken.

So könne es unter anderem sein, dass nicht alle Anforderungen an das zu entwickelnde System vollständig und im Vorfeld der Entwicklung definiert wurden. Unvorhergesehene Änderungen der Rahmenbedingungen während der Entwicklung können außerdem situationsabhängige Reaktionen erfordern. Zudem könne es vorkommen, dass die Entwicklung eines vollständigen Systems aus zeitlichen Beschränkungen nicht möglich ist.

Kemper, Mehanna und Unger (vgl. 2006, S. 142 f.) schreiben weiter, dass iterative Modelle durch den Einsatz von sukzessiven Teilschritten diese Probleme lösen können. Prinzipiell handle es sich um „jeweils eigenständige Erstellungsprozesse von neuen Systemversionen, bei denen die jeweiligen Vorgängersysteme von Fehlern bereinigt, von nicht länger erforderlichen Funktionen befreit bzw. um zusätzlich verlangte Funktionen in Form neuer Module bereichert werden“ (Kemper, Mehanna und Unger 2006, S. 143).

Ein dafür beispielhaftes iterativ evolutionäres Vorgehensmodell ist in Abbildung 4.3 zu sehen und wird im Folgenden weiter erklärt.

Eine der Besonderheiten dieses Modells ist die fehlende Definition der benutzerspezifischen Anforderungen vor Beginn der Entwicklung. Es wird lediglich ein Konzept erstellt und die Rahmenbedingungen werden mithilfe von Grundanforderungen festgelegt. Auf dieser Basis wird eine erste Version des gewünschten Systems entwickelt, welche vorwiegend die wesentliche unterliegende Architektur beinhaltet und noch keine über die Grundanforderungen hinausgehende Funktionalität bietet. Diese Version wird anschließend den Anwendern ausgeliefert und über das eingeholte Feedback wird eine erneute

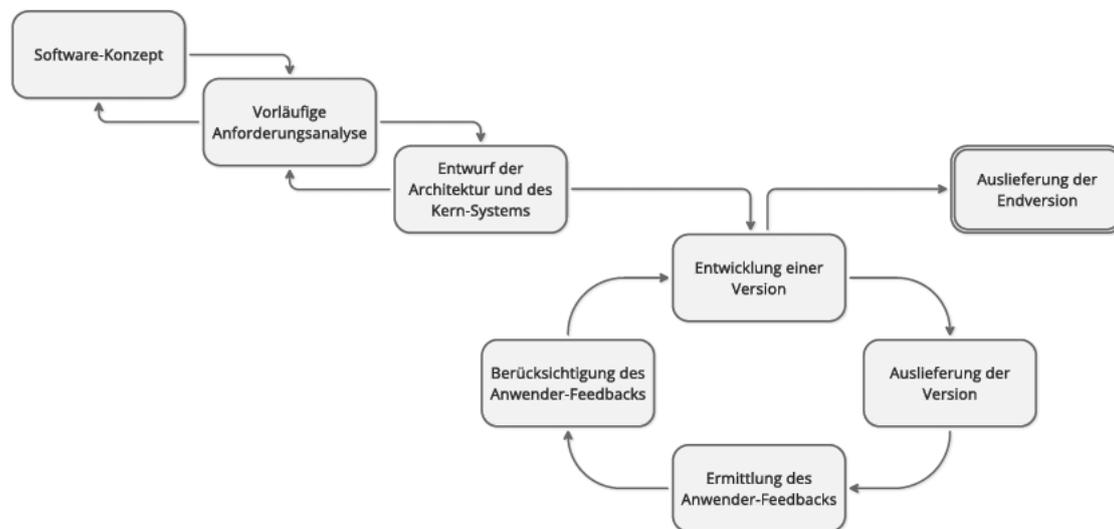


Abbildung 4.3: Evolutionäres Vorgehensmodell (vgl. Kemper, Mehanna und Unger 2006, S. 144)

Version erstellt. Dieser Zyklus von Entwicklung einer Teilversion, Auslieferung der Version, Ermittlung von Feedback und Formen von neuen Anforderungen wiederholt sich, bis der Status des Systems sowohl für die Auftraggeber als auch für die Anwender ausreichend ist. Zuletzt wird die endgültige Version erstellt und ausgeliefert, sodass die Entwicklung abgeschlossen werden kann (vgl. Kemper, Mehanna und Unger 2006, S. 144).

Nach Kemper, Mehanna und Unger (vgl. 2006, S. 144 f.) liegen die Vorteile eines evolutionären Modells vor allem in der Entwicklungsgeschwindigkeit. Zum einen könne innerhalb von kurzer Zeit ein erster Entwurf an die Anwender verteilt werden, sodass die weitere Entwicklung abhängig vom Feedback ebenfalls schnell fortgesetzt werden kann. Zum anderen seien Verbesserungen oder Änderungen während der Entwicklung problemlos möglich und durch die Veröffentlichung von Teilversionen schnell einsatzbereit.

Andererseits könne das Fehlen von Gesamtanforderungen auch dazu führen, dass die Basisarchitektur aufgrund von Änderungen der Anforderungen während der Entwicklung fehlerhaft wird und eine vollständige Neuentwicklung erforderlich ist, so Kemper, Mehanna und Unger (vgl. 2006, S. 145).

Da das zu entwickelnde Vorgehensmodell eine rechtzeitige Ablösung von veralteten, jedoch noch funktionsfähigen Prozessen vorsieht, sind schnelle und temporäre Entwürfe hier nicht nötig. Auch eine drohende Neuentwicklung durch zu starke Änderungen der

Anforderungen sollte aus zeitlichen Gründen verhindert werden, sodass eine ausschließlich auf dem Feedback der Anwender beruhende Entwicklung vermieden werden sollte.

Ein weiterer möglicher Kritikpunkt liegt in dem Abschluss des evolutionären Vorgehensmodells. Es wird zu Beginn der Entwicklung kein eindeutiges Ziel festgelegt, sodass sich der Anspruch an die endgültige Version im Laufe der Umsetzung ändern kann und der zeitliche Rahmen des Projektes angepasst werden muss. Dieses Problem kann im zu entwickelnden Vorgehensmodell umgangen werden, wenn bereits vor der iterativen Entwicklung die Ziele klar definiert wurden (Punkt 5 der nichtfunktionalen Anforderungen).

4.2.4 Phasenorientiertes BI-Vorgehensmodell

Das phasenorientierte BI-Vorgehensmodell löst die in Unterabschnitt 4.2.3 aufgeführten Probleme und definiert ein für Business Intelligence spezialisiertes Vorgehen zum Aufbau von BI-Lösungen. Das Modell sei nach Gluchowski, Gabriel und Dittmar (vgl. 2008, S. 256 f.) ebenfalls iterativ aufgebaut, da es bei Bedarf Rücksprünge in vorherige Phasen explizit ermögliche. Auch könne das Modell bei schon geleisteter Vorarbeit an einer beliebigen Stelle gestartet oder bei Bedarf mehrfach durchlaufen werden. Nach jeder der drei Phasen des Modells sei es zusätzlich möglich, eventuelle Datenqualitätsmängel in den unterliegenden Systemen zu beheben, falls diese in der Entwicklung zu Problemen führen.

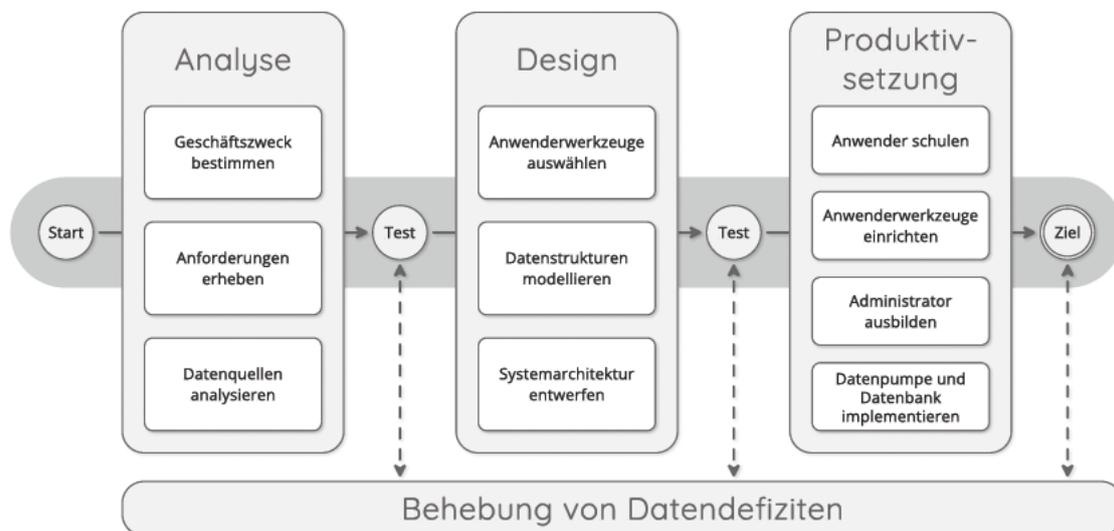


Abbildung 4.4: Phasenorientiertes BI-Vorgehensmodell (vgl. Gluchowski, Gabriel und Dittmar 2008, S. 257)

Abbildung 4.4 zeigt den Aufbau des phasenorientierten BI-Vorgehensmodells, welcher mit Verweis auf Gluchowski, Gabriel und Dittmar (vgl. 2008, S. 256 f.) im folgenden weiter erläutert wird.

1. Analyse

In der ersten Phase, der Analyse, werden in den anwenderbezogenen Aufgaben die angestrebten Geschäftszwecke bestimmt und die Anforderungen an die BI-Lösung definiert. Auf systemtechnischer Seite muss analysiert werden, ob die erforderlichen Informationen in den zur Verfügung stehenden Datenquellen vorhanden sind und wie darauf zugegriffen werden kann.

Diese Phase ähnelt den *Business Understanding* und *Data Understanding* Phasen des CRISP-DM Vorgehensmodells und sollte, wie bereits erwähnt, in das zu entwickelnde Modell einfließen.

2. Design

Anschließend wird in der zweiten Phase über die Wahl der Anwendersoftware entschieden und auf der Grundlage der Datenaktualität und -genauigkeit die inhaltliche Struktur der Daten bestimmt. Parallel dazu muss die Architektur des unterliegenden technischen Systems entworfen werden, wobei unternehmensspezifische Rahmenbedingungen und Vorschriften, die Wahl der Software oder Architektur beeinflussen können.

Da Unternehmen sowohl Richtlinien zur verwendbaren Software als auch unterschiedliche Datenquellen besitzen, sind die Wahl der Auswertungssoftware und die Gestaltung der Daten weitere wichtige Punkte, die in das Vorgehensmodell einfließen sollten.

3. Produktivsetzung

In der abschließenden dritten Phase muss die Software anwenderseitig eingerichtet und entsprechende Schulungen durchgeführt werden. Zusätzlich müssen alle erforderlichen technischen Systemkomponenten endgültig implementiert und die Administratoren für den Betrieb und die Wartung der Systeme geschult werden.

Zusammenfassend lässt sich sagen, dass das phasenorientierte BI-Vorgehensmodell gute vorbereitende Schritte sowohl in der *Analyse* als auch im *Design* auffasst, jedoch die eigentliche Entwicklung des Systems vor allem aus technischer Sicht betrachtet. Die Ausgliederung der Datenaufbereitung aus dem eigentlichen BI-Prozess stimmt nicht mit der für Business Intelligence aufgestellten Definition in Abschnitt 2.1 überein und sollte daher nicht übernommen werden.

4.3 Entwicklung des Vorgehensmodells

Die Entwicklung des Vorgehensmodells erfolgt auf Basis der in Abschnitt 4.1 definierten Anforderungen. Zusätzlich werden die in Abschnitt 4.2 vorgestellten Prozesse als ergänzende und strukturgebende Referenz dienen, welche das Modell weiter beeinflussen. Den finalen Schritt wird die Erstellung der grafischen Übersicht einnehmen, die es ermöglichen soll, den gesamten Prozessablauf zu überblicken und den Fortschritt einzuordnen.

Um dem Vorgehensmodell einen Wiedererkennungswert zu geben und während der Nutzung zu benennen, muss ein prägnanter Name gefunden werden, der die grundlegenden Funktionen des Modells beschreibt. Da ein Modell zur Optimierung und Ablösung von veralteten Business Intelligence Prozessen konzipiert werden soll, kann der daraus passend abgeleiteter Name „Model for Optimizing and Replacing Outdated Processes for Business Intelligence“ verwendet werden. Das zugehörige Akronym MOROP-BI soll als Kurzname für das Modell dienen und auch im weiteren Verlauf der Arbeit stellvertretend für das zu entwickelnde Vorgehensmodell stehen.

Da CRISP-DM, wie bereits in Unterabschnitt 4.2.2 erwähnt, ein viel benutztes und langjährig erprobtes Standardwerkzeug zur Umsetzung von Data-Mining Projekten ist, kann es gut als Referenzmodell für MOROP-BI genutzt werden. Vereinfacht gesehen kann CRISP-DM in drei übergeordnete Phasen unterteilt werden: der Vorbereitung, bestehend aus dem *Business* und *Data Understanding*, der Datenaufbereitung (*Data Preparation*) und dem Data-Mining (*Modeling*) mit anschließender *Evaluation* und *Deployment*. Indem die letzte Phase verallgemeinert als zielführender Prozess zum Verarbeiten der Daten verstanden wird, kann anhand dieses Schemas der grundlegende Aufbau für das neue Vorgehensmodell erstellt werden, welcher in Abbildung 4.5 festgehalten wurde.

Im Anschluss werden die hier angesprochenen drei Phasen in Bezug auf das neue Vorgehensmodell kurz erläutert, um sie dann ausführlicher in einzelne Schritte zu unterteilen

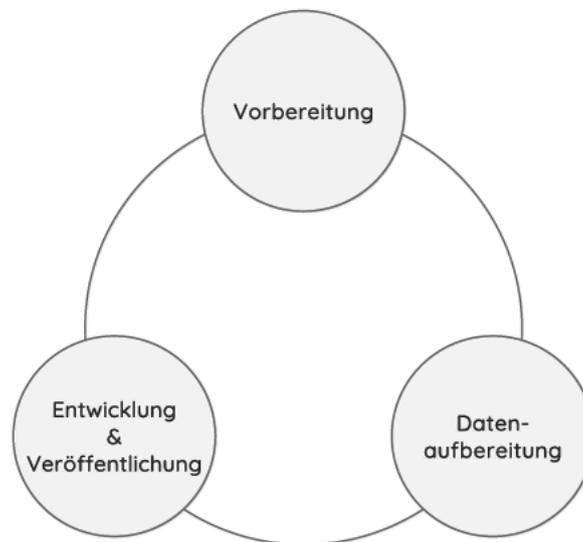


Abbildung 4.5: Grundlegende auf CRISP-DM basierte Phasen

und auszuarbeiten (Punkt 3 der nichtfunktionalen Anforderungen). Der zusammenfassende Aufbau des Modells kann schließlich im Unterabschnitt 4.3.4 nachvollzogen werden.

4.3.1 Vorbereitung

Um eine erfolgreiche Umsetzung von MOROP-BI zu gewährleisten, sollten, wie auch bei KDD und CRISP-DM, „die richtigen Antworten auf die falschen Fragen“ (Chapman u. a. 2000, S. 14) vermieden werden. Eine einführende Phase des Verständnisses und der Vorbereitung stellt somit einen wichtigen Aspekt im gesamten BI Entwicklungsprozess dar. Diese kann unter anderem durch die Berücksichtigung der ebenfalls vorbereitenden Phasen der KDD (Unterabschnitt 4.2.1) und CRISP-DM (Unterabschnitt 4.2.2) Prozesse erarbeitet werden.

Schritt 1 – Verständnis der Domäne

Da ein grundsätzliches Verständnis auf inhaltlicher Ebene sinnvoll für den weiteren Verlauf der Entwicklung ist, muss sich das entwickelnde BI Team mit der Domäne des Projektes vertraut machen und Hintergrundwissen zum jeweiligen Fachbereich sammeln. Das Anlegen einer internen Wissenssammlung kann außerdem dabei unterstützen, eine konsistente Wissensgrundlage zu schaffen, auf die sich im Laufe des Projektes jederzeit bezogen

werden kann (vgl. Louridas 2006, S. 88). Für die Erstellung einer solchen Sammlung ist es weiter sinnvoll, Domänenexperten hinzuzuziehen, die die inhaltliche Richtigkeit der Erkenntnisse gewährleisten und eventuell entscheidende Zusatzinformationen bereitstellen können (vgl. Müller und Lenz 2010, S. 252). Ein Fortschreiten im Projektablauf kann außerdem jederzeit dazu führen, dass die Wissenssammlung erweitert und unter Umständen berichtigt werden muss. Dieser Schritt schafft die Grundlage, um die in Abschnitt 4.1 definierten funktionalen Anforderungen erfüllen zu können.

Schritt 2 – Anforderungsanalyse

Im Anschluss an das Verständnis der Domäne, müssen im zweiten Schritt die Anforderungen an die finale BI-Auswertung definiert werden. Solche Anforderungsanalysen finden sich sowohl im iterativ evolutionären Vorgehensmodell (Unterabschnitt 4.2.3), als auch im phasenorientierten Vorgehensmodell (Unterabschnitt 4.2.4). Dazu gehören die Anforderungen an die technische Umsetzung, sowie die gewünschten Neuerungen und Optimierungen der Auswertung auf inhaltlicher Ebene. Eine Unterteilung in funktionale und nichtfunktionale Anforderungen kann in diesem Schritt hilfreich sein, wobei inhaltliche Punkte den funktionalen Anforderungen zugeordnet werden können und technische Themen den nichtfunktionalen. Damit die Entwicklung der Auswertung effizient und ergebnisorientiert erfolgen kann, ist die Gewichtung der Anforderungen ein weiterer sinnvoller Schritt, der um die Definition von zu erreichenden Zielen erweitert werden kann (Punkt 5 der nichtfunktionalen Anforderungen). Dieser Schritt wird vor allem zusammen mit den Auftraggebern sowie den Nutzern der Auswertung erarbeitet.

Schritt 3 – Analyse und Dokumentation der bisherigen Auswertung

Im dritten Schritt soll die bestehende Auswertung analysiert und dokumentiert werden, wobei dies in zwei Teilschritten erfolgen kann. Dieser Schritt ist speziell für den spezifischen Anwendungsfall von MOROP-BI und bezieht sich auf keines der bereits vorgestellten Vorgehensmodelle. Zum einen müssen die bisher genutzten Datenquellen auf ihre Vollständigkeit und Korrektheit analysiert und dokumentiert werden. Es wird im weiteren Verlauf davon ausgegangen, dass mehrere Datenquellen genutzt wurden und werden, obwohl es ebenfalls möglich ist, das Vorgehen auf nur eine Datenquelle anzuwenden. Zum anderen ist es nötig, die bisherige Auswertung zu analysieren und alle wichtigen Eigenschaften, insbesondere Berechnungen oder Kennzahlen zu verstehen und

zu dokumentieren. Bei der späteren Entwicklung der neuen Auswertung liefern diese Dokumentationen einen wichtigen Referenzpunkt, auf den jederzeit zurückgegriffen werden kann (Punkt 4 der funktionalen Anforderungen). Eine gesammelte Übersicht der in der alten Auswertung enthaltenen Logiken und Berechnungen bietet außerdem die Möglichkeit Fehler oder Trugschlüsse aufzudecken sowie weitere Optimierungspotenziale effizienter und schneller nutzbar zu machen.

Schritt 4 – Wahl der Auswertungssoftware

Da an diesem Punkt die Ziele der neuen Auswertung festgelegt und die Funktionen und Besonderheiten der alten Auswertung analysiert wurden, kann im vierten Schritt über die Software entschieden werden, in welcher die neue Auswertung entwickelt und dargestellt wird. Dieser Schritt leitet sich vorwiegend aus dem ersten Schritt der zweiten Phase des phasenorientierten BI-Vorgehensmodells (Unterabschnitt 4.2.4) ab.

Sollte im Unternehmen bereits eine Standardsoftware für BI Auswertungen existieren, so kann dieser Schritt, sofern die Software für den vorgesehenen Gebrauch geeignet ist, übersprungen werden. Andernfalls muss eine passende Software auf Basis der bisherigen Ergebnisse ausgewählt werden. Dabei sollten die Langlebigkeit und Stabilität der resultierenden Auswertungen im Vordergrund stehen (Punkt 1 und Punkt 2 der funktionalen Anforderungen). Außerdem müssen die Zugriffsberechtigungen, Bedienbarkeit, Anpassbarkeit und Entwicklungssicherheit der Software, sowie Unternehmensrichtlinien oder andere regulatorische Besonderheiten berücksichtigt werden.

