

Wintersemester 2019/2020

Visualisierungen für IT-Konsolidierungsprojekte

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang WirtschaftsNetze

an der Fakultät Wirtschaftsinformatik

der Hochschule Furtwangen University

vorgelegt von

Katharina Schemel

Referenten:

Prof. Dr. Marianne Andres

Martin Jelen

Eingereicht am

13. Februar 2020

Eidesstattliche Erklärung

Ich Katharina Schemel erklären hiermit an Eides statt, dass ich die vorliegende Bachelorthesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe.

Die verwendeten Quellen sind vollständig zitiert.

Furtwangen, den 13. Februar 2020

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Ziel der Arbeit	3
2	Grundlagen	5
2.1	Enterprise Architecture Management	5
2.2	Visualisierungskonzepte	8
2.3	Funktionsweise einer Graph-Datenbank.....	12
2.4	Cypher	14
2.5	Node.js	16
2.6	Webserver	16
2.7	Webtechnologien zur Visualisierung (Überdenken ob passend)	18
3	Konzeption	19
3.1	Infrastrukturaufbau.....	19
3.2	Datenmodellkonzept.....	20
3.3	Visualisierungskonzept für Graphen	22
3.4	Umsetzungsmöglichkeiten für künftige IT-Systementwürfe	26
3.5	Konzeption der Entwicklungsumgebung.....	29
4	Implementierung	32
4.1	Verwendete Technologien	33
4.2	Vorgehen bei der Visualisierung.....	34
4.3	Interpretation und Ergebnisse.....	34
5	Evaluation (Fazit)	42
6	Zusammenfassung und Ausblick	44
6.1	Zusammenfassung	44
6.2	Ausblick	45
	Glossar	48
	Anhang	49

Abbildungsverzeichnis

Abbildung 1: Brücke zwischen Business und IT (Business-IT)	6
Abbildung 2: Bebauungsplan Grafik.....	9
Abbildung 3: Funktionales Referenzmodell.....	10
Abbildung 4: Portfolio-Grafik	11
Abbildung 5:Zusammenhang Entitäten-Beziehungsmodell, Relationenmodell, Graphenmodell.....	13
Abbildung 6: Ergebniss Relationenabfrage	15
Abbildung 7: Ergebniss Relationenabfrage mit Einschränkungen auf Attributwerte	15
Abbildung 8: Client-Server Modell.....	17
Abbildung 9: Infrastrukturkonzept	19
Abbildung 10: Datenmodell	20
Abbildung 11: Mockup Grad der Vernetzung	23
Abbildung 12: Mockup Beziehung zu	23
Abbildung 13: Mockup Ampelsystem	24
Abbildung 14: Mockup Nutzeranzahl	24
Abbildung 15: Technologie Radar	26
Abbildung 16: Konzept der Entwicklungsumgebung	29
Abbildung 17: Windows PowerShell	36
Abbildung 18: Neo4j Datenbankauszug	36
Abbildung 19: Visual Studio Code.....	37
Abbildung 20: Notepad++	38
Abbildung 21: Excel Daten.....	39
Abbildung 22: Windows PowerShell	40
Abbildung 23: Daten Webansicht.....	41

1 Einleitung

In der Realität ist es oft so, dass innerhalb eines Unternehmens viele verschiedene IT-Systeme verwendet werden, welche schnell schwierig zu überblicken sind. Diese Systeme beinhalten Programme, welche auch gepflegt werden müssen. Dies resultiert beispielsweise daraus, dass diese immer auf dem aktuellen Stand sein sollten, um Sicherheitslücken zu vermeiden. In großen Unternehmen hat man kaum einen Überblick, welche IT-Systeme in den einzelnen Abteilungen verwendet werden. Hier empfiehlt sich eine Konsolidierung vorzunehmen. Das bedeutet, dass die IT-Systeme sozusagen verschlankt werden.

Ein Aspekt der Konsolidierung ist die Ineffizienz von IT-Systemen zu eliminieren. Oft werden verschiedene Systeme verwendet, welche im Prinzip, das exakt gleiche tun. Daher wäre es von Vorteil eine einheitliche Struktur festzulegen, um den Bestand der Software zu vereinfachen.

Ein weiterer Aspekt bei der Konsolidierung ist die Entscheidung, welche Komponenten konsolidiert werden sollten. Zu beachten ist jedoch, dass durch die Konsolidierung nie eine konkrete Entscheidung gefällt wird welche Komponenten konsolidiert werden, sondern diese lediglich eine Entscheidungsgrundlage darstellt. Der Anstoß für eine IT-Konsolidierung kann aus aktuellen Anlässen erfolgen, strategischen Nutzen für das Unternehmen bieten oder einfach von der Unternehmensführung gewünscht sein.

All diese Gründe führen zu der Konsolidierung, um beispielsweise den Geschäftswert bzw. die Betriebskosten des Unternehmens zu optimieren oder zur Zuordnung von Geschäftsprozessen, welche zu Outsourcing führen können. Somit ist deutlich zu erkennen, dass die Konsolidierung ein wichtiger Aspekt in Unternehmen darstellt.