Schritt 5 – Vorbereiten der Datenquellen

Der für die Vorbereitung abschließende fünfte Schritt des Vorgehensmodells liegt in der Erschließung von neuen oder der Optimierung der bestehenden Datenquellen. Dieser Schritt bereitet auf die zweite Phase des Modells (Datenaufbereitung) vor und baut auf den Erkenntnissen des dritten Schrittes auf. Sollte die Analyse der bisherigen Datenquellen gezeigt haben, dass diese weiterhin nutzbar sind, so können die Erkenntnisse weiter vertieft und die Datenquellen auf technischer Ebene optimiert werden. Geschwindigkeitsoptimierungen in der Datenbank oder die Anbindung an die Auswertungssoftware können weitere Schritte für eine technische Verbesserung sein, die sowohl in den Aufgabenbereich des BI Teams als auch der spezifischen IT-Abteilung des Unternehmens fallen können. Wenn die Datenquellen jedoch nicht für die neue Auswertung geeignet sind, so muss eine

oder wenn nötig mehrere neue erschlossen werden (Punkt 3 der funktionalen Anforderungen). Umfassende Datenfehler, Datenlücken oder veraltete und nicht mehr aktualisierbare Daten könnten Gründe für eine Neuorientierung sein. Auch hier gibt es viele Möglichkeiten, eine neue Datenbasis für die zukünftige Auswertung zu schaffen. Die Kombination von anderen schon bestehenden Daten stellt eine Möglichkeit dar, die mit relativ geringen Kosten verbunden ist, jedoch stark von dem Umfang und der Qualität der restlichen Unternehmensdaten abhängig ist. Schließlich muss, ähnlich wie im zweiten Schritt des CRISP-DM Vorgehensmodells (Unterabschnitt 4.2.2) und im ersten Schritt des phasenorientierten BI-Modells (Unterabschnitt 4.2.4), ein umfassendes Verständnis der Daten entwickelt werden, um diese in der nächsten Phase aufzubereiten. Dazu müssen Zusammenhänge innerhalb der Daten aufgedeckt und verstanden, sowie jegliche im Umgang entstehende Unklarheiten beseitigt werden. Die Erkenntnisse aus diesem Schritt sollten ebenfalls der Wissenssammlung zugeführt werden, um logische Fehler in der Auswertung zu vermeiden, welche durch inkonsistentes Verständnis der Daten entstehen können (Punkt 4 der funktionalen Anforderungen).

4.3.2 Schritt 6 – Datenaufbereitung

Sobald die vorbereitenden Schritte abgeschlossen sind, kann mit der Datenaufbereitungsphase begonnen werden. Die *Data Preparation* Phase des CRISP-DM Modells bietet hier einen optimalen Ablauf, um fehler- oder lückenhafte Daten in eine für die Auswertung optimale Form zu bringen (Punkt 3 der funktionalen Anforderungen). Gleichwohl unterteilt sich der nun sechste Schritt des Vorgehensmodells in fünf Teilschritte, die bereits im Unterunterabschnitt 4.2.2 aufgeführt und erklärt wurden.

Wie im gesamten Modell ist es hier für die finale Gesamtqualität notwendig zwischen den Teilschritten zu springen (Punkt 4 der nichtfunktionalen Anforderungen). Erkenntnisse aus den späteren Datenaufbereitungsschritten können jederzeit dazu führen, dass ein vorangegangener Schritt noch einmal ausgeführt werden sollte, um an früherer Stelle Verbesserungen einzufügen oder Mängel zu beseitigen.

Gerade die Datenaufbereitung ist ein stark auf sich aufbauender Prozess, dessen Endprodukt merklich variieren kann, je nachdem wie oft durch die Schritte iteriert wurde. Obwohl die Qualität von Daten in vielen Fällen immer weiter verbessert werden kann, muss der zeitliche Rahmen des Projektes eingehalten werden. Dafür ist es hilfreich, vor

dem Start der Datenaufbereitung (wahlweise bereits im Rahmen der Anforderungsanalyse) eine für die angestrebten Funktionen der Auswertung ausreichende Datenbasis zu definieren (Punkt 5 der nichtfunktionalen Anforderungen).

4.3.3 Entwicklung

Abschließend muss die Auswertung auf Basis des Wissensstandes und der vorbereiteten Daten entwickelt, getestet und veröffentlicht werden. Da die Entwicklung system- und unternehmensspezifisch, sowie von sich verändernder Software und technischen Fortschritten abhängig ist, kann das genaue Vorgehen variieren. Das Vorgehensmodell sollte dementsprechend ein generelles und generisches Vorgehen abbilden, welches ein hohes Maß an Anpassbarkeit bietet (Punkt 2 der nichtfunktionalen Anforderungen).

Schritt 7 – Entwicklung der Auswertung

Um mit der Entwicklung der Auswertung beginnen zu können, müssen die vorbereiteten Daten in die Auswertungssoftware geladen werden. Die Entwicklung erfolgt anschließend mit Hinblick auf die zuvor definierten Anforderungen und die Analyseergebnisse der alten Auswertung. Die Wissenssammlung liefert zusätzlich viele wichtige Informationen über die Berechnungen und Besonderheiten, die implementiert und beachtet werden müssen (Punkt 4 der funktionalen Anforderungen). Ableitend aus der im zweiten Schritt erfolgten Anforderungsanalyse, kann die Entwicklung in eine technische und eine inhaltliche Entwicklung aufgeteilt werden. Die richtige Konfiguration der Auswertungssoftware und Performanceoptimierungen der Datenbank sind nur Beispiele für Werkzeuge, um die nichtfunktionalen Anforderungen aus technischer Sicht umzusetzen. Um auf inhaltlicher Ebene die funktionalen Anforderungen zu erfüllen, kann es sinnvoll sein, bei der Erstinutzung der Software entsprechende vorbereitende Schulungen für das BI Team durchzuführen (3. Phase des phasenorientierten BI-Vorgehensmodells in Unterabschnitt 4.2.4). Auch hier ist es weiterhin wichtig, alle Berechnungsgrundlagen zu dokumentieren, um so den Ausgangspunkt für weitere einheitliche und strukturierte BI Auswertungen zu schaffen (Punkt 1 und Punkt 2 der funktionalen Anforderungen).

Schritt 8 – Testen der Auswertung

Da die Entwicklung einem iterativen Prozess folgen soll, muss, sobald der siebte Schritt begonnen hat und erste Teile der zukünftigen Auswertung entwickelt wurden, diese wiederholend getestet werden (Punkt 4 der nichtfunktionalen und Punkt 4 der funktionalen Anforderungen). Der Prüfprozess sollte sich auch hier an der Erfüllung der funktionalen und nichtfunktionalen Anforderungen orientieren. Zusätzlich dazu können, ähnlich dem iterativen Entwicklungsprozess des evolutionären Vorgehensmodells (Unterabschnitt 4.2.3) Anwender der Auswertung oder andere Domänenexperten in die Evaluation integriert werden, um weiterführende und individuellere Ergebnisse zu erhalten.

Schritt 9 – Abschluss und interne Veröffentlichung

Der gesamte Entwicklungs- und Prüfungsprozess kann sich einige Male wiederholen, bis der Stand der neuen Auswertung den angestrebten Zielen entspricht (Punkt 5 der nichtfunktionalen Anforderungen). Abschließend sollte eine zusammenfassende Projektdokumentation erstellt und intern veröffentlicht werden, um die Langlebigkeit und Stabilität der erstellten Auswertung weiter zu sichern (Punkt 1 und Punkt 2 der funktionalen Anforderungen).

Nach dem erfolgreichen Abschluss des Projektes sollte eine neue zukunftsichere, stabile und fehlerfreie Auswertung existieren, welche die Geschäftsprozesse optimal und gezielt darstellen und unterstützen kann. Außerdem sollte eine verlässliche und einheitliche Datenbasis verwendet oder geschaffen worden sein, auf der weitere Auswertungen aufgebaut werden können.

4.3.4 Zusammenfassender Aufbau des Vorgehensmodells

1. Verständnis der Domäne
 - 1.1 Wissenssammlung
 - 1.2 Domänenexperten
2. Anforderungsanalyse
 - 2.1 technische (nichtfunktionale) Anforderungen
 - 2.2 inhaltliche (funktionale) Anforderungen
3. Analyse und Dokumentation der bisherigen Auswertung
 - 3.1 Analyse der bisherigen Datenquellen
 - 3.2 Analyse der bisherigen Auswertung
 - 3.3 erste Optimierungen der Auswertung
4. Wahl einer Auswertungssoftware
5. Vorbereiten der Datenquellen
 - 5.1 Ausbau der Erkenntnisse aus Schritt 3.1
 - 5.2 Wenn erforderlich: Erschließen von neuen Datenquellen
 - 5.3 Verständnis der Daten
6. Datenaufbereitung
 - 6.1 Auswahl
 - 6.2 Bereinigung
 - 6.3 Konstruktion
 - 6.4 Integration
 - 6.5 Formatierung
7. Entwicklung der Auswertung
 - 7.1 technische (nichtfunktionale) Entwicklung
 - 7.2 inhaltliche (funktionale) Entwicklung
8. Testen der Auswertung
 - 8.1 Erfüllung der Anforderungen
 - 8.2 Evaluation mit Anwendern und Domänenexperten
9. Abschluss und interne Veröffentlichung

4.3.5 Grafische Umsetzung

Die Anforderungen an das Vorgehensmodell aus Abschnitt 4.1 definieren, dass die grafische Umsetzung des Modells klar strukturiert, verständlich, visuell ansprechend und inhaltlich anleitend sein soll (Punkt 1 der nichtfunktionalen Anforderungen). Angefangen bei der in Abbildung 4.5 bereits gezeigten vereinfachten Darstellung des CRISP-DM Prozesses, kann auch für das MOROP-BI Vorgehensmodell ein vergleichbarer dreiphasiger Prozess abgeleitet werden, der sich in die Phasen Vorbereitung, (Daten-) Aufbereitung und Entwicklung gliedert. Da diese drei Phasen jedoch nicht ausreichen, um eine inhaltlich leitgebende Übersicht zu erstellen, müssen sie weiter in ihre Teilschritte aufgeschlüsselt werden. Um das Diagramm weiterhin übersichtlich zu halten, werden die Teilschritte nach ihren übergeordneten Phasen gruppiert und räumlich voneinander getrennt. Außerdem können Pfeile, die sowohl zwischen den Phasen, als auch zwischen den Teilschritten eingefügt werden, die Übergänge verdeutlichen. Ein erster Entwurf des Modells ist in Abbildung 4.6 festgehalten.

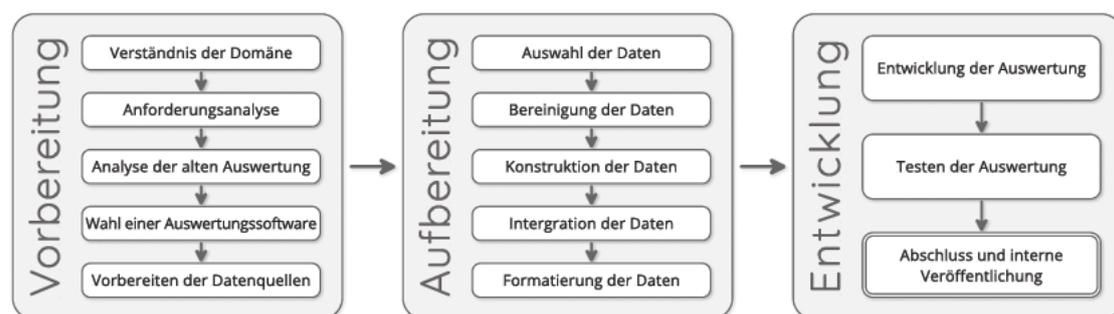


Abbildung 4.6: Erster Entwurf des MOROP-BI Vorgehensmodells

Dieser Entwurf beinhaltet jedoch keinen Hinweis auf die, in Punkt 4 der nichtfunktionalen Anforderungen geforderten, iterativen Prozessabläufe. Ein Wasserfallmodell würde sich trotz des eigentlich sequenziellen Ablaufs als grafische Anlehnung besser eignen, da die jeweiligen Schritte den generellen und gewünschten Ablauf des Modells verdeutlichen würden. Rückverbindungen durch optisch in den Hintergrund gerückte Pfeile können dieses Modell erweitern und so die iterative Natur des MOROP-BI Vorgehensmodells hervorheben.

Der fehlende Fokus auf die fortlaufende Dokumentation der Entscheidungen und Erkenntnisse während des Modelldurchlaufs stellt einen weiteren möglichen Kritikpunkt dar. Da

durch eine vollständige und aussagekräftige Dokumentation die Langlebigkeit der finalen Auswertung gesichert werden kann, sollte dieser Punkt in die Grafik aufgenommen werden (Punkt 1 der funktionalen Anforderungen). Ein passend beschrifteter und die Phasen einschließender Kreis kann abschließend helfen, diese Anforderung zu erfüllen.

Die finale grafische Umsetzung des Vorgehensmodells ist in Abbildung 4.7 zu sehen.

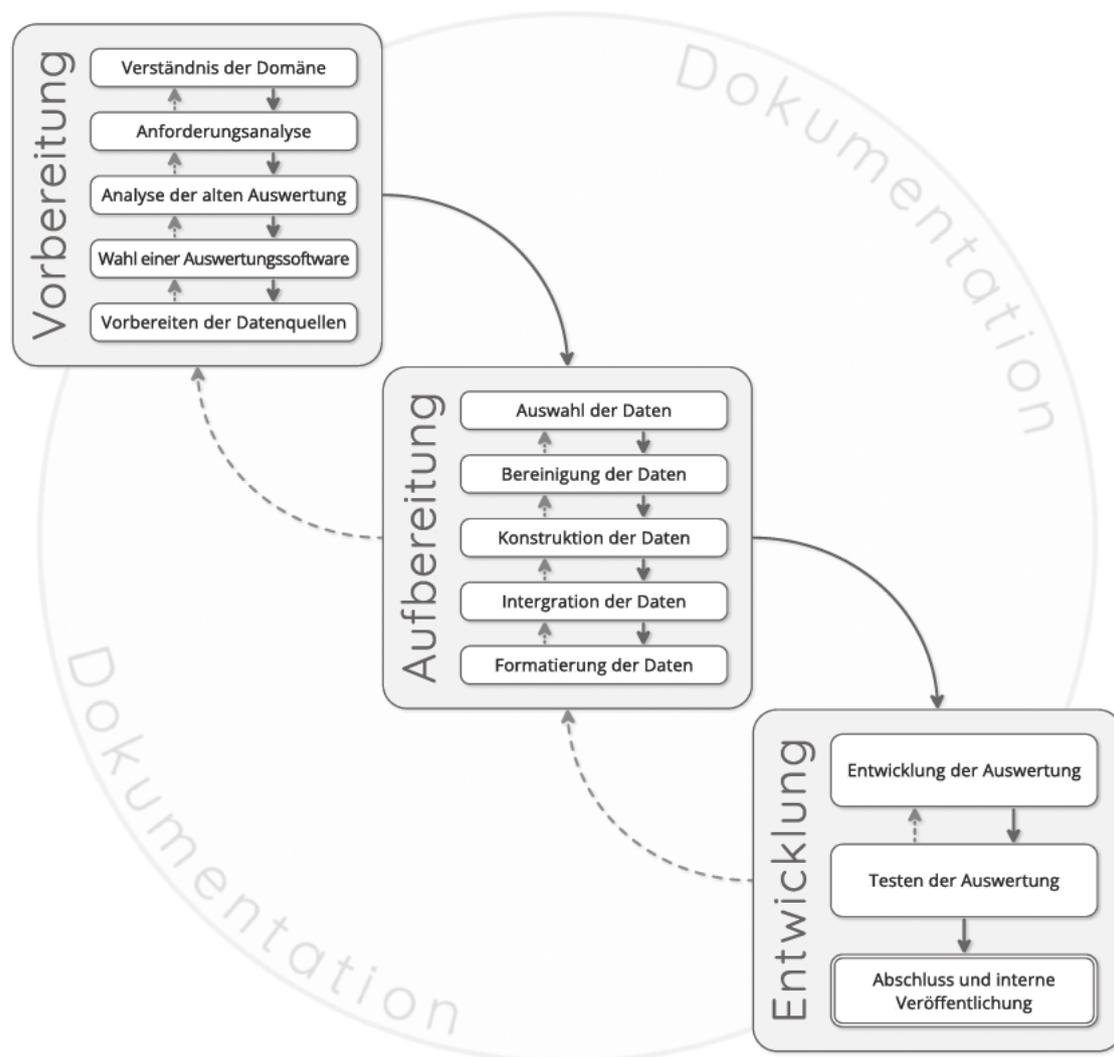


Abbildung 4.7: MOROP-BI Vorgehensmodell

4.4 Zwischenfazit

Um die Konzeption anhand der in Abschnitt 4.1 aufgestellten Anforderungen zu validieren, werden im Anschluss die jeweiligen Anforderungen mit den entwickelten Lösungen aufgeführt. Eine endgültige Bestätigung, ob das Vorgehensmodell für die vorgesehene Aufgabe geeignet ist, kann jedoch erst nach dem praktischen Einsatz erfolgen.

Nichtfunktionale Anforderungen

Punkt 1 der nichtfunktionalen Anforderungen forderte eine klar strukturierte und verständliche grafische Übersicht des Vorgehensmodells. In Unterabschnitt 4.3.5 wurde eine Grafik erstellt, welche sowohl die Teilschritte als auch die iterativen Prozesse des Modells passend darstellt. Räumliche Trennungen und eindeutig beschriftete Schritte sind zwei der Methoden, mit denen diese Anforderungen erfolgreich erfüllt wurden.

Die Anpassbarkeit an ähnliche Prozesse (Punkt 2) konnte ebenfalls erfüllt werden. Obwohl das MOROP-BI Vorgehensmodell eindeutige und für den spezifischen Einsatz besondere Schritte erfordert, kann unter anderem durch die freie Auswahl der Software, die offene Definition der Datenquellen und der iterativen Entwicklung eine ausreichende Anpassbarkeit gewährleistet werden.

Die klare Gliederung und Untergliederung der Teilschritte (Punkt 3) wird zusätzlich zur Ausarbeitung in der zusammenfassenden Übersicht des Modells (Unterabschnitt 4.3.4) festgehalten und kann daher gleichermaßen als erfüllt angesehen werden.

Auch die iterativen Prozesse (Punkt 4) wurden erfolgreich in das Vorgehensmodell implementiert und treten vor allem in der Datenaufbereitung (Unterabschnitt 4.3.2), der Entwicklung und dem Testen der Auswertung (Unterabschnitt 4.3.3) auf.

Zuletzt wurden in Punkt 5 der nichtfunktionalen Anforderungen klare Ziele gefordert. Die Definition dieser Ziele wurde primär in der Anforderungsanalyse (Unterabschnitt 4.3.1) des Modells eingeführt. Die Datenaufbereitungs- (Unterabschnitt 4.3.2) und Entwicklungsphasen (Unterabschnitt 4.3.3) mussten anschließend für die Erfüllung sorgen und konnten somit auch dieser Anforderung nachkommen.

Funktionale Anforderungen

Da das Erreichen der funktionalen Anforderungen in hohem Maße von der tatsächlichen Umsetzung des Modells abhängt, konnte lediglich eine optimale Gestaltung der Rahmenbedingungen angestrebt werden.

Um die Langlebigkeit (Punkt 1) und Stabilität (Punkt 2) der finalen Auswertung zu gewährleisten, wurde unter anderem ein Schwerpunkt auf die kontinuierliche Dokumentation aller Prozesse gerichtet. Darüber hinaus wurde bei der Auswahl der Auswertungssoftware besonders auf die Langlebigkeit und Stabilität dieser hingewiesen.

Auf die Wichtigkeit der Qualität der verwendeten Daten (Punkt 3) wurde hingegen sehr deutlich sowohl in der Vorbereitung (Unterabschnitt 4.3.1) als auch im Datenaufbereitungsprozess (Unterabschnitt 4.3.2) aufmerksam gemacht. Mithilfe der aus CRISP-DM stammenden Schritte sollte es somit möglich sein, eine gute Datenqualität zu erreichen.

Grundlage für eine fehlerfreie Auswertung – der abschließenden vierten funktionalen Anforderung – sollte das vorbereitende Verständnis der Domäne bilden. Expertenwissen, eine im Laufe der Umsetzung erweiterbare Wissenssammlung und die zuerst aufgestellten und abschließend zu validierenden Anforderungen ergänzen diesen Punkt weiter.

5 Praktische Umsetzung

In der folgenden praktischen Umsetzung des MOROP-BI Vorgehensmodells wird das entwickelte Modell phasenweise durchlaufen, wobei die Aufgaben und Ergebnisse zusammengeführt werden. Aus Gründen der Lesbarkeit werden Sprünge zwischen den einzelnen Phasen weitgehend ausgelassen und in einen linearen Verlauf zusammengefasst. Insbesondere die Datenaufbereitung und die Entwicklung des Dashboards mussten mehrere Iterationen durchlaufen, um ein Endergebnis zu erzielen, das den Anforderungen entsprach.

Um die Vertraulichkeit und Sensibilität der tatsächlichen Patienten- und Behandlungsparameter zu wahren, wurde für die folgenden Abbildungen ein künstlich generiertes Datenmodell verwendet. Da dieses Datenmodell dieselben Parameter und Kennzahlen verwendete wie das echte Modell, konnte während der Entwicklung der Auswertung je nach Bedarf zwischen den Modellen gewechselt werden, ohne dass die Auswertung geändert werden musste.

5.1 Verständnis der Domäne

Einleitend lässt sich sagen, dass die in diesem Fallbeispiel anfallende Domäne speziell für das Unternehmen ist und sich einzelne Ausprägungen und Spezifikationen von anderen Dialyseeinrichtungen unterscheiden können. Zusätzlich dazu war durch das bereits bestehende Arbeitsverhältnis schon vor dem Beginn der Umsetzung ein umfassendes Domänenwissen vorhanden, welches im ersten Schritt erweitert und vor allem zusammengefasst wurde.

Allgemein tauchen in dieser Auswertung ausschließlich Patienten auf, die an einem vollständigen Nierenversagen leiden und deshalb dialysiert werden müssen. Sobald die Filtrationsrate der Niere unter dem Schwellenwert von ungefähr 15 Prozent liegt, wird von

einem Nierenversagen gesprochen. Die sogenannte glomeruläre Filtrationsrate (GFR) der Niere liegt dann bei unter 15 ml/min.

Allgemein gesehen gibt es verschiedene Arten einen Dialysepatienten zu behandeln. Die in diesem Fallbeispiel am meisten angewandte Form der Dialyse ist die Hämodialyse (HD). Hier wird das Blut des Patienten über einen permanenten Gefäßzugang (meist Shunt) außerhalb des Körpers geleitet, um es mithilfe eines Dialysators zu filtern und dem Patienten anschließend wieder zuzuführen. Eine HD dauert ungefähr drei bis sechs Stunden und wird in der Regel dreimal pro Woche durchgeführt.

Im Gegensatz dazu steht die Peritonealdialyse (PD), bei der das Blut nicht mithilfe eines externen Filters von Giftstoffen gereinigt wird, sondern das Bauchfell des Patienten diese Aufgabe übernimmt. Hierfür wird eine Dialyseflüssigkeit in die Bauchhöhle des Patienten geleitet, welche durch Diffusion mit den abzutransportierenden Giftstoffen angereichert wird. Wenn die PD Behandlung kontinuierlich, also jeden Tag tags- und nachtsüber und zusätzlich dazu vom Patienten selbst entweder zu Hause oder bei der Arbeit durchgeführt wird, so wird diese Behandlung kontinuierliche ambulante Peritonealdialyse (CAPD) genannt. Im Gegensatz dazu wird von einer automatisierte Peritonealdialyse (APD) gesprochen, wenn der Patient nachts von einer Maschine die Bauchhöhlenflüssigkeit automatisch getauscht bekommt. Sollte der Patient zu Hause nicht die Möglichkeit haben, eine CAPD oder APD Dialyse durchzuführen, so kann diese ähnlich der HD meist dreimal pro Woche im Dialysezentrum als sogenannte intermittierende Peritonealdialyse (IPD) erfolgen.

Zusätzlich kann es vorkommen, dass Patienten meist aufgrund von Begleiterkrankungen andere extrakorporale Sonderverfahren¹ erhalten können. In diesem Fallbeispiel werden diese unterschiedlichen Blutreinigungsverfahren als Apheresen (ANs) zusammengefasst und meist einmal pro Woche durchgeführt.

Da die Anzahl der Behandlungstage pro Woche bei HD und PD Verfahren unterschiedlich sind, würden bei Auswertungen mit gemischten Behandlungsverfahren, die ungleichen Gewichtungen zu einem verfälschten Gesamtergebnis führen. Um diesen Effekt zu normalisieren, können die PD Behandlungen als sogenanntes HD-Äquivalent verstanden werden, indem die 7-Tage basierten APD und CAPD Behandlungen auf die 3-Tage basierten HD Behandlungen umgerechnet werden. HD-Äquivalente PD Zahlen ergeben sich demnach bei folgender Rechnung:

$$\text{IPD} + (\text{APD} + \text{CAPD}) \cdot \frac{3}{7}$$

¹Überbegriff für außerkörperliche Blutreinigungsverfahren (vgl. Klingele und Brodmann 2017, S. 240)

Eine weitere wichtige Unterscheidung der Dialysepatienten und -behandlungen liegt in der Klassifikation als chronisch, akut oder Gast. So gelten Patienten, mit einer chronischen Niereninsuffizienz als Stammpatienten, welche wie bereits beschrieben klare Dialysetermine haben, um die fehlende Funktion der Niere auszugleichen. Patienten, die an einer akuten Niereninsuffizienz leiden, müssen unter anderem aus Krankenhäusern in das Dialysezentrum verlegt werden, da sie schnell eine unvorhergesehene Dialyse benötigen. Die Funktion der Niere ist bei solchen Fällen z. B. durch eine Verletzung oder einen Flüssigkeitsverlust bei einer Operation stark eingeschränkt. Gastpatienten hingegen sind oft Urlauber, die über einen begrenzten Zeitraum in einem für sie neuen Zentrum behandelt werden.

Als weiterführende Literatur über die Verfahren und Besonderheiten der Dialyse kann das Buch „Einführung in die Nephrologie und Nierenersatzverfahren: Für Pflegende, Medizinstudenten und Assistenzärzte“ von Klingele und Brodmann (2017) empfohlen werden.

Auf die Erstellung einer gesonderten Wissenssammlung wurde zu diesem Zeitpunkt verzichtet, da das hier aufgeführte Domänenwissen als grundlegend im Unternehmen galt und bereits eine für andere BI Auswertungen angelegte Wissenssammlung existierte, die dieses Wissen abdeckte.

5.2 Anforderungsanalyse

Um die Anforderungen an die überarbeitete Version des wöchentlichen Berichts zu erfassen und auszuarbeiten, war es nötig, die beeinflussenden Faktoren zu finden und einen gemeinsamen Erwartungskonsens zu schaffen. In diesem Fall waren vorwiegend die Ideen der Nutzer der Anwendung entscheidend für den weiteren Verlauf. Die Anwender setzten sich aus den leitenden Ärzten der jeweiligen Zentren und dem regional übergeordneten Management zusammen. Nach einem Gespräch mit einem der zentrumsleitenden Ärzte wurden schließlich erörtert, welche Änderungen sinnvoll wären und welche Priorität diese Anforderungen haben. Weitere Gespräche mit dem Management ergänzten diese Anforderungen weiter und bildeten die Basis für die nutzerorientierten Anforderungen, welche nun weiter ausgeführt werden.

Als für alle Nutzergruppen notwendiger Punkt wurde die eindeutige Unterscheidung der Patientenzahlen von den Behandlungszahlen angegeben. Patientenzahlen bezeichnen dabei im gesamten Fallbeispiel die Anzahl der Patienten, die in der jeweiligen Woche min-

destens eine Behandlung erhalten haben. Behandlungszahlen hingegen definieren sich als Gesamtzahl an durchgeführten Behandlungen in der jeweils aktuellen Woche.

Die Unterscheidung begründete sich darin, dass die Patientenzahlen nicht grundsätzlich durch die Division der Behandlungszahlen durch die jeweilige Regeldialyseanzahl errechnet werden können. Durch Urlaube oder Krankenhausaufenthalte erscheinen Patienten teilweise unregelmäßig zur Dialyse und sorgen so für ein wechselndes Verhältnis von Behandlungen zu behandelten Patienten.

Vorgreifend zu der Analyse der alten Auswertung wurde sich außerdem eine bessere grafische Ansicht der zeitlichen Entwicklung aller Kennzahlen gewünscht. Da der Fokus der bisherigen Auswertung auf den reinen Behandlungszahlen der jeweiligen Woche und den durchschnittlichen Behandlungszahlen der letzten 4 bis 52 Wochen basierte und somit kein eindeutiger Trend der Zahlen zu erkennen war, wurde viel Wert auf die Einführung einer im Fokus stehenden grafischen Entwicklung gelegt.

Zusätzlich dazu stellte auch der grafische Vergleich zum Vorjahr ein elementar benötigtes Werkzeug dar, dass die momentane Entwicklung der Kennzahlen ins Verhältnis setzen würde.

Es stellte sich außerdem heraus, dass aus Sicht eines einzelnen Zentrums der Vergleich zu anderen Zentren eher unwichtig war. Dem gegenüber stand jedoch das Management, welches den Vergleich der Zentren, sowie die Gesamtheit der Patienten- und Behandlungszahlen als essenziell wichtig empfand. Daraus resultierte die Anforderung, dass die Auswertung anpassbar und personalisierbar sein sollte. Die Umsetzung als interaktives Dashboard würde es dem Anwender ermöglichen, nach einzelnen Zentren zu filtern und somit sowohl bei der Detail- als auch bei der Gesamtansicht ausreichend viele Informationen zugänglich zu machen.

Ableitend aus diesem zentrumsbasierten Filter sollten auch Filter für die unterschiedlichen Behandlungsverfahren und -klassifikationen, mit der optionalen Funktion der HD-Äquivalenzrechnung für PD Behandlungen eingeführt werden.

Um das Dashboard weiterhin anpassbar zu halten, sollte es dem Anwender ermöglicht werden, die zu betrachtende Woche zu ändern, um weiterführende Informationen zu den bereits vergangenen Wochen zu erhalten.

Außerdem musste aus technischer Sicht eine Auswertungssoftware verwendet werden, die die Umsetzung der genannten Anforderungen ermöglicht und zeitgleich dem Nutzer des Dashboards eine intuitive und leicht zu erlernende Bedienbarkeit bietet. Es sollte weiter möglich sein, dass die Auswertung automatisch jede Woche ihre Daten aktualisiert

und inhaltlich mit den Zahlen der bisherigen Berichte aus den einzelnen Dialysezentren übereinstimmt.

Zusammenfassend ließen sich folgende funktionale und nichtfunktionale Anforderungen formulieren:

5.2.1 Funktionale Anforderungen

1. Unterscheidung von Patienten- und Behandlungszahlen
2. Grafische Entwicklung aller Kennzahlen
3. Grafischer Vergleich zum Vorjahr
4. Filtern der Zentren
5. Filtern der Behandlungsverfahren und -klassifikationen
6. PD Zahlen optional auch als HD-Äquivalenz
7. Auswahl der zu betrachtenden Woche

5.2.2 Nichtfunktionale Anforderungen

1. anpassbare und personalisierbare Auswertungssoftware
2. intuitive Bedienbarkeit
3. automatisches Aktualisieren der Daten
4. inhaltliche Schlüssigkeit

5.3 Analyse und Dokumentation der bisherigen Auswertung

Da nun die Anforderungen an das Dashboard definiert wurden, konnte mit der Analyse und Dokumentation des veralteten wöchentlichen Berichts begonnen werden. Zuerst galt es die Frage nach der Weiterverwendbarkeit der bisherigen Datenquellen zu beantworten. Dafür muss jedoch zuerst der wöchentliche Prozess der Datensammlung erläutert werden.

Die in jedem Zentrum entstehenden Behandlungszahlen aus dem Zentrumsmanagementsystem werden, nachdem die aktuelle Woche abgeschlossen und die Zahlen vom leitenden Arzt bestätigt wurden, aggregiert und als PDF exportiert. Die Aggregation und der Export der Zahlen wurden seit Beginn der wöchentlichen Behandlungsmeldungen im Unternehmen über individuell für das Zentrum angelegte Berichte einer SQL Berichtserstellungssoftware (SAP Crystal Reports) realisiert. Sie beinhalteten die wöchentlichen Behandlungszahlen der letzten 52 Wochen, gruppiert nach den Behandlungsverfahren und erstreckten sich je nach Zentrum über ein bis drei Seiten mit unterschiedlichen und nicht standardisierten Layouts. Eine Unterscheidung der Behandlungsklassen fand bis auf wenige ausgewiesene Gastdialysen nicht statt. In Anhang A wird ein beispielhafter Wochenbericht aufgeführt.

Sobald ein Bericht zu Beginn der Folgewoche erstellt und geprüft wurde, musste dieser per E-Mail an den Auszuwertenden gesendet werden. Da die gesamte Berechnung des finalen wöchentlichen Berichts in einer zentralen Excel-Arbeitsmappe erfolgte, mussten die Behandlungszahlen aus jedem Zentrum nach dem Empfang in die richtige Stelle der Arbeitsmappe übertragen werden. Dieser Prozess wurde in der Vergangenheit bereits durch den Einsatz einer Texterkennungssoftware stellenweise automatisiert, war jedoch noch immer zeitaufwendig und fehleranfällig. Durch Urlaube oder andere Abwesenheiten der zuständigen Belegschaft in den Zentren verzögerte sich auch der Versand der erstellten Einzelberichte regelmäßig und sorgte für unvollständige oder verspätete Endreports.

Der wöchentliche manuelle Aufwand, der sowohl dezentral von Mitarbeitern der Dialysezentren als auch zentral von einem oder mehreren Auszuwertenden durchgeführt werden musste, sorgte für die Entscheidung, dass der gesamte bisherige Prozess der Datensammlung als veraltet und nicht weiter verwendbar angesehen wurde.

Da mit dieser Erkenntnis die Analyse der bisherigen Datenquellen abgeschlossen war, folgte im Anschluss die Analyse des wöchentlichen Berichts. Die resultierende Übersicht des Excel-Arbeitsblattes kann in Anhang B betrachtet werden.

Für jedes Zentrum wurde im Übersichtsblatt eine Datenreihe erzeugt, welche mehrere unterschiedliche Kennzahlen und visuelle Indikatoren beinhaltete. Es beinhaltete außerdem Informationen über die Zentren, darunter die Namen der Zentren und die der regionalen Leitung. Zentral im Fokus standen die Behandlungszahlen der aktuellen Woche, wobei wie im gesamten Bericht nicht zwischen den Behandlungsverfahren unterschieden wurde. Stattdessen wurden die PD Behandlungen als HD-Äquivalent gerechnet und zusammen mit den HD und Apheresezahlen addiert.

Ähnlich verhielt es sich mit der Klassifikation der Behandlungen in akute, chronische oder Gastbehandlungen. Die wenigen Ausnahmen, bei denen diese Angaben gesondert in den PDF-Berichten ausgewiesen wurden, wurden im Bericht ausgelassen und zählten somit nicht in das Gesamtergebnis.

Entwicklung				aktuelle Woche	Durchschnittswerte			
aktuelle Woche vs. letzten 4 Wochen (%)	letzten 4 Wochen vs. 12 Wochen (%)	letzten 12 Wochen vs. 24 Wochen (%)	letzten 24 Wochen vs. 52 Wochen (%)	14.02.2022 - 20.02.2022	4 Wochen (Ø)	12 Wochen (Ø)	24 Wochen (Ø)	52 Wochen (Ø)
0,4%	1,1%	-1,5%	1,5%	396	394	390	396	390
7,0%	1,3%	20,4%	-9,3%	179	167	165	137	151
-2,3%	3,4%	-4,0%	12,7%	146	150	145	151	134
-1,5%	2,5%	-0,8%	6,0%	374	380	370	373	352
0,0%	1,9%	5,3%	-0,7%	322	322	316	300	302
2,9%	-1,9%	14,9%	-5,2%	234	228	232	202	213
-2,9%	2,4%	0,0%	11,4%	136	140	137	137	123

Abbildung 5.1: Auszug aus dem veralteten Wochenbericht - durchschnittliche Behandlungszahlen und Trends

Zusätzlich zu den Behandlungszahlen der aktuellen Woche wurden die durchschnittlichen Behandlungen der letzten 4, 12, 24 und 52 Wochen, sowie prozentuale Vergleiche zwischen den Durchschnittswerten angezeigt. Die Vergleiche der Durchschnittszahlen dienten vor allem als visuelle Indikatoren über die Performance eines jeden Zentrums, wobei, wie in Abbildung 5.1 zu sehen, negative Abweichungen rot und positive Abweichungen grün gekennzeichnet wurden. Insgesamt zeigte der Anteil von durchschnittlichen zu aktuel-

len Zahlen, dass der Wochenbericht stark von der historischen Entwicklung der Zentren abhängig war und diese Darstellung vom Management als wichtig angesehen wurde.

Neben der Anzeige und dem Vergleich der Durchschnittszahlen der letzten Wochen, wurden außerdem weitere Vergleiche mit historischen Werten hergestellt. Wie in Abbildung 5.2 zu sehen ist, wurden die Behandlungszahlen der Vorjahreswoche und die des 4-Wochen Durchschnitts des Vorjahres mit den aktuellen Zahlen nach dem gleichen Muster wie in Abbildung 5.1 verglichen.

Vorjahr 2021			
KW 07 2021 vs. 2022	KW 07 2021	4 Wochen 2021 vs. 2022 (%)	4 Wochen (Ø) 2021
-2,7%	407	-7,0%	424
-1,6%	182	-9,9%	186
3,5%	141	-2,9%	154
-2,6%	384	-1,9%	387
-2,1%	329	3,2%	312
11,4%	210	1,7%	224
-9,3%	150	3,1%	136

Abbildung 5.2: Auszug aus dem veralteten Wochenbericht - Vorjahresvergleich

Als letzte visuelle Information über die Entwicklung des jeweiligen Zentrums dienten kleine Diagramme, die den Behandlungsverlauf der letzten 12 Wochen darstellten. Im Rahmen der Anforderungsanalyse wurden diese Diagramme bereits als nützlich, im Vergleich zur restlichen Auswertung jedoch als zu klein und ungenau empfunden.

Abschließend wurden unter den Einzelansichten der Zentren Summen bzw. Durchschnitte gebildet, die die Gesamtentwicklung aller Zentren darstellen sollte, und ebenfalls für das Management von Bedeutung war. Sollten einzelne Zentren durch ein verspätetes Zuschieken nicht in dem Bericht enthalten sein, so wurde die Berechnung der Gesamtsummen automatisch angepasst und das Fehlen kenntlich gemacht.

Sobald alle Behandlungsmeldungen eingetroffen waren und der Wochenbericht erstellt und geprüft wurde, konnte dieser als PDF exportiert und an alle zuständigen Empfänger per E-Mail verteilt werden.

Die Erkenntnisse dieser Phase des MOROP-BI Vorgehensmodells konnten in zwei Sachverhalte zusammengefasst werden. Zum einen war der bisherige Wochenbericht, abgesehen von der dezentralen und nicht vollständig einheitlichen Form, nur von einer vergleichsweise einfach aufgebauten Datenquelle abhängig, welche jedoch aufgrund des hohen manuellen Aufwandes ersetzt werden musste. Und zum anderen stellten sich die Logiken hinter den Berechnungen der Kennzahlen als nachvollziehbar und mit einfachen Mitteln reproduzierbar dar.

Nachdem die Ergebnisse der Analyse in Vorbereitung auf die Entwicklung hinreichend dokumentiert wurden, konnte im Anschluss mit der Auswahl der Auswertungssoftware begonnen werden.

5.4 Wahl einer Auswertungssoftware

Die Auswahl einer geeigneten Software zur Umsetzung und Nutzung der geplanten Auswertung kann von vielen unterschiedlichen Faktoren beeinflusst werden. Diese Faktoren ergeben sich vorwiegend aus der Tatsache, dass die Software anpassungsfähig und individuell gestaltbar sein muss, um allen Anforderungen gerecht zu werden. Weitere administrative oder technische Vorgaben ergänzen diese und bilden so den grundlegenden Rahmen für die gesuchte Software. Eine administrative Vorgabe könnte etwa das einfache Einrichten und Verwalten von Nutzern sein. Wobei die Unterstützung von mehreren Nutzern, die gleichzeitig und unabhängig voneinander auf die Auswertung zugreifen können, zu einer möglichen technischen Vorgabe zählen könnte.

Da diese Vorgaben universell für alle zukünftigen Auswertungen eines Unternehmens gelten sollten, wird eine Software benötigt, die nicht nur auf eine Anwendung spezialisiert ist. Das Unternehmen, in dem das Vorgehensmodell in der Praxis angewendet wurde, arbeitete bereits an der Einführung einer solchen Business-Intelligence-Software, wobei die Umsetzung des Vorgehensmodells frühzeitig und dadurch vor der unternehmensweiten Einführung ermöglicht wurde.

Die Wahl der Software fiel somit auf Qlik Sense Enterprise² der Firma QlikTech, einer webbasierten interaktiven Dashboard-Lösung mit Self-Service Möglichkeiten, die die Umsetzung der Anforderungen an die Auswertung vollumfänglich ermöglichte.

²<https://www.qlik.com/products/qlik-sense>

Zusätzlich zu erwähnen sei, dass in der Vergangenheit bereits andere Lösungen erprobt wurden, darunter Power BI³ von Microsoft und eine eigens programmierte Dashboard-Lösung, welche vor allem auf den Python Packages Dash⁴ und pandas⁵ aufbaute. Da Power BI nur sehr eingeschränkte Anpassungsmöglichkeiten bot und die Python-basierte Lösung einen hohen Verwaltungs- und IT-Aufwand hatte, wurden diese Optionen jedoch nach der Entwicklung mehrerer Prototypen verworfen.

5.5 Vorbereiten der Datenquellen

Da die Analyse der bisherigen Datenquellen gezeigt hat, dass diese aus unterschiedlichen Gründen nicht mehr zu verwenden sind, musste eine neue Datenquelle erschlossen werden. Ansatzpunkt dafür boten die Praxismanagementsysteme der jeweiligen Dialysezentren. Jedes System nutzt intern eine SQL Datenbank, in der alle Informationen über die Patienten und Behandlungen abgespeichert werden. Die schon angesprochene Berichtserstellungssoftware rief diese Daten ab, um die PDF-basierten wöchentlichen Berichte zu erstellen. Problematisch war jedoch, dass alle Datenbanken räumlich getrennt auf den lokalen Servern der Praxen betrieben werden. Das Unternehmen erkannte diese Problematik bereits vor einigen Jahren und entwickelte ein Datenbanksystem, welches täglich die aktuellsten Daten aus den einzelnen Zentren importiert und somit an einem Ort zugänglich macht.

In der Vergangenheit wurde die daraus resultierende SQL Datenbank bereits für kleinere Auswertungen genutzt, sodass vor Beginn dieses Schrittes schon ein grundlegendes Vorwissen über die Architektur der Datenbank bestand. Um die wöchentliche Anzahl an Behandlungen und daraus resultierend auch an Patienten herauszufinden, wurde primär nur eine Tabelle in der Datenbank gebraucht. Die sogenannte *Treatments* Tabelle beinhaltete je eine Zeile pro Behandlung, wobei die genauen Parameter und Eigenschaften der Behandlungen als Spalten organisiert waren. Es lagen insgesamt 67 mögliche Attribute vor, darunter die ID der Behandlung, die ID des Patienten, das Behandlungsdatum, die Behandlungsart, die Betriebsstättennummer (BSNR) des Dialysezentrums, die Angabe, ob die Behandlung erfolgreich abgeschlossen wurde, die lebenslange Arztnummer (LANR) des behandelnden Arztes, Informationen über die Dialysemaschine und medizinische Informationen über die Behandlung.

³<https://powerbi.microsoft.com/>

⁴<https://plotly.com/dash/>

⁵<https://pandas.pydata.org/>

Die bereits erwähnte Betriebsstättennummer ist eine festgelegte und eindeutig identifizierende Nummer, die in den entsprechenden Namen des Zentrums und der betriebsinternen Gruppierung nach der übergeordneten Region übersetzt werden musste.

In den hier genutzten Beispieldaten teilen sich die Regionen in Region A, B und C, wobei es pro Region vier Zentren gibt. Die Verknüpfung der BSNR mit den tatsächlichen Daten wurde durch eine zusätzliche Tabelle realisiert, welche alle nötigen Informationen beinhaltete und in der angesprochenen SQL Datenbank angelegt wurde.

Um die Daten aus der SQL Datenbank in Qlik zugänglich zu machen, musste eine Verbindung mithilfe des Qlik eigenen Microsoft SQL Server Connectors hergestellt werden. Dieser Schritt erwies sich als problemlos und es konnte im Anschluss eine vorläufige Erkundung der Daten durchgeführt werden. Dabei wurde vor allem auf die Vollständigkeit und Plausibilisierung der Attribute geachtet, um auf die Datenaufbereitung vorzubereiten.

Erste Prüfungen ergaben, dass die BSNR bei ungefähr 3 % aller Behandlungen fehlte. Da die Unterteilung der Zentren für die Funktion des Dashboards essenziell war, mussten die fehlenden Werte in der folgenden Phase vervollständigt werden.

Ein weiteres gefordertes Attribut fehlte vollständig. Die Behandlungsklasse tauchte zwar in der Hilfstabelle *TreatmentClass* auf, es war aber schnell ersichtlich, dass die vollständige Einführung eines aussagekräftigen Wertes ein aufwendiges und fehlerbehaftetes Unterfangen werden würde. Begründet wurde dies durch die Unvollständigkeit der Angaben in der *TreatmentClass* Tabelle sowie der Verwendung von offenen und begrenzten Zeiträumen innerhalb der Tabelle, als Angabe wann ein Patient welche Behandlungsklasse innehatte. Obwohl die Unterteilung der Behandlungsklasse eine vergleichsweise niedrige Priorität hatte und auch nach der Veröffentlichung des Dashboards erfolgen konnte, so sollte aus Gründen der Zukunftssicherheit trotzdem eine geeignete Lösung gefunden werden.

Jegliche Optimierungen der Datenbank wurden nach der Datenaufbereitung vorgenommen, da diese von der finalen Architektur der SQL Abfrage abhängig waren (siehe Unterabschnitt 5.6.3).

5.6 Datenaufbereitung

Damit mit der Entwicklung des Dashboards angefangen werden konnte, mussten die beiden, in der vorläufigen Erkundung aufgedeckten Datenlücken genauer analysiert und im Anschluss beseitigt werden. Obwohl Qlik eine eigene Ladefunktion für SQL Abfragen zur Verfügung stellte, gestaltete sich die Entwicklung der Abfrage in Microsofts SQL Server Management Studio (SSMS) als schneller und einfacher. Zusätzlich dazu sollte der Datenaufbereitungsprozess vom eigentlichen Entwicklungsprozess getrennt werden, um die Wartung und Weiterentwicklungen zu vereinfachen. Die logische Trennung beider Bereiche wurde mit der Erstellung einer Sicht in der Datenbank realisiert. Das Qlik Ladeskript musste dann lediglich alle verfügbaren Werte aus der entsprechenden Sicht laden, um die aktuellsten Behandlungen zu erhalten.

Um die Sicht zu erstellen, mussten die fünf Schritte der Datenaufbereitung durchlaufen werden. Da in Folge dieser Durchgänge viele neue Erkenntnisse über die Besonderheiten und Qualitätsmerkmale der Daten gewonnen wurden, wurde oft zwischen den Schritten gewechselt, bis die gewünschte Datenqualität erreicht war. Diese ergab sich aus unterschiedlichen Qualitätsmerkmalen, die im Laufe der Aufbereitung wiederholend geprüft und validiert wurden. Es wurde beispielsweise auf die Vollständigkeit und Korrektheit der benötigten Attribute geachtet, um eine lückenlose Datenbasis zu schaffen. Gleichermaßen wurden die resultierenden Behandlungszahlen mit den manuellen Wochenberichten abgeglichen, um bei Abweichungen mithilfe von detaillierteren Analysen einzelner Patienten Aufschlüsse über die Besonderheiten zu erhalten.

Die Datenaufbereitungsphase wird, um eine bessere Lesbarkeit und logische Stringenz zu erreichen, im weiteren Verlauf in drei Schritte zusammengefasst.

Zuerst wurde eine Vorauswahl getroffen, aus der eine Basisabfrage der Behandlungen resultierte. Dann wurden die fehlenden Werte konstruiert und ergänzt, um zuletzt die Abfragen zusammenzuführen und abschließend als Sicht in der Datenbank verfügbar zu machen.

Da die hier verwendeten Bezeichnungen für einige Werte (z. B. Behandlungsart und BSNR) von den tatsächlichen Attributbezeichnungen der Datenbank abwichen, befindet sich im Anhang C eine erklärende Übersicht aller in der Abfrage vorkommenden Attribute.

```
SELECT
  T.idPatientTreatment,
  T.fkPatientMasterdata,
  IIF(T.TreatmentTypeGroup IN ('HD', 'AN'), T.TreatmentTypeGroup,
    T.TreatmentTypeShortCut) AS TreatmentTypeGroup,
  T.DialysisType,
  T.PatientCode,
  T.DialysisShort,
  T.TreatmentDate,
  T.SiteNumber
FROM Treatment T
WHERE T.fkPatientMasterdata != -1
AND T.IsCompleted = 1
AND (T.TreatmentTypeGroup IN ('HD', 'AN') OR T.TreatmentTypeShortCut IN
  ('IPD', 'APD', 'CAPD'))
AND YEAR(T.TreatmentDate) >= YEAR(GETDATE()) - 2
```

Auflistung 5.1: Basis Abfrage der Behandlungen (*TreatmentCTE*)

5.6.1 Auswahl

Da sowohl für das Dashboard als auch für die Aufbereitung nicht alle der möglichen Spalten der *Treatments* Tabelle benötigt wurden, konnte eine Vorauswahl getroffen werden, die lediglich acht resultierende Attribute bereitstellte: *idPatientTreatment*, *fkPatientMasterdata*, *TreatmentTypeGroup*, *DialysisType*, *PatientCode*, *DialysisShort*, *TreatmentDate* und *SiteNumber*. Es wurde sich in Folge dieser Vorauswahl dazu entschieden, eine gleiche Datenbasis für alle Abfragen zu schaffen, die auf die Behandlungen der *Treatments* Tabelle Bezug nehmen. Umgesetzt werden konnte dies durch die Verwendung einer Common Table Expression (CTE), welche die Abfrage übersichtlicher machte und dazu führte, dass Anpassungen der *Treatments* Abfrage konsistent im gesamten Dokument übernommen wurden. Da CTEs bei jeder Referenz neu berechnet werden, konnten keine Performanceverbesserungen festgestellt werden. Die Verwendung einer temporären Tabelle, die zu Beginn mit den entsprechenden Daten befüllt werden würde, wäre unter diesem Aspekt die bessere Wahl gewesen, eignete sich jedoch nicht, da temporäre Tabellen nicht in Sichten verwendet werden können.

Zusätzlich zu der Auswahl der benötigten Spalten wurde das Ausmaß der Daten mithilfe einiger Filter eingeschränkt. Wie in Auflistung 5.1 zu sehen wurden Patienten mit einer ID von -1 herausgefiltert, da diese noch nicht vollständig im Praxismanagementsystem angelegt waren. Es wurden außerdem ausschließlich Behandlungen aufgenommen, die als abgeschlossen gekennzeichnet wurden und eindeutig als HD, IPD, APD, CAPD oder AN

zu erkennen waren. Da sowohl HD als auch AN Behandlungen ähnlich wie PD weiter unterteilt werden können – dies jedoch nicht erwünscht war – wurden sowohl in der Auswahl als auch im Filter zusätzlich zu der Spalte *TreatmentTypeGroup* ebenfalls die feingranularere Spalte *TreatmentTypeShortCut* genutzt. Zuletzt wurden, um den Datensatz zu verkleinern und damit die Performance zu steigern, ausschließlich Behandlungen aufgenommen, die aus Sicht des aktuellen Jahres maximal zwei Jahre in der Vergangenheit lagen. Dies reichte aus, da das Dashboard lediglich das Vorjahr der aktuell ausgewählten Woche anzeigen sollte (Punkt 3 der funktionalen Anforderungen). Abbildung 5.3 zeigt einen beispielhaften Auszug aus der erstellten *TreatmentCTE* Basisabfrage.

	idPatientTreatment	fkPatientMasterdata	TreatmentTypeGroup	DialysisType	PatientCode	DialysisShort	TreatmentDate	SiteNumber
1	1	1105	HD	ZD/Hämodialyse	Zentrum	HD	23.12.2021 00:00	020000100
2	2	1251	HD	ZD/Hämodialyse	Ambulant-LC	1	23.12.2021 00:00	020000300
3	3	671	HD	ZD/Hämodialyse	ambulant LC	1	23.12.2021 00:00	020000500
4	4	1189	HD	HD	Ambulant	1	23.12.2021 00:00	020000200
5	5	1340	HD	2013 Hämodialyse	ZD	5	23.12.2021 00:00	020000400
6	6	1217	CAPD	CAPD	ambulant	2	23.12.2021 00:00	020000300
7	7	1333	HD	2013 Hämodialyse	ZD	5	23.12.2021 00:00	020000200
8	8	1236	HD	2008/ZD/Hämodialyse	Ambulant	1	23.12.2021 00:00	020000500
9	9	869	HD	ZD/Hämodialyse	Ambulant Süd	1	23.12.2021 00:00	020000300
10	10	694	HD	ZD/Hämodialyse	ambulant LC	1	23.12.2021 00:00	020000400

Abbildung 5.3: Auszug aus der *TreatmentCTE* Abfrage

5.6.2 Bereinigung und Konstruktion

Wie im vorherigen Kapitel bereits beschrieben, wurden bei der Erkundung der Daten zwei problematische Attribute gefunden, die für den erfolgreichen Einsatz des Dashboards relevant waren. Zum einen fehlten ca. 3 % der Betriebsstättennummern in den Behandlungen und zum anderen gab es keine unmittelbar naheliegende und vollständige Angabe der Behandlungsklasse. Da die fehlenden Betriebsstättennummern eine höhere Priorität als die Unterteilung der Behandlungsklassen hatten, wurde dieses Problem zuerst behoben.

Ersetzen der fehlenden Betriebsstättennummern

Erneute Auszählungen, nun auf Basis der bereits gefilterten Behandlungen aus der *TreatmentCTE* Abfrage, ergaben eine Verbesserung der Fehlerrate auf ungefähr 0,6 %. Da jedoch eine nahezu ausnahmslose Vollständigkeit der Daten angestrebt wurde, musste

eine Möglichkeit gefunden werden, die noch fehlenden Werte zu ersetzen. Ein festgelegtes Zentrum pro Patient konnte im Übrigen nicht existieren, da teilweise aufgrund der Lokaltäten zwischen einzelnen Zentren gewechselt wurde.

Abhilfe schuf in diesem Falle die *TreatmentBillingItem* Tabelle, welche für jede durchgeführte Behandlung oft mehrere angesetzte kassenärztliche Leistungen aufführte. Als Leistungen zählten in diesem Beispiel meist Kostenpauschalen für Dialysebehandlungen, welche mit der von der Kassenärztliche Bundesvereinigung (KBV) vorgegebenen EBM Ziffer angegeben wurden. Beispiele können in Abbildung 5.4 gefunden werden.

Ergebnisse		Meldungen				
	idTreatmentBillingItem	fkPatientTreatment	fkPatientMasterdata	SiteNumber	ServiceDiget	Description
1	9	5	1340	020000400	13610	Dauerbetreuung bei Blutwäsche Dialyse und Sonder...
2	10	5	1340	020000400	40824	Kostenpauschale für Dialyse bei Versicherten ab voll...
3	11	5	1340	020000400	40834	Zuschlag zur Kostenpauschale 40824, 40826, 40827...
4	19	9	869	020000300	13610	Dauerbetreuung bei Blutwäsche Dialyse und Sonder...
5	20	9	869	020000300	40823	Kostenpauschale für Dialyse bei Versicherten ab voll...
6	21	9	869	020000300	40829	Zuschlag zur Kostenpauschale 40823 oder 40825 be...

Abbildung 5.4: Auszug aus der *TreatmentBillingItem* Tabelle

Zusätzlich zu den in dieser Anwendung nicht benötigten Angaben zu eventuellen Leistungen und Krankenkassen, wurden abermals die Betriebsstättennummern der abrechnenden Zentren angegeben. Nachdem eine Abfrage aufgebaut wurde, die ein Ersetzen der fehlenden Werte mit denen aus der *TreatmentBillingItem* Tabelle ermöglichte, ergab sich, dass diese Lösung für 99,91 % aller fehlenden Werte aufkam. Obwohl dieses Ergebnis die Anforderungen an die Vollständigkeit der BSNR erfüllte, wurde zusätzlich als Ausfallsicherung ein Verweis auf das zuerst besuchte Unternehmen eines Patienten eingefügt, welches in der Tabelle *Patient* zu finden war. Sowohl die BSNR-Abfrage als auch die Ausfallsicherung und die Übersetzung in die entsprechende Gruppierung nach Region, Unternehmen und Zentrum wurden in eine CTE-Unterabfrage zusammengefügt, um diese an die finale Abfrage anzuhängen. Der resultierende SQL Code kann in Auflistung 5.2 nachvollzogen werden.

Hinzufügen der Behandlungsklassen

Wie eingangs bereits erwähnt, stellte das Erschließen der Behandlungsklassen infolge der Datenqualität und -organisation eine größere Herausforderung dar. Aufgrund der unterschiedlichen Importsysteme, welche die Datenbank mit den aktuellsten Informationen über die durchgeführten Behandlungen befüllte, musste die vom System automatisch

```

SELECT

T.idPatientTreatment AS fkPatientTreatment,
ISNULL(SN.Region, FBRL.Region) AS Region,
ISNULL(SN.LegalEntity, FBRL.LegalEntity) AS LegalEntity,
SN.Location

FROM TreatmentCTE T

--> Fallback SiteNumber
LEFT JOIN (
    SELECT B.fkPatientTreatment, B.SiteNumber FROM (
        SELECT
            A.fkPatientTreatment,
            A.SiteNumber,
            ROW_NUMBER() OVER(PARTITION BY A.fkPatientTreatment ORDER BY
                A.Counter, A.SiteNumber DESC) AS rownumber
        FROM (
            SELECT
                TBI.fkPatientTreatment,
                TBI.SiteNumber,
                COUNT(*) AS Counter
            FROM TreatmentBillingItem TBI
            GROUP BY TBI.fkPatientTreatment, TBI.SiteNumber
        ) A
    ) B WHERE B.rownumber = 1
) FBSN ON FBSN.fkPatientTreatment = T.idPatientTreatment

--> Region, LegalEntity, Location
LEFT JOIN SiteNumbers SN ON SN.SiteNumber = IIF(T.SiteNumber = '',
    FBSN.SiteNumber, T.SiteNumber)

--> Fallback Region, LegalEntity
LEFT JOIN (
    SELECT DISTINCT
        P.idPatientMasterdata,
        SN.Region,
        SN.LegalEntity
    FROM Patient P
    LEFT JOIN (
        SELECT DISTINCT SN.Region, SN.LegalEntity, SN.ClinicFk
        FROM SiteNumbers SN
    ) SN ON SN.ClinicFk = P.ClinicId
) FBRL ON FBRL.idPatientMasterdata = T.fkPatientMasterdata

```

Auflistung 5.2: kombinierte Abfrage des Behandlungsortes nach Region, Unternehmen und Zentrum (*RegionLegalEntityLocationCTE*)

angelegte *TreatmentClass* Tabelle genutzt werden, um Informationen über die Behandlungsklasse eines Patienten zu erlangen. In der *TreatmentClass* Tabelle wurden Zeiträume aufgelistet, in denen bestimmte Patienten eine Behandlungsklasse innehatten.

Ergebnisse		Meldungen					
	idTreatmentClass	fkPatientMasterdata	Begin	End	ShortCut	Description	
1	1	2	2020-01-01 00:00:00.000	NULL	Gast	Gastpatient	
2	2	4	2020-05-27 00:00:00.000	2020-05-30 00:00:00.000	Akut	Akutpatient	
3	3	4	2020-05-31 00:00:00.000	2020-08-09 00:00:00.000	Kurzzeit	Kurzzeitpatient	
4	4	4	2020-08-10 00:00:00.000	2021-12-03 00:00:00.000	Chronisch	Chronischer Patient	
5	5	5	2020-05-29 00:00:00.000	NULL	Chronisch	Chronischer Patient	
6	6	6	2020-05-15 00:00:00.000	2020-07-31 00:00:00.000	Kurzzeit	Kurzzeitpatient	
7	7	6	2020-08-01 00:00:00.000	NULL	Chronisch	Chronischer Patient	

Abbildung 5.5: Auszug aus der *TreatmentClass* Tabelle

Abbildung 5.5 liefert einen Auszug der wichtigsten in der *TreatmentClass* Tabelle enthaltenen Daten. Da diese Angaben jedoch teilweise widersprüchlich und nach ausführlicher Analyse der abweichenden Fälle auch fehlerhaft waren, mussten verschiedene Lösungen gefunden werden, welche kombiniert befriedigende Ergebnissen erreichten. Zum einen wurden zwei Versionen einer Abfrage verwendet, deren Daten aus der *TreatmentClass* Tabelle stammten. Diese Abfragen unterschieden sich in der Methode, wie die angegebenen Zeiträume interpretiert werden.

Einerseits wurden in der Tabelle klar abgeschlossene zeitliche Abschnitte genannt, bei denen das Vertrauen in die Korrektheit der angegebenen Informationen sehr hoch war. Damit war es möglich eine Abfrage zu erstellen, die ausschließlich mit den geschlossenen Zeiträumen arbeitete und in Auflistung 5.3 im oberen Bereich zu sehen ist.

Andererseits wurden Zeiträume ohne definiertes Enddatum angeführt, was darauf schließen ließ, dass dieser Behandlungsabschnitt noch nicht abgeschlossen war und damit bis zum Zeitpunkt der Abfrage Bestand hatte. Problematisch an dieser Angabe war, dass Patienten temporäre Behandlungsklassen wie Akut oder Gast teilweise noch Monate oder Jahre nach dem Beginn zugeordnet waren. Zeitintervalle dieser Länge waren bei chronischen Patienten hingegen normal, da ein Patient einmal als chronisch eingestuft normalerweise nur durch eine Transplantation diesen Status wieder verlieren kann. Im unteren Bereich der Auflistung 5.3 ist die Umsetzung dieses offenen Endes des Zeitraumes dokumentiert.

Da die geschlossenen Intervalle den kleinsten Interpretationsspielraum boten, mussten diese für eine abschließende Berechnung der Behandlungsklassen bevorzugt werden. Soll-

```

--> Treatment class based closed time interval solution
SELECT
  T.idPatientTreatment,
  CASE
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Chronisch%' THEN 'Chronisch'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Kurzzeit%' THEN 'Chronisch'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Akut%' THEN 'Akut'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Gast%' THEN 'Gast'
  END AS TreatmentClass
FROM TreatmentCTE T
  LEFT JOIN TreatmentClass TC ON T.fkPatientMasterdata =
    TC.fkPatientMasterdata
WHERE (TC.Begin <= T.TreatmentDate AND T.TreatmentDate <= TC.End)
GROUP BY T.idPatientTreatment

--> Treatment class based solution with open time interval
SELECT
  T.idPatientTreatment,
  CASE
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Chronisch%' THEN 'Chronisch'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Kurzzeit%' THEN 'Chronisch'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Akut%' THEN 'Akut'
    WHEN STRING_AGG(TC.ShortCut, ' ') LIKE '%Gast%' THEN 'Gast'
  END AS TreatmentClass
FROM TreatmentCTE T
  LEFT JOIN TreatmentClass TC ON T.fkPatientMasterdata =
    TC.fkPatientMasterdata
WHERE (TC.Begin <= T.TreatmentDate AND T.TreatmentDate <= ISNULL(TC.End,
  GETDATE()))
GROUP BY T.idPatientTreatment

```

Auflistung 5.3: Abfrage der Behandlungsklassen nach *TreatmentClass* Tabelle (Teil der *TreatmentClassCTE*)

te für die spezifische Behandlung kein geschlossener Zeitraum zugeordnet werden können, so konnte ein offener genutzt werden, solange dieser als chronisch deklariert war. Die bestehenden, dem Import aus den unterschiedlichen Zentren geschuldeten Datenlücken machten es jedoch erforderlich, eine zusätzliche Lösung für das Gewinnen der Behandlungsklasse zu finden.

Nach einigen ausführlichen Tests konnte diese mithilfe von drei sekundären und bereits in der Vorauswahl gezeigten Behandlungsattributen realisiert werden. Die Spalten *Dialysis-Type*, *PatientCode* und *DialysisShort* der *Treatments* Tabelle eigneten sich, um mithilfe der in ihnen enthaltenen, von Zentrum zu Zentrum unterschiedlichen und nicht standardisierten Informationen eine Schätzung der Behandlungsklasse zu ermöglichen. Die

```

--> Treatments based text matching approach
SELECT
  T.idPatientTreatment,
  CASE
    WHEN T.DialysisType LIKE '%[Gg]ast%' THEN 'Gast'
    WHEN T.DialysisType LIKE '%FP' THEN 'Gast'
    WHEN T.DialysisType LIKE '%[Aa]kut%' THEN 'Akut'
    WHEN T.DialysisType LIKE '%[Kk]ranke%' THEN 'Akut'
    WHEN T.DialysisType LIKE '%[Ss]tat%' THEN 'Akut'
    WHEN T.DialysisType LIKE '%[Ii]ntensi%' THEN 'Akut'
    WHEN T.DialysisType LIKE '%KH%' THEN 'Akut'
    WHEN T.DialysisType LIKE '%[Cc]hro%' THEN 'Chronisch'
    WHEN T.DialysisType LIKE '%ZD%' THEN 'Chronisch'
    WHEN T.DialysisType LIKE '%[Aa]mbu%' THEN 'Chronisch'
    WHEN T.PatientCode LIKE '%[Gg]ast%' THEN 'Gast'
    WHEN T.PatientCode LIKE '%[Uu]rurlaub%' THEN 'Gast'
    WHEN T.PatientCode LIKE '%[Ss]tat%' THEN 'Akut'
    WHEN T.PatientCode LIKE '%[Aa]kut%' THEN 'Akut'
    WHEN T.PatientCode LIKE '%[Kk]ranke%' THEN 'Akut'
    WHEN T.PatientCode LIKE '%KH%' THEN 'Akut'
    WHEN T.PatientCode LIKE '%[Ii]ntensi%' THEN 'Akut'
    WHEN T.PatientCode LIKE '%[Aa]mbu%' THEN 'Chronisch'
    WHEN T.DialysisShort LIKE '%[Gg]ast%' THEN 'Gast'
    WHEN T.DialysisShort LIKE '%[Uu]rurlaub%' THEN 'Gast'
    WHEN T.DialysisShort LIKE '%[Ss]tat%' THEN 'Akut'
    WHEN T.DialysisShort LIKE '%[Aa]kut%' THEN 'Akut'
    WHEN T.DialysisShort LIKE '%[Kk]ranke%' THEN 'Akut'
    WHEN T.DialysisShort LIKE '%KH%' THEN 'Akut'
    WHEN T.DialysisShort LIKE '%[Ii]ntensi%' THEN 'Akut'
    WHEN T.DialysisShort LIKE '%[Aa]mbu%' THEN 'Chronisch'
  END AS TreatmentClass
FROM TreatmentCTE T

```

Auflistung 5.4: Abfrage der Behandlungsklassen nach *Treatments* Tabelle (Teil der *TreatmentClassCTE*)

auf übereinstimmenden Textmustern basierte Implementierung dieser Schätzung kann in Auflistung 5.4 nachvollzogen werden.

Die Schätzung konnte der bereits definierten Entscheidungslogik hinzugefügt werden, um die Datenlücken der *TreatmentClass* Tabelle auszugleichen. Eine zusammenfassende Ansicht der resultierenden, ebenfalls als CTE realisierten Unterabfrage wird in Auflistung 5.5 aufgeführt und der finalen Abfrage unter dem Namen *TreatmentClassCTE* zur Verfügung gestellt.

```
SELECT
    T.idPatientTreatment AS fkPatientTreatment,
    ISNULL(TCCI.TreatmentClass, IIF(TCOI.TreatmentClass = 'Chronisch',
        TCOI.TreatmentClass, ISNULL(TTM.TreatmentClass,
        TCOI.TreatmentClass))) AS TreatmentClass
FROM TreatmentCTE T

--> Treatment class based closed time interval solution
LEFT JOIN (...) TCCI ON TCCI.idPatientTreatment = T.idPatientTreatment

--> Treatment class based solution with open time interval
LEFT JOIN (...) TCOI ON TCOI.idPatientTreatment = T.idPatientTreatment

--> Treatments based text matching approach
LEFT JOIN (...) TTM ON TTM.idPatientTreatment = T.idPatientTreatment
)
```

Auflistung 5.5: kombinierte Abfrage der Behandlungsklassen (*TreatmentClassCTE*)

5.6.3 Integration und Formatierung

Sobald die benötigten Unterabfragen fertiggestellt waren, konnte im letzten Schritt eine finale Abfrage konstruiert werden, welche alle aus der Vorauswahl stammenden Behandlungen beinhaltet und sowohl die vollständigen Standortinformationen als auch die richtigen Behandlungsklassen für jede Behandlung einschloss. Die Abfrage wurde als Sicht innerhalb der Datenbank gespeichert und abrufbar gemacht. Durch die Architektur der Unterabfragen war die Hauptabfrage klar strukturiert und bot, sofern erforderlich, die Möglichkeit weitere Attribute hinzuzufügen.

Zuletzt wurden mithilfe des im SSMS verfügbaren Datenbankoptimierungsratgebers automatisch Indizes erstellt, welche die wiederkehrenden Abfragen auf die *Treatments* Tabelle optimierten und so die Abfragegeschwindigkeit verbesserten. Auflistung 5.6 zeigt abschließend die zusammengefasste Abfrage aller aufbereiteten und für das Dashboard benötigten Daten. Ein Einblick in die durch diese Abfrage entstehenden Daten kann in Abbildung 5.6 gewonnen werden.

```

WITH
    TreatmentCTE AS (...),

    RegionLegalEntityLocationCTE AS (...),

    TreatmentClassCTE AS (...)

SELECT DISTINCT

    T.fkPatientMasterdata,
    T.TreatmentDate,
    RLL.Region,
    RLL.LegalEntity,
    RLL.Location,
    T.TreatmentTypeGroup,
    TC.TreatmentClass,
    1 AS Counter

FROM TreatmentCTE T

    --> Region, LegalEntity, Location
LEFT JOIN RegionLegalEntityLocationCTE RLL ON RLL.fkPatientTreatment =
    T.idPatientTreatment

    --> TreatmentClass
LEFT JOIN TreatmentClassCTE TC ON TC.fkPatientTreatment =
    T.idPatientTreatment
    
```

Auflistung 5.6: Finale Abfrage der Behandlungen (CTEs wurden zusammengefasst)

	fkPatientMasterdata	TreatmentDate	Region	LegalEntity	Location	TreatmentTypeGroup	TreatmentClass	Counter
1	140	2020-01-22 00:00:00	B	Zentrum B4	Zentrum B4	HD	Chronisch	1
2	357	2020-10-09 00:00:00	C	Zentrum C1	Zentrum C1	HD	Chronisch	1
3	1015	2020-10-15 00:00:00	C	Zentrum C3	Zentrum C3	APD	Chronisch	1
4	171	2021-01-06 00:00:00	A	Zentrum A3	Zentrum A3	HD	Gast	1
5	298	2021-11-22 00:00:00	C	Zentrum C2	Zentrum C2	HD	Chronisch	1
6	1072	2020-03-15 00:00:00	B	Zentrum B4	Zentrum B4	CAPD	Chronisch	1
7	551	2020-03-25 00:00:00	A	Zentrum A3	Zentrum A3	HD	Chronisch	1
8	14	2021-03-03 00:00:00	A	Zentrum A2	Zentrum A2	HD	Chronisch	1
9	942	2021-12-29 00:00:00	C	Zentrum C2	Zentrum C2	HD	Chronisch	1
10	282	2020-09-25 00:00:00	B	Zentrum B2	Zentrum B2	HD	Chronisch	1
11	587	2021-12-06 00:00:00	A	Zentrum A3	Zentrum A3	HD	Chronisch	1
12	286	2021-11-24 00:00:00	C	Zentrum C2	Zentrum C2	HD	Akut	1
13	756	2021-01-20 00:00:00	A	Zentrum A4	Zentrum A4	HD	Chronisch	1
14	444	2020-07-03 00:00:00	C	Zentrum C4	Zentrum C4	HD	Chronisch	1

Abbildung 5.6: Auszug aus der finalen Abfrage

5.7 Entwicklung der Auswertung

Nach der erfolgreichen Aufbereitung der Daten konnte mit der Entwicklung der Auswertung begonnen werden. Die in Abschnitt 5.2 definierten Anforderungen an das Dashboard dienten dabei als Leitfaden und mussten sukzessiv umgesetzt werden. Im Detail lässt sich festhalten, dass die Umsetzung einen beständigen Wechsel zwischen den funktionalen und nichtfunktionalen Anforderungen erforderte und einige Features im Laufe der Entwicklung wiederholend angepasst werden mussten, um die finale Funktionalität zu gewährleisten. Wie schon in den vorangegangenen Phasen von MOROP-BI wird diese Entwicklung zur besseren Lesbarkeit als linearer Prozess dargestellt.

Das finale Dashboard kann im Anhang D gefunden werden und enthält sowohl eine ungefilterte als auch eine gefilterte Ansicht.

5.7.1 Laden der Daten im Dateneditor

Im Dateneditor von Qlik können Daten aus unterschiedlichen Quellen geladen und weiter verarbeitet werden. Wenn nötig, kann dieser Bereich in verschiedene Skripte unterteilt werden, um die auszuführenden Operationen übersichtlicher zu gestalten. Einmal eingerichtet, kann dieser Bereich nach der Veröffentlichung der App nicht weiter angepasst werden.

Qlik nutzt einige globale Variablen, um die Formatierung und Interpretation von Währungen, Zahlen und Zeiten regional anpassbar zu machen. Diese Variablen wurden bei der Erstellung des Dashboards automatisch angelegt und mussten angepasst werden, um Missinterpretationen innerhalb der Auswertung zu vermeiden (Punkt 4 der nichtfunktionalen Anforderungen). Gerade die Definition des Wochenstarts und der ersten Woche im Jahr mussten für das Wochenberichts-Dashboard angepasst werden. Der erste Tag einer Woche wurde somit als Montag definiert, und die erste Woche eines Jahres als die Woche mit dem ersten Donnerstag im Januar.

Da im Zuge von Abschnitt 5.5 bereits eine Verbindung zu der SQL Datenbank hergestellt wurde, konnte die erstellte Sicht ohne weitere Konfiguration von Qlik geladen werden. Damit den Spalten der geladenen Tabelle die richtigen Qlik spezifischen Datentypen zugewiesen werden konnten, mussten diese separat definiert werden (zu sehen in Auflistung 5.7). Qlik ermöglichte dies, indem das Laden der SQL Abfrage von dem Laden der Daten in die Auswertung getrennt wurde.

```
LIB CONNECT TO 'Microsoft_SQL_Server_[...]';

[Treatments]:
LOAD
    Num(fkPatientMasterdata) AS fkPatientMasterdata,
    Date(TreatmentDate) AS TreatmentDate,
    Text('Region ' & Region) AS Region,
    Text(LegalEntity) AS LegalEntity,
    Text(Location) AS Location,
    Text(TreatmentTypeGroup) AS TreatmentTypeGroup,
    Text(TreatmentClass) AS TreatmentClass,
    Num(If(TreatmentTypeGroup = 'APD' or TreatmentTypeGroup = 'CAPD', 3/7,
    1)) AS HDPDEquivalentFactor,
    Num(Counter) AS Counter;

SELECT
    T.[fkPatientMasterdata],
    T.[TreatmentDate],
    T.[Region],
    T.[LegalEntity],
    T.[Location],
    T.[TreatmentTypeGroup],
    T.[TreatmentClass],
    T.[Counter]
FROM [TreatmentsBA] T;
```

Auflistung 5.7: Ladeskript im Dateneditor von Qlik Sense

An dieser Stelle konnte außerdem die Funktionalität der HD-Äquivalenzrechnung vorbereitet werden (Punkt 6 der funktionalen Anforderungen). Wie in Auflistung 5.7 zu sehen, wurde in der von der SQL Abfrage separaten LOAD-Funktion eine zusätzliche Spalte *HDPDEquivalentFactor* eingefügt, welche als Alternative zur *Counter* Spalte diente und bei jeder APD oder CAPD Behandlung einen Wert von $\frac{3}{7}$ annahm. Bei der vollständigen Implementierung der Äquivalenzrechnung konnte somit bei der Anzahl der Behandlungen entweder die Summe aus *Counter* oder aus *HDPDEquivalentFactor* berechnet werden, um HD-äquivalente PD Zahlen zu erhalten.

Zusätzlich zu der Konfiguration der globalen Variablen und dem Ladeskript für die Behandlungen wurden von Qlik automatisch zusätzliche berechnete Kalenderfelder für das erkannte Datumfeld *TreatmentDate* eingefügt. Der sogenannte *autoCalendar* lieferte abgeleitete Attribute wie zum Beispiel *Year*, *Quarter*, *Month*, *Week*, *Day*, *YearsAgo*, *MonthsAgo* und *WeeksAgo*.

Nachdem alle Komponenten erstellt und angepasst wurden, konnten die vorbereiteten Daten, mit allen nötigen Formatierungen und zusätzlichen Attributen in das Qlik interne Datensystem geladen werden.

5.7.2 Layout des Dashboards

Aufgrund der Forderungen nach unterschiedlichen Kennzahlen, Filtern und Optionen, sowie nach Anpassbarkeit und intuitiver Bedienbarkeit der Auswertung, war es erforderlich, das resultierende Dashboard zuerst konzeptionell zu entwerfen (Punkt 2 der funktionalen und Punkt 1 und Punkt 2 der nichtfunktionalen Anforderungen). Um alle Features umzusetzen und eventuelle Sackgassen in der Entwicklung zu vermeiden, wurde sich deshalb intensiv mit der von Qlik bereitgestellten Dokumentation⁶ auseinandergesetzt.

Resultierend aus den Empfehlungen von Qlik wurde sich dafür entschieden, jegliche Inhalte auf einer einzelnen Seite darzustellen, um das Dashboard fokussiert und kompakt zu halten (QlikTech International AB 2022). Es musste jedoch beachtet werden, dass nicht zu viele Informationen auf zu kleinem Raum untergebracht werden. Um zum Beispiel zwischen Patienten und Behandlungen zu unterscheiden, sollten in den entsprechenden Visualisierungen Reiter eingeführt werden, die ein Umschalten zwischen den Zählweisen zuließen und somit das Dashboard visuell entlasten.

Da sich außerdem das lokale Schriftbild von links nach rechts und von oben nach unten richtet, war schnell klar, dass die Gesamtzahlen der Patienten und Behandlungen im oberen Bereich des Dashboards als hervorgehobene und separierte KPIs angezeigt werden sollten. Filter und andere Auswahlmöglichkeiten sollten hingegen links ausgerichtet werden, um eine zu anderen Webseiten gleichförmige und intuitive Bedienbarkeit zu ermöglichen (QlikTech International AB 2022). Zu den Auswahlmöglichkeiten zählten Schalter für das Ein- bzw. Ausblenden der Vorjahresansicht, sowie das Umschalten der HD-äquivalenten PD Behandlungszahlen (Punkt 3 und Punkt 6 der funktionalen Anforderungen). Alle durch diese Schalter beeinflussten Visualisierungen sollten sich beim Umschalten automatisch aktualisieren (Punkt 1 der nichtfunktionalen Anforderungen). Eine weitere Interaktionsmöglichkeit sollte in der Anzeige und Auswahl der aktuellen Woche liegen, welche durch ihre Wichtigkeit oben links im Dashboard platziert werden sollte (Punkt 7 der funktionalen Anforderungen).

⁶https://help.qlik.com/de-DE/cloud-services/Content/Sense_Helpsites/Home.htm



Abbildung 5.7: Layoutentwurf des Dashboards

Um die Größe und Position aller angesprochenen Elemente zu testen, wurde ein Mockup des Dashboards erstellt, welches in Abbildung 5.7 zu sehen ist. Qlik nutzte zur Positionierung und Skalierung der einzelnen Elemente ein Rastersystem, welches auf feinsten Stufe 24 Spalten und 12 Reihen zuließ. Da die Visualisierung der Patienten- und Behandlungszahlen zusammengefasst werden sollten, konnten drei weitere Visualisierungen untergebracht werden. Es lag nahe, dass einer der freien Bereiche für eine Tabelle genutzt wurde, die dem Aufbau und der visuellen Darstellung des ursprünglichen Wochenberichts nachempfunden war. Der ehemalige Bericht war einerseits beim Management sehr beliebt und andererseits bot das Dashboard noch keinen direkten Vergleich zu den alten Wochenberichten.

Die restlichen freien Plätze wurden erst bei der Erstellung der Visualisierungen gefüllt. Durch die Entwicklung ergab sich, dass sowohl eine grafische Entwicklung der Behandlungen pro Patient, als auch ein Wasserfalldiagramm der Patientenzahlen zwei aussagekräftige Auswertungsmöglichkeiten darstellten, die zusätzlich aufgenommen werden sollten.

5.7.3 Implementierung der Filter und Auswahlmöglichkeiten

Wie bereits bei den Überlegungen zum Layout des Dashboards angesprochen, musste eine Lösung gefunden werden, um die speziellen Auswahlmöglichkeiten des Dashboards zu realisieren. Durch das Filtern der Behandlungsklassen und -gruppen konnte das Dashboard ohne weitere Konfigurationen anpassbar und dynamisch gemacht werden (Punkt 5 der funktionalen Anforderungen). Qlik bot hier eine speziell für diesen Zweck vorgesehene Filterkomponente, welche in das Dashboard eingefügt und mit den Feldern *TreatmentTypeGroup* und *TreatmentClass* verknüpft werden konnte. Diese Filter schränkten bei der Auswahl die Gesamtheit der geladenen Daten ein und wirkten sich so auf alle Visualisierungen, die diese Daten nutzten, aus.

Anders verhielt es sich bei den Schaltern für das Vorjahr (Punkt 3 der funktionalen Anforderungen) und der HD-Äquivalenzrechnung (Punkt 6 der funktionalen Anforderungen), sowie der Auswahl der aktuellen Woche (Punkt 7 der funktionalen Anforderungen). Diese Bedienmöglichkeiten mussten mithilfe von Variablen realisiert werden, welche die entsprechenden Zustände speicherten und an die Visualisierungen weitergaben. Es wurden von Qlik bereitgestellte Schaltflächen verwendet, um die Werte der Variablen bei Betätigung zu verändern.

Umschalten des Vorjahres und der HD-Äquivalenzrechnung

Da für Anzeige des Vorjahres und der Berücksichtigung der HD-Äquivalenzrechnung lediglich jeweils zwei mögliche Zustände nötig waren, konnten für die Werte der Variablen Booleans genutzt werden. Somit wurde die Variable *ShowPriorYear* mit dem vorgegebenen Wert **True** () initialisiert und die Variable *HDEquivalent* mit **False** (). Die Schalter im Dashboard mussten anschließend für den Zustandswechsel bei der Betätigung verantwortlich sein (zu sehen in Auflistung 5.8) und je nach Auslösestatus den auf der Schaltfläche angezeigten Text anpassen.

Auswahl der Woche

Für die Auswahl der aktuell ausgewählten Woche wurde die Variable *WeeksAgo* eingeführt. Dies ermöglichte eine auf den *autoCalendar* angepasste Funktionalität, die zu wählende Woche nicht nach der Wochennummer, sondern nach der Woche relativ zur aktuellen Woche zu definieren.

```
// Bei Betätigung wird jeweils folgender Wert angenommen

// Schalter Vorjahr
If(ShowPriorYear = True(), False(), True())

// Schalter HD-Äquivalenzrechnung
If(HDEquivalent = True(), False(), True())
```

Auflistung 5.8: Schalterlogiken zum Umschalten der Variablen

Um die Vorteile dieser Methode zu verdeutlichen, soll angenommen werden, das Dashboard wird in Kalenderwoche 11 im Jahr 2022 geöffnet und es soll die Vorwoche angezeigt werden. Für die Berechnung der Behandlungen in Kalenderwoche 10 muss eine Aggregation der geladenen Daten stattfinden. Es läge nahe, eine Summe von *Counter* zu bilden, wobei als einschränkende Filter die Woche des Behandlungsdatums auf Kalenderwoche 10 und das Jahr auf 2022 festgesetzt werden würde. Um sowohl die ausgewählte Woche als auch das ausgewählte Jahr unabhängig von der Berechnung zu machen, könnten passende anderweitig anpassbare Variablen gewählt werden. Die Umsetzung in Qlik sähe wie folgt aus:

```
Sum({$<[TreatmentDate.autoCalendar.Week]={"$ (= aktuelleWoche )"}> *
    $<[TreatmentDate.autoCalendar.Year]={"$ (= aktuellesJahr )"}>}
    Counter)
```

Obwohl diese Umsetzung unterjährig reibungslos funktionieren würde, würde ein möglicher Jahreswechsel separate Berechnungsschritte bei der Anpassung der Variablen erfordern.

Die eigentliche Problematik ergäbe sich jedoch bei der grafischen Entwicklung eines Zeitraumes. Zeitliche Gruppierungen, wie Wochen oder Monate werden bei Qlik gesondert in der Konfiguration der Visualisierung vermerkt und beeinflussen die Berechnung der Kennzahl nur insofern, dass diese nun Werte für den gewünschten Zeitraum liefern muss. Eine beispielhafte und voll funktionsfähige Implementierung – an dieser Stelle ohne die Verwendung von Variablen – wäre folgende:

```
Sum({$<[TreatmentDate.autoCalendar.Week]={"> 5 <= 10"}> *
    $<[TreatmentDate.autoCalendar.Year]={ "2022"}>} Counter)
```

Sobald sich jedoch der gewünschte Zeitraum über einen Jahreswechsel erstrecke, gäbe es keine elegante Möglichkeit diesen Zeitraum über die oben genannte Methode darzustellen.

Die Verwendung einer relativen Woche wäre von dieser Problematik nicht betroffen, da Jahreswechsel und unterschiedliche Anzahlen von Wochen im Jahr keine Relevanz hätten. Übertragen auf das oben genannte Beispiel müsste aus Sicht von KW11 für die Anzahl der Behandlungen aus KW10, eine Variable eingesetzt werden, die angäbe, dass die gesuchte Woche eine Woche in der Vergangenheit liege. Der durch Qlik bereitgestellte *autoCalendar* würde ebenfalls für jedes Datum den Abstand zur aktuellen Woche liefern und so sowohl einzelne Zeitpunkte als auch Zeiträume darstellen können.

```
// WeeksAgo = 1
Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]={"> $(= WeeksAgo+5 )
    <= $(= WeeksAgo )" }>} Counter)
```

Für die Auswahl der Woche wurden zwei Schalter eingefügt, welche den Wert der Variablen *WeeksAgo* um eins erhöhen, bzw. um eins verringern. Der einzige zusätzliche Berechnungsschritt lag darin, den Wert der Variablen in einem geschlossenem Intervall von eins bis zehn zu beschränken. Die Anwender sollten lediglich in der Lage sein, die bereits abgeschlossenen Vorwochen zu analysieren und nicht die aktuell laufende Woche (Punkt 7 der funktionalen und Punkt 4 der nichtfunktionalen Anforderungen).

5.7.4 Implementierung der Visualisierungen

In Qlik werden jegliche Visualisierungen über Kombinationen von Dimensionen und Kennzahlen realisiert, wobei unterschiedliche Visualisierungen unterschiedliche Anzahlen an Dimensionen und Kennzahlen benötigen, um zu funktionieren. Dimensionen sind dabei die Felder oder Attribute der Daten und dienen der Gruppierung. Das Behandlungsdatum wäre eine solche Dimension und könnte beispielsweise in einem Liniendiagramm als Zeitachse fungieren.

Kennzahlen wiederum stellen die Aggregationen der entstehenden Gruppierungen dar. Wie bereits angesprochen wäre die Summe des *Counter* Feldes eine mögliche Kennzahl, um die Anzahl an Behandlungen zu erhalten.

Globale, durch einen Anwender des Dashboards ausgewählte, Filter können diese Aggregationen einschränken und damit ein personalisierbares und dynamisches Dashboard ermöglichen. Innerhalb der Aggregationen können zusätzlich dazu weitere Filter gesetzt oder auch die globalen Filter ignoriert werden, so wie es im vorherigen Beispiel der relativen Wochenzählung bereits zu sehen war.

KPIs

Für die Einbindung von KPIs bot Qlik eine eigene Komponente, die mit der anzuzeigenden Kennzahl gefüllt werden musste. Es wurde sich dazu entschieden, im oberen Bereich des Dashboards sieben KPIs zu platzieren, die dem Anwender die wichtigsten Kennzahlen präsentieren sollten.



Abbildung 5.8: KPIs im oberen Bereich des Dashboards

Zwei der KPIs wurden für die Anzeige der Patienten und Behandlungszahlen der aktuell ausgewählten Woche verwendet und sind auf der linken Seite in Abbildung 5.8 zu sehen (Punkt 1 der funktionalen Anforderungen). Die Berechnung der Behandlungen musste dafür um die Option der HD-Äquivalenz erweitert werden (Punkt 6 der funktionalen Anforderungen). Um dies umzusetzen musste eine, von der Variablen *HDPDEquivalent* abhängige, Entscheidungslogik eingeführt werden, die bei gewollter äquivalenter Rechnung statt *Counter* das Feld *HDPDEquivalentFactor* aufaddierte.

```
// Anzahl der Patienten mit mind. einer Behandlung in der ausgewählten Woche
Count({$<[TreatmentDate.autoCalendar.WeeksAgo]={"$ (= WeeksAgo )"}>} distinct
fkPatientMasterdata)

// Anzahl der Behandlungen in der ausgewählten Woche
Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]={"$ (= WeeksAgo )"}>}
If(HDPDEquivalent = True(), HDPDEquivalentFactor, Counter))
```

Auflistung 5.9: Berechnungen der Patienten- und Behandlungszahlen

Die Zählung der Patienten unterschied sich insofern von der, der Behandlungen, als dass die distinkten IDs der Patienten (*fkPatientMasterdata*) gezählt wurden, um alle Patienten zu zählen, die mindestens eine Behandlung erhalten haben. Die resultierenden Formeln wurden in Auflistung 5.9 bereitgestellt.

Die restlichen fünf KPIs wurden für farblich indikative und prozentuale Anzeigen des vierwöchigen Patienten- und Behandlungszahlen Trends verwendet. Dieser Trend basierte auf der Berechnungslogik des ehemaligen Wochenberichts und stellte den prozentualen

Unterschied der aktuellen Zahlen zu dem jeweiligen vierwöchigen Durchschnitt dar. Es wurden zum einen die Trends der Patienten- und Behandlungszahlen gezeigt und zum anderen die gemittelten Werte dieser, gruppiert nach den Regionen. Die Anzeige der Regionen sollte sich in diesem Fall nur verändern, wenn nach der Behandlungsklasse oder -art gefiltert wird, nicht jedoch nach der Auswahl einer bestimmten Region. Alle Berechnungen für die KPIs wurden in Auflistung 5.10 aufgeführt.

```
// Trend der Patienten
Count({$<[TreatmentDate.autoCalendar.WeeksAgo]='$ (=WeeksAgo) '}>} distinct
      fkPatientMasterdata)
/
(Sum(Aggr(Count({$<[TreatmentDate.autoCalendar.WeeksAgo]=">= $ (=WeeksAgo) <
      $ (=WeeksAgo+4) "}>} distinct fkPatientMasterdata),
      [TreatmentDate.autoCalendar.Week])) / 4) - 1

// Trend der Behandlungen
Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]='$ (=WeeksAgo) '}>}
      If(HDEquivalent = True(), PDHDEquivalentFactor, Counter))
/
(Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]=">= $ (=WeeksAgo) <
      $ (=WeeksAgo+4) "}>} If(HDEquivalent = True(), PDHDEquivalentFactor,
      Counter)) / 4) - 1

// Trend der Regionen
(
  // Trend der Patienten für Region X
  Count({$<[TreatmentDate.autoCalendar.WeeksAgo]='$ (=WeeksAgo) ',
        Region=, LegalEntity=, Location=> * 1<Region='Region X'>} distinct
        fkPatientMasterdata)
  /
  (Sum(Aggr(Count({$<[TreatmentDate.autoCalendar.WeeksAgo]=">=
        $ (=WeeksAgo) < $ (=WeeksAgo+4) ", Region=, LegalEntity=, Location=> *
        1<Region='Region X'>} distinct fkPatientMasterdata),
        [TreatmentDate.autoCalendar.Week])) / 4) - 1
+
  // Trend der Behandlungen für Region X
  Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]='$ (=WeeksAgo) ', Region=,
        LegalEntity=, Location=> * 1<Region='Region X'>} If(HDEquivalent =
        True(), PDHDEquivalentFactor, Counter))
  /
  (Sum({$<[TreatmentDate.autoCalendar.WeeksAgo]=">= $ (=WeeksAgo) <
        $ (=WeeksAgo+4) ", Region=, LegalEntity=, Location=> *
        1<Region='Region X'>} If(HDEquivalent = True(),
        PDHDEquivalentFactor, Counter)) / 4) - 1
) / 2
```

Auflistung 5.10: Berechnungen der KPIs

Entwicklung der Patienten- und Behandlungszahlen

Die grafische Entwicklung der letzten 24 Wochen teilte sich in die Kennzahlen für die aktuelle Entwicklung und die Entwicklung des Vorjahres. Nachfolgend wird ausschließlich die Implementierung der Patientenzahlen erklärt, da sich für die Unterscheidung der Patienten und Behandlungen lediglich die Zählweisen und nicht die in der Berechnung gesetzten Filter änderten.

Das für diese Entwicklung eingesetzte Liniendiagramm nutzte für die Darstellung der zeitlichen Entwicklung das über den *autoCalendar* abgeleitete Feld *TreatmentDate.autoCalendar.Week*, um die ausgewählten Daten in die vergangenen Wochen zu gruppieren. Durch die Verwendung der bereits angesprochenen relativen Wochen konnte die aktuelle Entwicklung mithilfe einer einfachen Berechnung implementiert werden (zu sehen in Auflistung 5.11).

```
// Patienten über 24 Wochen
Count ({ $ < [TreatmentDate.autoCalendar.WeeksAgo] = { ">= $ (= WeeksAgo ) < $ (=
  WeeksAgo + 24 ) " } > } DISTINCT fkPatientMasterdata )
```

Auflistung 5.11: Berechnungen der Patientenzahlen - Aktuelles Jahr

Das Einfügen der Vorjahreswerte gestaltete sich berechnungstechnisch als aufwendiger, da nicht alle Jahre in gleich viele Wochen unterteilt werden. Der prinzipielle Aufbau der zeitlichen Einschränkung ähnelte dem aus Auflistung 5.11, es musste jedoch eine Lösung gefunden werden, um auf die relative Angabe der aktuellen Woche (*WeeksAgo*) die Anzahl der Wochen des Vorjahres zu addieren. Dies war nötig, da ein Jahr entweder 52 oder 53 Wochen haben kann, aber eine ausgewählte Kalenderwoche in diesem Fall immer mit exakt derselben Kalenderwoche des Vorjahres verglichen werden sollte. Die nötigen Berechnungen konnten mithilfe der Qlik eigenen Datumsformeln gelöst und in Auflistung 5.12 nachvollzogen werden. Ein korrektes Anzeigen der Vorjahreswochen war somit für die Patienten möglich und konnte parallel für die Behandlungen übertragen werden (Punkt 1 und Punkt 3 der funktionalen Anforderungen).

Um beide Diagramme zu vereinen und über auswählbare Reiter zugänglich zu machen, wurde eine sogenannte Sammelbox eingesetzt. Die resultierende Visualisierung kann in Abbildung 5.9 betrachtet werden.

5 Praktische Umsetzung

```
// Patienten Vorjahr über 24 Wochen
// ">= $(= WeeksAgo + [Wochen d. Vorjahres] ) < $(= WeeksAgo + [Wochen d.
  Vorjahres] + 24 + [1 Wenn Vorvorjahr 53 Wochen hatte])"

If(ShowPriorYear = True(),
Count({
  $<[TreatmentDate.autoCalendar.WeeksAgo]={">= $(= WeeksAgo +
    If(WeekYear(Today()-WeeksAgo*7)-1 =
    WeekYear(MakeWeekDate(WeekYear(Today()-WeeksAgo*7)-1, 53, 3)), 53,
    52) ) < $(= WeeksAgo + If(WeekYear(Today()-WeeksAgo*7)-1 =
    WeekYear(MakeWeekDate(WeekYear(Today()-WeeksAgo*7)-1, 53, 3)), 53,
    52) + 24 + If(WeekYear(Today()-WeeksAgo*7)-2 =
    WeekYear(MakeWeekDate(WeekYear(Today()-WeeksAgo*7)-2, 53, 3)), 1, 0)
  }">
}) distinct fkPatientMasterdata)
)
```

Auflistung 5.12: Berechnungen der Patientenzahlen - Vorjahr

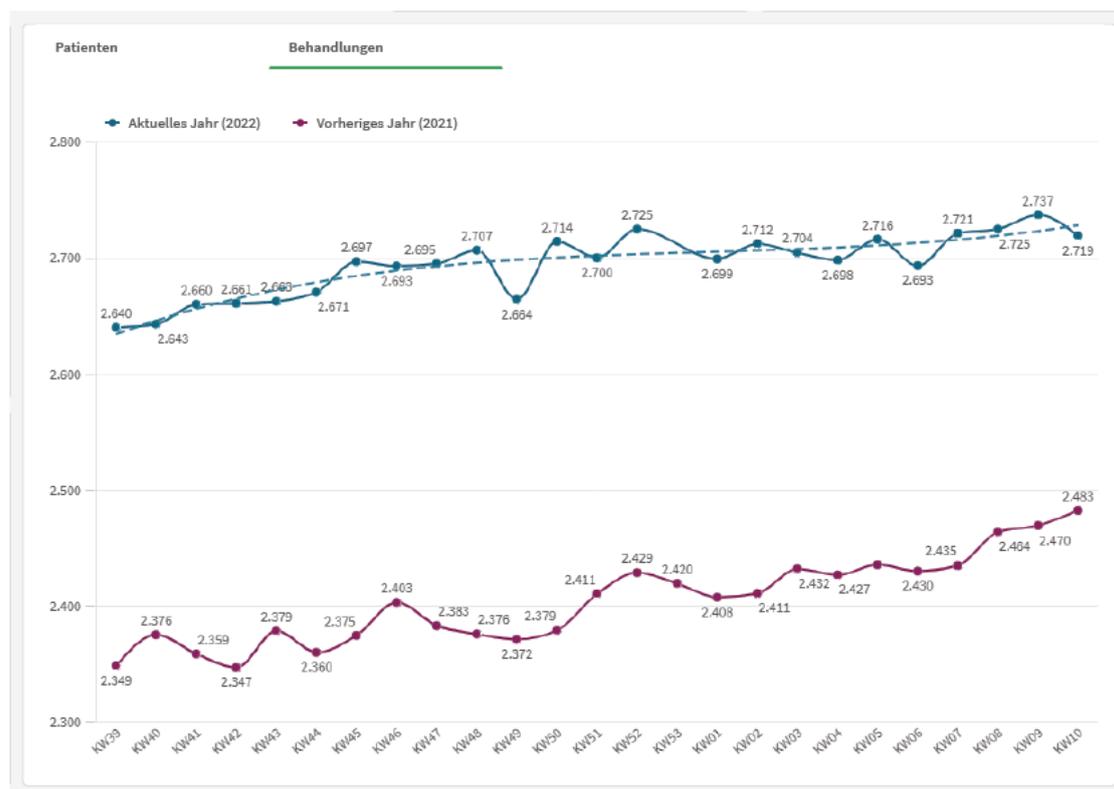


Abbildung 5.9: Grafische Entwicklung der Patienten- und Behandlungszahlen

Entwicklung der Behandlungen pro Patient

Die Erkundung der Domäne in Abschnitt 5.1 hatte ergeben, dass sich je nach Behandlungsklasse die optimale Anzahl der Behandlungen pro Woche unterscheiden. Bei einer Abweichung des tatsächlichen Verhältnisses vom jeweiligen Richtwert konnte deshalb von geplanten oder ungeplanten Besonderheiten, wie zum Beispiel vermehrten Krankenhausaufenthalten oder Urlauben ausgegangen werden.

Um eine grafische Übersicht der durchschnittlichen Behandlungen pro Patient pro Woche zu implementieren, mussten die Berechnungen aus Unterunterabschnitt 5.7.4, getrennt nach dem aktuellen und dem vorherigen Jahr, ins Verhältnis gesetzt und angezeigt werden. Die Konfiguration des eingesetzten Liniendiagramms unterschied sich dabei nicht nennenswert von der des zuvor entwickelten Patienten- und Behandlungszahlen Diagramms und konnte weitestgehend übernommen werden.

Nachstellung des ehemaligen Wochenberichts

Durch die vorherige Dokumentation des ehemaligen Wochenberichts und der schon erfolgten Entwicklung des Dashboards, konnten die Kennzahlen schnell implementiert werden. Die resultierende Visualisierung bestand aus einer Pivot-Tabelle, welche die Gruppierungen der Standorte als Reihen anzeigte und die Durchschnitts- und Vergleichswerte als Zeilen (zu sehen in Abbildung 5.10).

Region Q		Zentrum Q		Werte							
		KW10 vs. 4 Wochen Ø	4 vs. 12 Wochen Ø	12 vs. 24 Wochen Ø	24 vs. 52 Wochen Ø	KW10	4 Wochen Ø	12 Wochen Ø	24 Wochen Ø	52 Wochen Ø	KW10 2021 vs. 2022
Gesamt		0,00%	0,48%	0,62%	2,55%	992	992,00	987,25	981,17	956,81	9,13%
Region A	Zentrum A1	-1,91%	-1,36%	-2,00%	-1,17%	77	78,50	79,58	81,21	82,17	-6,10%
	Zentrum A2	0,31%	1,87%	1,42%	4,69%	82	81,75	80,25	79,13	75,58	17,14%
	Zentrum A3	-0,91%	-1,68%	-0,79%	3,36%	82	82,75	84,17	84,83	82,08	9,33%
	Zentrum A4	0,29%	1,29%	0,65%	1,75%	85	84,75	83,67	83,13	81,69	7,59%
Region B	Zentrum B1	0,30%	0,30%	0,81%	3,76%	83	82,75	82,50	81,83	78,87	9,21%
	Zentrum B2	-0,61%	2,27%	2,00%	2,42%	82	82,50	80,67	79,08	77,21	10,81%
	Zentrum B3	2,13%	1,02%	-0,71%	-0,19%	84	82,25	81,42	82,00	82,15	2,44%
	Zentrum B4	0,00%	1,08%	0,49%	3,49%	86	86,00	85,08	84,67	81,81	8,86%
Region C	Zentrum C1	0,59%	1,81%	3,00%	3,90%	85	84,50	83,00	80,58	77,56	19,72%
	Zentrum C2	0,00%	0,10%	1,04%	3,51%	85	85,00	84,92	84,04	81,19	14,86%
	Zentrum C3	-0,62%	-0,41%	0,83%	2,88%	80	80,50	80,83	80,17	77,92	9,59%

Abbildung 5.10: Nachstellung des ehemaligen Wochenberichts

Es wurden wie bereits bei der Implementierung der grafischen Entwicklung zwei prinzipiell gleiche Tabellen aufgebaut, welche die Patienten und Behandlungen getrennt voneinander in einer Sammelbox zugänglich machten. Die Durchschnittswerte wurden bei den Patienten mithilfe einer zusätzlichen zeitlichen Aggregation und anschließenden Teilung durch die Anzahl der Wochen für die jeweiligen Zeiträume berechnet und konnte bereits in Auflistung 5.10 nachvollzogen werden. Da für die Zählung der Behandlungen lediglich Summen des *Counter* bzw. *PDHDEquivalentFactor* Feldes gebildet werden mussten, konnte dieser zusätzliche Aggregationsschritt hierbei weggelassen werden. Für die Berechnungen der Vergleichswerte wurden ebenfalls die prozentualen Abweichungen der zu vergleichenden Kennzahlen gebildet, wobei ein Beispiel in Auflistung 5.13 zu sehen ist.

```
(Sum(Aggr(Count({$<[TreatmentDate.autoCalendar.WeeksAgo]={">= $(= WeeksAgo )
< $(= WeeksAgo+4 )"}>} distinct fkPatientMasterdata), Region,
LegalEntity, Location, [TreatmentDate.autoCalendar.Week])) / 4)
/
(Sum(Aggr(Count({$<[TreatmentDate.autoCalendar.WeeksAgo]={">= $(= WeeksAgo )
< $(= WeeksAgo+12 )"}>} distinct fkPatientMasterdata), Region,
LegalEntity, Location, [TreatmentDate.autoCalendar.Week])) / 12) - 1
```

Auflistung 5.13: Berechnungen des 4 vs. 12 Wochen Trends der Patienten

Ergänzt wurden diese Kennzahlen durch einen Vergleich der aktuell ausgewählten Woche zu der Vorjahreswoche. Hier wurden ebenfalls die Vorjahreswerte und die prozentualen Vergleiche als Spalten in die Tabelle eingefügt. Die Berechnung der Vorjahreswerte wurde in diesem Fall anders als in Auflistung 5.12 umgesetzt, da keine Notwendigkeit für einen Vorjahreszeitraum bestand. Alternativ wurde die *WeeksAgo* Variable in die aktuell ausgewählte Woche konvertiert und zusätzlich zu dem daraus kalkulierbaren Vorjahr als Filter für die Berechnung genommen. Die resultierende Formel wird in Auflistung 5.14 aufgeführt.

```
Count({$<
[TreatmentDate.autoCalendar.Week]={"$ (=Num(Week(Today())-WeeksAgo*7),'00')"},
[TreatmentDate.autoCalendar.WeekYear]={"$ (=WeekYear(Today())-WeeksAgo*7)-1"}
>} distinct fkPatientMasterdata)
```

Auflistung 5.14: Berechnungen der Patienten der Vorjahreswoche

Eine weitere Besonderheit des ehemaligen Wochenberichts lag in der farblichen Markierung der prozentualen Vergleiche. Bei einer negativen prozentualen Entwicklung wurde der Trend rot eingefärbt und bei einer positiven grün. Um dies im Dashboard umzuset-

zen, wurden die originalen Farben für die Hintergründe und den Text übernommen. Für jede farbige Spalte in der Tabelle musste die entsprechende Berechnung der Kennzahl kopiert und in die Formel der Hintergrund- und Textfarbe eingefügt werden.

Entwicklung der Patienten als Wasserfalldiagramm

Da jederzeit zwischen den Patienten und Behandlungszahlen der grafischen Entwicklung gewechselt werden konnte, jedoch keine simultane Ansicht beider Kennzahlen im Vergleich möglich war, sollte abschließend ein für die Patientenentwicklung exklusives Wasserfalldiagramm eingeführt werden. Dieses sollte aufgrund des alleinigen Fokus ohne Vorjahresvergleich auskommen, wobei die Limitierungen des Diagrammtyps dies ebenfalls nicht zuließen. Die Umsetzung erforderte demzufolge lediglich die gesonderte Konfiguration aller angezeigten 24 Wochen inklusive der visuellen Indikatoren.

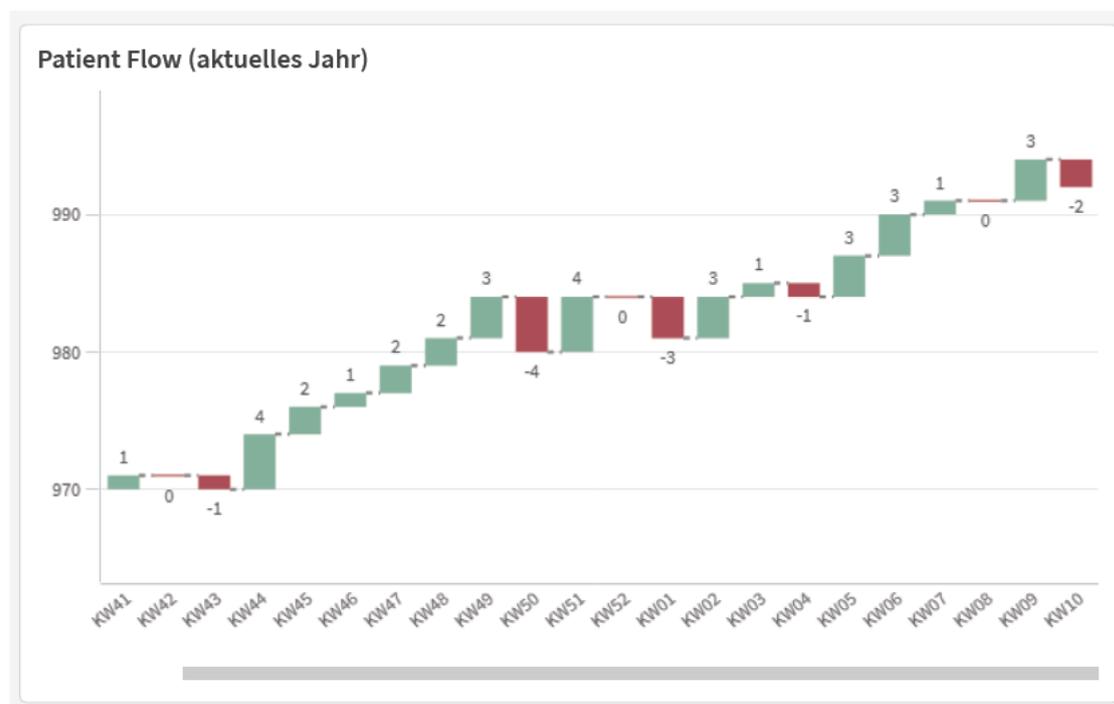


Abbildung 5.11: Patientenentwicklung als Wasserfalldiagramm

5.8 Testen der Auswertung

Um zu überprüfen, ob die Entwicklung des Dashboards abgeschlossen werden konnte, wurden die Anforderungen aus Abschnitt 5.2 auf ihre Erfüllung hin beurteilt. Da auch dieser Schritt wiederkehrend, während der Entwicklung stattfand, werden die Ergebnisse des Prüfprozesses hier zusammengefasst und ungeachtet der tatsächlichen Implementierungsreihenfolge vorgestellt. Dabei werden zuerst die funktionalen Anforderungen betrachtet, um im Anschluss die Erfüllung der nichtfunktionalen Anforderungen zu begutachten.

5.8.1 Funktionale Anforderungen

Punkt 1 der funktionalen Anforderungen bestand aus der Unterscheidung von Patienten- und Behandlungszahlen. Im Dashboard wurden dafür entsprechende Reiter implementiert, die ein Umschalten der entsprechenden Visualisierungen ermöglichten. Die KPIs hingegen zeigten sowohl die Behandlungszahlen als auch die Patientenzahlen getrennt voneinander. Zusätzlich dazu wurde eine kombinierte Visualisierung der durchschnittlichen Behandlungen pro Patient entwickelt.

Punkt 2 der funktionalen Anforderungen wurde ebenfalls erfüllt. Es wurde eine grafische Entwicklung der letzten 24 Wochen eingeführt, die sowohl die Patienten- als auch die Behandlungszahlen mithilfe eines beschrifteten Liniendiagramms darstellen konnte.

Der grafische Vergleich zum Vorjahr (Punkt 3) wurde als zusätzliche Datenreihe innerhalb der grafischen Entwicklung und der kombinierten Visualisierung zugänglich gemacht. Es war außerdem möglich, über eine Schaltfläche die Vorjahreswerte ein- oder auszublenden.

Obwohl Qlik eine generelle Möglichkeit bot nach allen Feldern der geladenen Daten zu filtern, konnten ergänzend dazu die Regionen und Zentren über die Nachstellung des ehemaligen Wochenberichts ausgewählt werden (Punkt 4). Aufgrund der niedrigen Priorität dieser Auswahlmöglichkeit wurde auf ein zusätzliches Filterpaneel verzichtet.

Die Forderung nach Filtermöglichkeiten für die unterschiedlichen Behandlungsarten und -klassen aus Punkt 5 konnten am linken Rand des Dashboards untergebracht werden.

Mithilfe einer weiteren Schaltfläche wurde dem Anwender außerdem ermöglicht, in allen Berechnungen der Behandlungszahlen HD-äquivalente PD Zahlen zu verwenden. Punkt 6 der funktionalen Anforderungen konnte als Folge dessen ebenfalls erfüllt werden.

Zuletzt wurde die Auswahl der aktuellen Woche mithilfe zwei weiterer Schaltflächen implementiert, die es dem Anwender ermöglichten sowohl vergangene als auch aktuelle Wochen auszuwählen (Punkt 7). Dabei berücksichtigten alle Visualisierungen die ausgewählte Woche und aktualisierten sich automatisch bei einer Änderung.

5.8.2 Nichtfunktionale Anforderungen

Der erste Punkt der nichtfunktionalen Anforderungen bestand aus der Anpassbarkeit und Personalisierbarkeit der Auswertung. Durch den Einsatz von Qlik als Auswertungssoftware wurde bereits eine Grundfunktionalität geschaffen, welche es dem Anwender ermöglichte nach Feldern zu filtern, ohne dass dies weiterer Konfiguration erforderte. Zusätzlich dazu wurden Filterpaneele hinzugefügt, die die wichtigsten Behandlungsparameter einschränken konnten und Schaltflächen implementiert, um weiterführende Funktionen des Dashboards zu steuern.

Die geforderte intuitive Bedienbarkeit aus Punkt 2 konnte ebenfalls gewährleistet werden. Zur Benutzung des Dashboards war keine anwenderseitige Ersteinrichtung nötig, da beim Start automatisch die Vorwoche ausgewählt und alle Visualisierungen entsprechend angepasst wurden. Außerdem wurde durch die hierarchische Platzierung der einzelnen Komponenten, die farblich markierten Kennzahlen und die Nachstellung des bereits bekannten Wochenberichts, ein natürlicher Wiedererkennungswert geschaffen. Das UX-Design wurde zudem mithilfe des bereits im Rahmen der Anforderungsanalyse befragten leitenden Arztes analysiert und getestet, wobei das sehr positive Feedback die gelungene Umsetzung des Dashboards bestätigten.

Punkt 3 der nichtfunktionalen Anforderungen konnte hingegen nicht implementiert werden. Das automatische Aktualisieren der Daten hätte mithilfe der Qlik Management Console (QMC) umgesetzt werden können, wobei IT-seitige Besonderheiten im Unternehmen ein Anpassen der QMC-Konfigurationen verhinderten. Ein nachträgliches Einrichten der automatischen Aktualisierung wäre jedoch mit wenigen Schritten durchführbar. Als Alternative wurde eine außerplanmäßige Schaltfläche eingefügt, welche dem Anwender ein manuelles Aktualisieren der Daten ermöglichte.

Zuletzt musste die inhaltliche Schlüssigkeit der Daten geprüft werden (Punkt 4). Obwohl auch alle bisherigen Anforderungen schon während der Entwicklung wiederholt geprüft wurden, so war dieser Punkt bereits für die Aufbereitung der Daten entscheidend. Die

Behandlungszahlen wurden, wie bereits in Abschnitt 5.6 beschrieben, mit den ehemaligen Wochenberichten abgeglichen und Unklarheiten wurden über stichprobenartige Tests plausibilisiert. Mithilfe von monatlichen Patientenauswertungen aus den Managementsystemen der Zentren konnten zusätzlich dazu die genauen Patientenzahlen überprüft und Abweichungen verfolgt werden, sodass auch die inhaltliche Schlüssigkeit des Dashboards bestätigt werden konnte.

5.9 Abschluss und interne Veröffentlichung

Sobald alle Anforderungen erfüllt und die angestrebten Funktionen erreicht waren, konnte die Entwicklung beendet werden. Es wurde überprüft, ob sowohl die SQL Abfragen, als auch die Qlik-Berechnungen ausreichend formatiert und kommentiert waren, um bei zukünftigen Anpassungen oder Erweiterungen ein schnelles Verständnis für die angewandten Logiken zu erreichen. Abschließend wurden jegliche Dokumentationen finalisiert und alle im Laufe der Entwicklung erstellten Dokumente gebündelt und abgelegt.

Das Dashboard konnte zum Abschluss mithilfe weniger Schritte veröffentlicht werden und war somit von allen angelegten internen Nutzern abrufbar. Da es jedoch unternehmensseitige Probleme mit der Einrichtung der erforderlichen Benutzer gab, konnte das Dashboard den alten Wochenbericht noch nicht vollständig ersetzen.

6 Fazit und Ausblick

Die vorliegende Arbeit hatte zum Ziel, ein Vorgehensmodell zu entwerfen und dessen Erfolg anhand einer praktischen Anwendung zu beurteilen.

Zu diesem Zweck wurde sich mit den Grundlagen der Business Intelligence befasst und auf Basis von verschiedenen Vorgehensmodellen MOROP-BI entwickelt: Ein Vorgehensmodell, welches dazu anleitet sollte veraltete BI Prozesse zu analysieren, benötigte Daten bei Bedarf aufzubereiten und abschließend eine neue, optimierte Auswertung zu entwickeln. Die Anwendung des Modells erfolgte im Zuge der Neuentwicklung eines wöchentlichen Behandlungsberichts in einem nephrologischen Unternehmen.

Mithilfe der in Abschnitt 4.1 aufgestellten Anforderungen an das MOROP-BI Vorgehensmodell konnte der Entwicklungsprozess bereits aus theoretischer Sicht validiert werden. Die erfüllten Anforderungen wurden während der Entwicklung des Vorgehensmodells (Abschnitt 4.3) angesprochen und in einem positiven Zwischenfazit (Abschnitt 4.4) zusammengefasst.

Ergänzend zu der Beurteilung der funktionalen Anforderungen aus dem Zwischenfazit lässt sich nach der praktischen Implementierung folgendes festhalten:

- Die Langlebigkeit und Stabilität der Auswertung kann aufgrund der Länge des Einsatzes noch nicht weiter bewertet werden, bisher waren jedoch keine ungeplanten Vorkommnisse zu verzeichnen.
- Die Qualität der verwendeten Daten wurde durch den Einsatz des Vorgehensmodells stark verbessert und führte zu einem umfangreichen und fehlerfreien Datenmodell, welches mittlerweile auch von anderen Auswertungen im Unternehmen genutzt wird.
- Das erstellte Dashboard konnte durch die iterativen Entwicklungs- und Teststufen fehlerfrei und mit erweitertem Funktionsumfang erstellt werden.

Wie bereits in Abschnitt 5.8 ausführlich dargelegt, verlief auch die praktische Anwendung bis auf zwei Ausnahmen sehr erfolgreich. Zum einen konnte aus administratorischen Gründen keine automatische Aktualisierung der Daten konfiguriert werden und zum anderen gab es interne Probleme mit dem Einrichten der erforderlichen Benutzer, sodass das Dashboard noch nicht vollständig einsetzbar war.

Eine kritische Betrachtung von MOROP-BI zeigt dennoch einige mögliche Verbesserungen, welche nachfolgend weiter erläutert werden.

- Aufgrund der internen administratorischen Schwierigkeiten mit dem Anlegen der Benutzer, sollte das Einrichten und die Konfiguration neuer Software als fester Bestandteil in das Modell aufgenommen werden. Dieser Punkt wurde in der Konzeption des Modells vernachlässigt, stellt sich jedoch als eine erforderliche Maßnahme auf technischer Ebene heraus.
- Die Dokumentation der Prozesse erwies sich in vielen Fällen als schwierig, da eine geeignete Granularität gefunden werden musste. Die vorgeschlagene Wissenssammlung könnte allerdings eine geeignete Plattform bieten, um die Erstellung von begleitenden Dokumenten zu vereinfachen und diese an einem Ort zu sammeln und zugänglich zu machen.
- Früh definierte Anforderungen können sich durch neue Erkenntnisse aus der Analyse der veralteten Auswertung stark verändern. Der iterative Charakter des Modells muss dementsprechend auch bei den anfänglichen Schritten klar kommuniziert und gelebt werden.
- Um die Erfüllung der Anforderungen zu vereinfachen, könnten im Vorfeld der Entwicklung Testszenarien entwickelt werden, die als standardisierte Methode zur Überprüfung der Auswertung dienen würden.

Abschließend soll ein Ausblick auf weitere mögliche Entwicklungen des Modells gegeben werden.

Das bereits in Abschnitt 2.2 angesprochene BI Reifegradmodell der Gartner Analysten Howson und Duncan ermöglicht eine Einordnung des Business Intelligence Entwicklungsstandes eines Unternehmens auf einer fünfstufigen Skala. Die Entstehung von veralteten BI Auswertungen wurde in diesem Modell sowohl in der ersten als auch in der zweiten Phase eingeordnet, da dort unter anderem noch keine standardisierten Datenquellen oder Prozesse existieren.

Das Vorgehensmodell setzt genau an diesen Problemen an und schafft mithilfe eines klar definierten Prozesses erste konsistente Datenquellen und dokumentierte und beständige Auswertungen innerhalb eines Unternehmens.

MOROP-BI kann somit bei einer strukturierten und umfassenden Anwendung die Probleme der ersten und zweiten Stufe des BI Reifegradmodells mindern und einem Unternehmen eine Überleitung in die dritte Entwicklungsstufe ermöglichen.

Eine Adaption des Modells auf die finalen Entwicklungsstufen wäre zusätzlich denkbar, sodass lediglich ein Vorgehensmodell auf alle BI Entwicklungsprozesse angewendet werden kann.

Zusammenfassend lässt sich sagen, dass die Entwicklung und der Einsatz des Vorgehensmodells trotz möglicher Verbesserungspotenziale als gesamtheitlicher Erfolg angesehen werden kann. Eine Übertragung des Vorgehensmodells auf andere Unternehmen und Branchen wäre aufgrund dieser Ergebnisse durchaus möglich.

Literatur

- Baars, Henning, Bodo Rieger, Ralf Finger, Peter Gluchowski, Andreas Hilbert und Hans-Georg Kemper (2011). „Umbenennung der GI-Fachgruppe "Management Support Systems"(MSS) in "Business Intelligence"(BI)“. In: *Positionspapier des Leitungsgremiums der GI-Fachgruppe Management Support Systems*. URL: <https://fg-wi-bia.gi.de/organisatorisches/benennung-der-fachgruppe> (besucht am 05.04.2022).
- Bange, Carsten (2016). „Werkzeuge für analytische Informationssysteme“. In: *Analytische Informationssysteme: Business Intelligence-Technologien und -Anwendungen*. Hrsg. von Peter Gluchowski und Peter Chamoni. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 97–126. ISBN: 978-3-662-47763-2. DOI: 10.1007/978-3-662-47763-2_6.
- Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer und Rüdiger Wirth (Aug. 2000). *CRISP-DM 1.0: Step-by-step data mining guide*.
- European Union (1998). *CRISP-DM Project*. URL: <https://cordis.europa.eu/project/id/25959> (besucht am 05.04.2022).
- Fayyad, Usama M., Gregory Piatetsky-Shapiro und Padhraic Smyth (1996a). „From Data Mining to Knowledge Discovery in Databases“. In: *AI Magazine* 17.3, S. 37–54. DOI: 10.1609/aimag.v17i3.1230.
- (1996b). „From Data Mining to Knowledge Discovery: An Overview“. In: *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, S. 1–34. ISBN: 978-0-262-56097-9.
- Gluchowski, Peter, Roland Gabriel und Carsten Dittmar (2008). „Business Intelligence“. In: *Management Support Systeme und Business Intelligence: Computergestützte Informationssysteme für Fach- und Führungskräfte*. Springer Berlin Heidelberg, S. 89–116. ISBN: 978-3-540-68269-1. DOI: 10.1007/978-3-540-68269-1.

- Howson, Cindi und Alan D. Duncan (2015). *BI and Analytics Maturity Model*. Gartner Inc. URL: <https://www.gartner.com/en/documents/3136418/itscore-overview-for-bi-and-analytics> (besucht am 05.04.2022).
- IBM (2021). *IBM SPSS Modeler CRISP-DM Guide*. URL: https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=SS3RA7_sub/modeler_crispdm_ddita/modeler_crispdm_ddita-gentopic1.html (besucht am 05.04.2022).
- Kemper, Hans-Georg, Walid Mehanna und Carsten Unger (2006). „Entwicklung integrierter BI-Anwendungssysteme“. In: *Business Intelligence - Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung*. 2. Aufl. Wiesbaden: Vieweg Verlag, S. 139–173. ISBN: 978-3-8348-9056-6. DOI: 10.1007/978-3-8348-9056-6_4.
- Klinge, Matthias und Doreen Brodmann, Hrsg. (2017). *Einführung in die Nephrologie und Nierenersatzverfahren: Für Pflegende, Medizinstudenten und Assistenzärzte*. Springer. ISBN: 978-3-662-54583-6. DOI: 10.1007/978-3-662-54583-6.
- Louridas, Panagiotis (2006). „Using wikis in software development“. In: *IEEE Software* 23.2, S. 88–91. DOI: 10.1109/MS.2006.62.
- Luhn, Hans Peter (1958). „A Business Intelligence System“. In: *IBM Journal of Research and Development* 2.4, S. 314–319. DOI: 10.1147/rd.24.0314.
- Müller, Roland M. und Hans-Joachim Lenz (2010). *Business Intelligence*. Springer. ISBN: 978-3-642-35559-2. DOI: 10.1007/978-3-642-35560-8.
- Nascimento, Gleison S., Cirano Iochpe, Lucinéia H. Thom und Manfred Reichert (2009). „A Method for Rewriting Legacy Systems using Business Process Management Technology“. In: *ICEIS 2009 - Proceedings of the 11th International Conference on Enterprise Information Systems*. Bd. 1, S. 57–62. ISBN: 978-989-8111-86-9. DOI: 10.5220/0001950400570062.
- QlikTech International AB (2022). *Best Practices für das Gestalten von Visualisierungen*. URL: https://help.qlik.com/de-DE/cloud-services/Subsystems/Hub/Content/Sense_Hub/Visualizations/create-viz-best-practices.htm (besucht am 05.04.2022).
- Totok, Andreas (2016). „Von der Business-Intelligence-Strategie zum Business Intelligence Competency Center“. In: *Analytische Informationssysteme: Business Intelligence-Technologien und -Anwendungen*. Hrsg. von Peter Gluchowski und Peter Chamoni. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 33–53. ISBN: 978-3-662-47763-2. DOI: 10.1007/978-3-662-47763-2_3.

A Auszug aus einem Wochenbericht

Hämo- / LDL / IPD-Zahlen pro Woche seit 22.02.2021 (52 Wochen)					20.02.2022
	Gesamt-HD			LDL/Plasma	nicht abg.
22.02.2021	49	24	25	7	0
01.03.2021	44	27	17	6	0
08.03.2021	40	21	19	6	0
15.03.2021	43	20	23	7	0
22.03.2021	44	22	22	7	0
29.03.2021	46	25	21	7	0
05.04.2021	47	27	20	7	0
12.04.2021	42	25	17	7	0
19.04.2021	43	27	16	7	0
26.04.2021	40	28	12	7	0
03.05.2021	43	26	17	6	0
10.05.2021	48	29	19	7	0
17.05.2021	42	22	20	7	0
24.05.2021	41	19	22	7	0
31.05.2021	44	28	16	7	0
07.06.2021	45	24	21	6	0
14.06.2021	44	21	23	7	0
21.06.2021	45	20	25	7	0
28.06.2021	45	24	21	7	0
05.07.2021	45	28	17	7	0
12.07.2021	48	22	26	6	0
19.07.2021	48	27	21	6	0
26.07.2021	42	28	14	6	0
02.08.2021	43	21	22	6	0
09.08.2021	38	24	14	7	0
16.08.2021	36	23	13	7	0
23.08.2021	43	26	17	7	0
30.08.2021	45	20	25	6	0
06.09.2021	40	24	16	6	0
13.09.2021	39	24	15	6	0
20.09.2021	38	27	11	6	0
27.09.2021	37	27	10	7	0
04.10.2021	39	26	13	6	0
11.10.2021	41	26	15	7	0
18.10.2021	41	20	21	7	0
25.10.2021	47	27	20	7	0
01.11.2021	47	28	19	6	0
08.11.2021	41	25	16	7	0
15.11.2021	46	29	17	7	0
22.11.2021	46	20	26	7	0
29.11.2021	47	24	23	7	0
06.12.2021	45	28	17	7	0
13.12.2021	43	28	15	7	0
20.12.2021	44	29	15	7	0
27.12.2021	45	17	28	6	1
03.01.2022	45	28	17	7	0
10.01.2022	45	26	19	6	0
17.01.2022	40	28	12	7	0
24.01.2022	46	20	26	7	0
31.01.2022	45	24	21	7	0
07.02.2022	46	21	25	7	0
14.02.2022	46	26	20	7	0

A Auszug aus einem Wochenbericht

PD-Zahlen pro Woche seit 22.02.2021 (52 Wochen)					20.02.2022
	Gesamt	APD	CAPD	IPD	Seite 1 von 1
22.02.2021	38	14	21	3	
01.03.2021	38	14	21	3	
08.03.2021	38	14	21	3	
15.03.2021	38	14	21	3	
22.03.2021	38	14	21	3	
29.03.2021	38	14	21	3	
05.04.2021	38	14	21	3	
12.04.2021	38	14	21	3	
19.04.2021	38	14	21	3	
26.04.2021	38	14	21	3	
07.05.2021	38	14	21	3	
10.05.2021	38	14	21	3	
17.05.2021	38	14	21	3	
24.05.2021	38	14	21	3	
31.05.2021	38	14	21	3	
07.06.2021	38	14	21	3	
15.06.2021	38	14	21	3	
21.06.2021	38	14	21	3	
01.07.2021	38	14	21	3	
05.07.2021	38	14	21	3	
12.07.2021	38	14	21	3	
19.07.2021	38	14	21	3	
26.07.2021	45	14	28	3	
02.08.2021	45	14	28	3	
09.08.2021	45	14	28	3	
16.08.2021	45	14	28	3	
23.08.2021	45	14	28	3	
30.08.2021	45	14	28	3	
06.09.2021	45	14	28	3	
13.09.2021	45	14	28	3	
20.09.2021	45	14	28	3	
27.09.2021	45	14	28	3	
04.10.2021	45	14	28	3	
11.10.2021	45	14	28	3	
18.10.2021	45	14	28	3	
25.10.2021	45	14	28	3	
01.11.2021	45	14	28	3	
08.11.2021	45	14	28	3	
15.11.2021	45	14	28	3	
22.11.2021	45	14	28	3	
29.11.2021	45	14	28	3	
06.12.2021	45	14	28	3	
13.12.2021	45	14	28	3	
20.12.2021	45	14	28	3	
27.12.2021	45	14	28	3	
03.01.2022	45	14	28	3	
10.01.2022	45	14	28	3	
17.01.2022	45	14	28	3	
24.01.2022	45	14	28	3	
31.01.2022	45	14	28	3	
07.02.2022	45	14	28	3	
14.02.2022	45	14	28	3	

B Auszug aus den wöchentlichen Behandlungszahlen

Wöchentliche Behandlungszahlen

Gesellschaften
07.03.2022 - 13.03.2022
KW 07

Standorte	Entwicklung				aktuelle Woche 14.02.2022 - 20.02.2022	Durchschnittswerte			Vorjahr 2021		Entwicklung Behandlungen der letzten 12 Wochen			
	aktuelle Woche vs. letzten 4 Woche (%)	letzten 4 Woche vs. 12 Wochen (%)	letzten 12 Woche vs. 24 Wochen (%)	letzten 24 Woche vs. 52 Wochen (%)		4 Woche (Ø)	12 Woche (Ø)	24 Woche (Ø)	52 Woche (Ø)	KW 07 2021 vs. 2022		4 Wochen 2021 vs. 2022 (%)		
Zentrum A1	0,4%	1,1%	-1,5%	1,5%	396	394	390	396	390	-2,7%	407	-7,0%	424	
Zentrum A2	7,0%	1,3%	20,4%	-9,3%	179	167	165	137	151	-1,6%	182	-9,9%	186	
Zentrum A3	-2,3%	3,4%	-4,0%	12,7%	146	150	145	151	134	3,5%	141	-2,9%	154	
Zentrum A4	-1,5%	2,5%	-0,5%	6,0%	374	380	370	373	352	-2,6%	384	-1,9%	387	
Zentrum B1	0,0%	1,9%	5,3%	-0,7%	322	322	316	300	302	-2,1%	329	3,2%	312	
Zentrum B2	2,9%	-1,9%	14,9%	-5,2%	234	228	232	202	213	11,4%	210	1,7%	224	
Zentrum B3	-2,9%	2,4%	0,0%	11,4%	136	140	137	137	123	-9,3%	150	3,1%	136	
Zentrum B4	-3,4%	-3,1%	-3,8%	7,1%	191	198	204	212	198	-10,7%	214	-9,1%	218	
Zentrum C1	-3,7%	0,6%	-1,3%	2,0%	527	547	543	550	539	-8,1%	573	-4,1%	570	
Zentrum C2	-8,2%	0,1%	-0,3%	5,9%	280	305	305	306	289	3,3%	271	5,1%	290	
Zentrum C3	-4,2%	0,4%	3,8%	-1,1%	623	650	648	624	631	-9,2%	686	-1,3%	659	
Zentrum C4	2,7%	2,0%	0,5%	0,0%	415	404	396	394	394	-5,3%	438	-9,4%	446	
Gesamt	-1,6%	0,9%	1,8%	1,8%	3.823	3.884	3.851	3.782	3.716	-4,1%	3.985	-3,0%	4.005	

C Erklärung der Datenbankattribute

Tabellenname	Attribut	Erklärung
Treatment	idPatientTreatment	eindeutige ID der Behandlung
	fkPatientMasterdata	eindeutige ID des Patienten
	TreatmentTypeGroup	Behandlungsart (HD, PD, AN)
	TreatmentTypeShortCut	Behandlungsart (IPD, APD, CAPD)
	DialysisType	Zusatzinformationen zur Behandlung
	PatientCode	Zusatzinformationen zur Behandlung
	DialysisShort	Zusatzinformationen zur Behandlung
	TreatmentDate	Datum der Behandlung
	SiteNumber	BSNR des Zentrums
	IsCompleted	Behandlung wurde abgeschlossen
Treatment-BillingItem	fkPatientTreatment	eindeutige ID der Behandlung
	SiteNumber	BSNR des Zentrums
Treatment-Class	fkPatientMasterdata	eindeutige ID des Patienten
	Begin	Beginn der Behandlungsklasse
	End	Ende der Behandlungsklasse
	ShortCut	Behandlungsklasse im Zeitraum
SiteNumbers	SiteNumber	BSNR des Zentrums
	Region	zugehörige Region
	LegalEntity	zugehöriges Unternehmen
	Location	Name des Zentrums
	ClinicFk	FK zu Patient.ClinicId
Patient	idPatientMasterdata	eindeutige ID des Patienten
	ClinicId	zuerst besuchtes Unternehmen

D Finales Dashboard

Wochenbericht

KW10

< [] >

Letzte Behandlung: 01.05.2022

Q Daten aktualisieren

HB-Aquivalenz

* Verschiebepfeil

Patienten in KW10

992

Trend der Patienten

↑ +0,00%

Behandlungen in KW10

2.719

Trend der Behandlungen

↓ -0,24%

Q Behandlungsgruppe

HD

IPD

APD

CAPD

AN

Q Behandlungsklasse

Akut

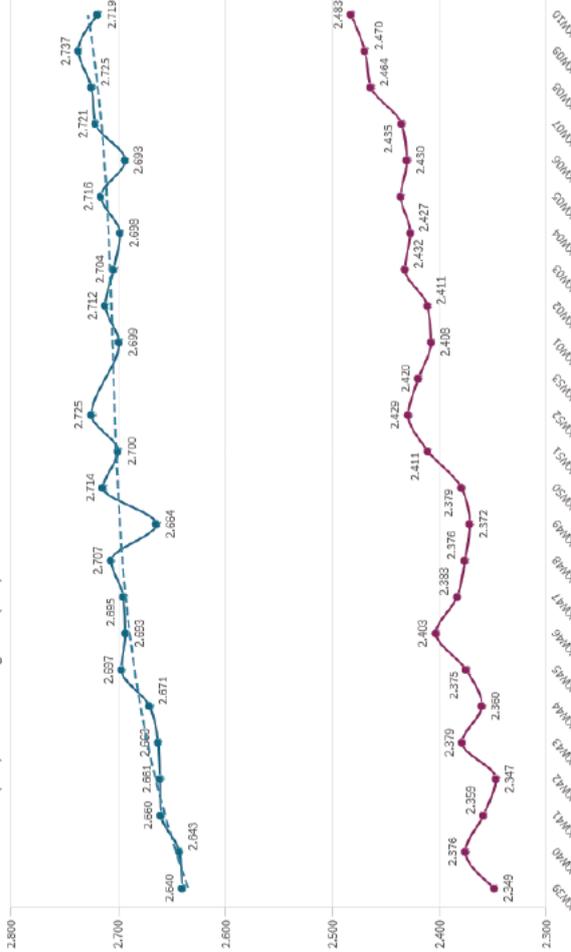
Chronisch

Gast

Patienten

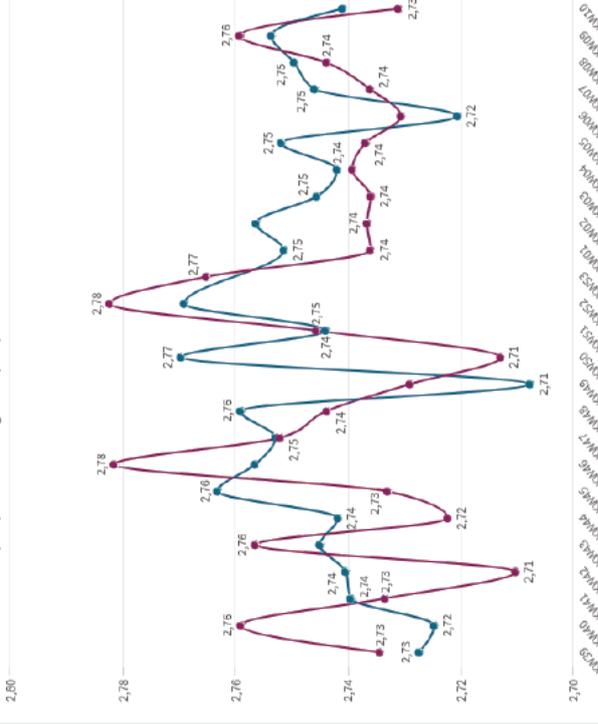
Behandlungen

● Aktuelles Jahr (2022) ● Vorheriges Jahr (2021)

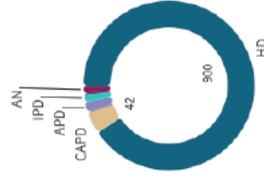


Behandlungen pro Patient

● Aktuelles Jahr (2022) ● Vorheriges Jahr (2021)



Behandlungsgruppe



Patienten KW10

Behandlungen KW10

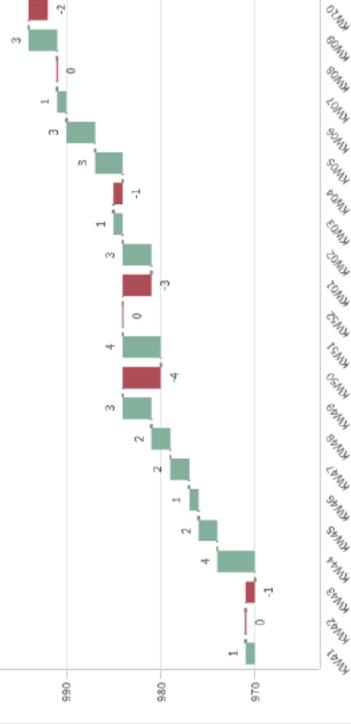
Region Q

Zentrum Q

Werte

Region Q	Zentrum Q	KW10 vs. 4		12 vs. 12		24 vs. 52		KW10		52 Wochen		KW10 2021 vs. 2022	
		Wochen	Ø	Wochen	Ø	Wochen	Ø	Wochen	Ø	Wochen	Ø	Wochen	Ø
Gesamt		0,00%	0,49%	0,62%	2,55%	992	992,00	987,25	981,17	956,81	9,13%		
Region A	Zentrum A1	-1,91%	-1,36%	-2,00%	-1,17%	77	78,50	81,21	82,17	82,17	-6,10%		
	Zentrum A2	0,31%	1,87%	1,42%	4,69%	82	81,75	80,25	79,13	75,58	17,14%		
	Zentrum A3	-0,91%	-1,68%	-0,79%	3,36%	82	82,75	84,17	84,83	82,08	9,33%		
	Zentrum A4	0,29%	1,29%	0,65%	1,75%	85	84,75	83,67	83,13	81,69	7,59%		
Region B	Zentrum B1	0,30%	0,30%	0,81%	3,76%	83	82,75	82,50	81,83	78,87	9,21%		
	Zentrum B2	-0,81%	2,27%	2,06%	2,42%	82	82,50	80,67	79,08	77,21	10,81%		
	Zentrum B3	2,13%	1,02%	-0,71%	-0,19%	84	82,25	81,42	82,00	82,15	2,44%		
	Zentrum B4	0,00%	1,08%	0,49%	3,49%	86	86,00	85,08	84,67	81,81	8,86%		
Region C	Zentrum C1	0,59%	1,81%	3,00%	3,90%	85	84,50	83,00	80,58	77,56	19,72%		
	Zentrum C2	0,00%	0,10%	1,04%	3,51%	85	85,00	84,92	84,04	81,19	14,86%		
	Zentrum C3	-0,62%	-0,41%	0,83%	2,88%	80	80,50	80,83	80,17	77,92	9,58%		

Patient Flow (aktuelles Jahr)



Wochenbericht

KW10

< >

Letzte Behandlung: 01.05.2022

🔄 Daten aktualisieren

🔍 HD-Aquivalenz

🔍 Verschiebe Vorgahr

Patienten in KW10

5

Trend der Patienten

↓ -4,76%

Behandlungen in KW10

13

Trend der Behandlungen

↓ -8,77%

🔍 Behandlungsgruppe

AN

CAPD

IPD

APD

HD

🔍 Behandlungsklasse

Akut

Chronisch

Gast

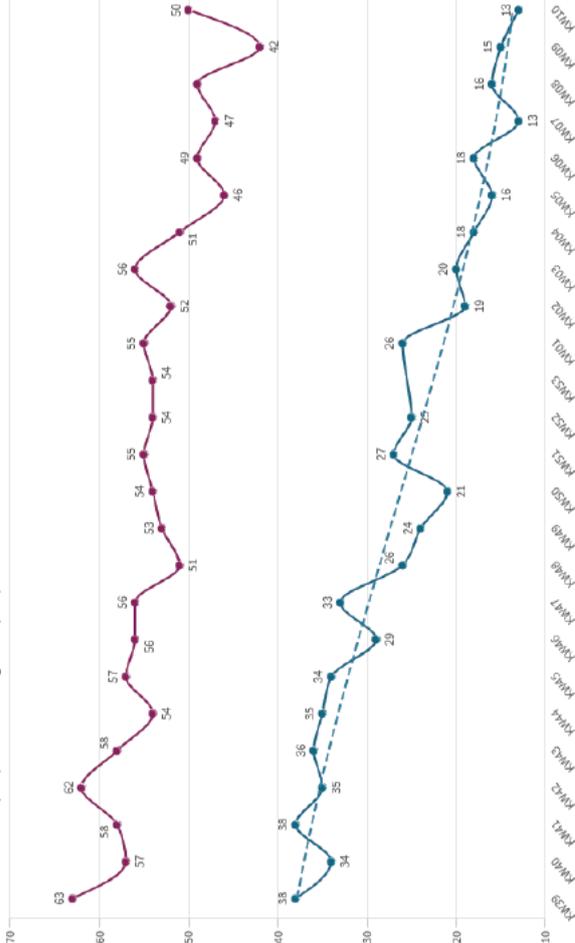
☰ Behandlungsgruppe



Patienten

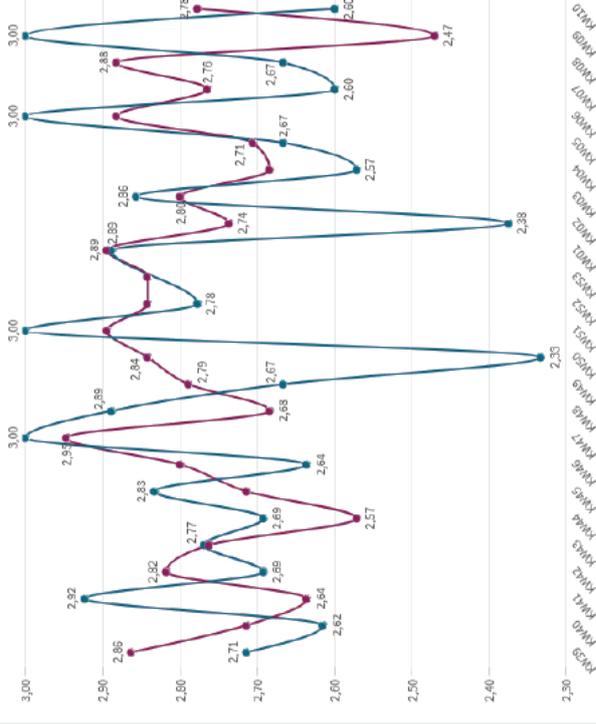
Behandlungen

🔵 Aktuelles Jahr (2022) 🔴 Vorheriges Jahr (2021)



Behandlungen pro Patient

🔵 Aktuelles Jahr (2022) 🔴 Vorheriges Jahr (2021)



Patienten KW10

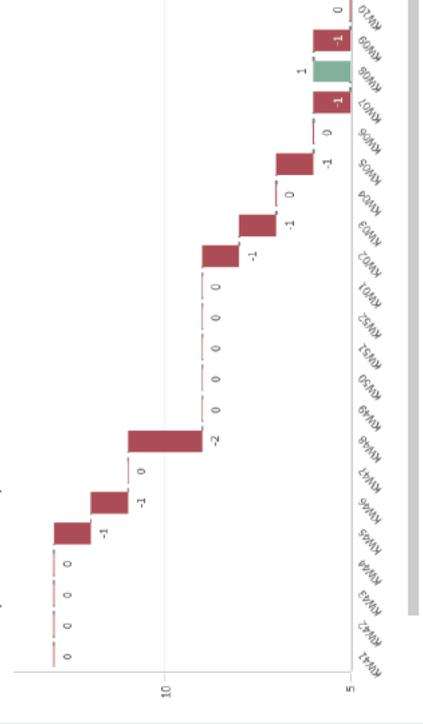
Behandlungen KW10

Region Q, Zentrum Q

Werte

Region Q	Zentrum Q	Behandlungen KW10					
		KW10 vs. 4 Wochen Ø	4 vs. 12 Wochen Ø	12 vs. 24 Wochen Ø	24 vs. 52 Wochen Ø	KW10	24 Wochen Ø
Gesamt		-4,76%	-23,17%	-26,13%	-25,19%	5	6,83
Region A	Zentrum A1	-	-100,00%	-50,00%	-46,26%	0	0,33
	Zentrum A2	0,00%	-4,000%	-21,57%	6,25%	1	1,67
	Zentrum A3	-	-	-	-	0	0,00
	Zentrum A4	-	-	-	-	0	0,00
Region B	Zentrum B1	-	-	-100,00%	-63,89%	0	0,00
	Zentrum B2	33,33%	20,00%	50,00%	-72,92%	1	0,75
	Zentrum B3	-	-100,00%	-33,33%	-20,41%	0	0,00
	Zentrum B4	-	-	-100,00%	-64,55%	0	0,00
Region C	Zentrum C1	-	-	-100,00%	-60,00%	0	0,00
	Zentrum C2	-	-	-100,00%	-35,42%	0	0,25
	Zentrum C3	-20,00%	-11,76%	-10,53%	-5,90%	2	2,83

Patient Flow (aktuelles Jahr)



Glossar

Business Intelligence BI dient der Präsentation und Analyse von geschäftsrelevanten Informationen und lässt sich in die Bereitstellung und Aufbereitung von Daten, sowie die Wissensgewinnung und abschließende Darstellung dieser unterteilen. Die Ausführlichkeit der Ausführung der einzelnen Punkte ist dabei abhängig von der spezifischen Anwendung und den damit verbundenen Anforderungen an die Ergebnisse des Business Intelligence Prozesses.

Common Table Expression Ein allgemeiner Tabellenausdruck ist die Ergebnismenge einer Abfrage, die nur vorübergehend und zur Verwendung im Kontext einer größeren Abfrage existiert.

Key Performance Indicator KPIs sind Leistungskennzahlen, welche den Fortschritt, das Erfolgsmaß oder die Auslastung wichtiger Zielsetzungen oder Komponenten in einem Unternehmen messen.

Nephrologie Die Nephrologie ist ein Teilgebiet der Inneren Medizin und beschäftigt sich vor allem mit der Prävention, Diagnostik, Therapie und Nachsorge von Nieren- und Hochdruckerkrankungen.

Niereninsuffizienz Die Niereninsuffizienz, auch Nierenversagen, bezeichnet die Fehlfunktion einer oder beider Nieren. Es wird die chronische von der akuten Niereninsuffizienz unterschieden, wobei die Funktion der Niere bei einem chronischen Versagen dauerhaft und kontinuierlich abnimmt und bei einem akuten Versagen plötzlich durch z. B. Operationen beeinträchtigt wird.

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original