Das Problem bei der Konsolidierung ist jedoch, die einzelnen Komponenten zu identifizieren. Eine Datenbeschaffung ist recht zeitaufwendig und kann fast nie vollständig erfolgen. Hieraus entsteht meist ein Vorgehen auf Basis von Teilbeständen, was letztlich nicht dem Ideal entspricht.

Die Komponente Visualisierung wird sowohl für die Präsentation der Analyseergebnisse wie auch als Werkzeug für die Analyse eingesetzt. Um die Analyse zu verbessern ist es vorstellbar Graphdatenbanken in diesen Prozess mit einzubinden, da ein Graph schnell einen Überblick über Beziehungen und Zusammenhänge der einzelnen

Komponenten vermittelt. Daraus kann eine Entscheidungsgrundlage gebildet werden, welche Komponenten entsprechend eliminiert oder verändert werden sollten.

1.1 Motivation

Die Thematik der IT-Konsolidierung ist noch recht jung und sehr aktuell, beispielsweise hat die IT-Konsolidierung des Bundes erst 2015 begonnen. Daher ist es auch sehr spannend diese Thematik weiter voran zu bringen. Auslöser für eine IT-Konsolidierung kann z. B. eine vorherige Fusion darstellen. Laut (Bundesministeriums des Innern, für Bau und Heimat, 2020) sind die folgenden Punkte Ziele einer IT-Konsolidierung:

- Informationssicherheit trotz steigender Komplexität zu gewährleisten
- die eigene IT soll zu jederzeit Souverän und beherrscht sein
- flexibel auf innovative technologische Trends reagieren zu können
- ein zukunftsfähiger Betrieb, welcher leistungsfähig und stabil ist
- für das IT-Fachpersonal ein attraktiver Arbeitgeber bleiben

IT-Konsolidierung: Eine einfache Definition

„Konsolidierung bedeutet einfach übersetzt „Zusammenführung“. Im Bereich der IT sollen also Infrastrukturen, Datenbestände und Anwendungen zusammengeführt und idealerweise auch vereinheitlicht werden. Dadurch werden Kosten gespart, Abläufe vereinfacht sowie beschleunigt und die IT-Qualität erhöht sich insgesamt.“ (Klein, 2017)

Diese Definition zeigt, dass IT-Konsolidierung sehr wichtig ist, da neben einer Kosteneinsparung auch die Qualität der IT erhöht werden soll, was insgesamt eine Bereicherung für die heutige Zeit darstellt, in welcher die Themen Digitalisierung und Industrie 4.0 immer allgegenwärtiger werden. Auch die Zusammenführung von Infrastrukturen, Datenbeständen und Anwendungen erleichtern das tagtägliche Arbeiten, sowie die Pflege der Software erheblich. Die Konsolidierung wird im Enterprise Architecture Management (EAM) durchgeführt, da diese einen konzeptionellen und organisatorischen Rahmen bietet, damit die Architektur zielgerichtet aufgebaut und erweitert werden kann. Zudem kann die Architektur Zusammenhänge z. B. mittels Diagramme sehr gut darstellen. Besonders bei komplexen Sachverhalten ist diese Weise deutlich effizienter.

Die Thematik der Visualisierung im Kontext IT-Konsolidierung ist ein wichtiger Bestandteil dieser Arbeit, da eine Präsentation der Analyseergebnisse mittels Excel-Tabellen, wie es häufig von IT-Beratern praktiziert wird, nur einen geringen Mehrwert bietet.

1.2 Ziel der Arbeit

Das Ziel der Arbeit besteht sozusagen in der „Grundsteinlegung“ auf dem Gebiet der Visualisierung von IT-Konsolidierungsprojekten mit Anbindung an eine Graphdatenbank für die Firma ISB AG. Die Arbeit gliedert sich somit in einen theoretischen wie auch einen praktischen Teil. Der theoretische Teil umfasst die Einarbeitung auf den Themengebieten EAM, Graphdatenbanken und Visualisierung mittels D3JS. Neben der Einarbeitung stellt sich die zentrale Frage, wie Graphdatenbanken und die Visualisierung auf dem Gebiet der IT-Konsolidierungsprojekte zusammen zu bringen sind. Hierfür sollten prototypische Visualisierungskonzepte erarbeitet werden, unter Berücksichtigung von typischen Visualisierungskonzepten aus dem Bereich des EAM.

Es sei jedoch zu beachten, dass nicht jeder Datensatz, aufgrund seiner spezifischen Eigenschaften, zu jedem Visualisierungskonzept passt. Aus diesem Grund sollten Datenbestände zunächst analysiert werden und dementsprechend eine geeignete Visualisierung aufgrund der Daten gewählt werden. Bezüglich einer Graphdatenbank folgt aus einer derartigen Analyse der Daten in der Regel ein Datenmodell, welches im Zuge dieser Arbeit für eine fiktive IT-Landschaft prototypisch zu erstellen war. Dieses wiederum bildete die Grundlage für die Erstellung der Visualisierungskonzepte.

Die Grundlagenlegung für den praktischen Teil resultierte wiederum in der Erweiterung des theoretischen Teils. Es ging im Zuge dessen darum ein Konzept für eine Infrastruktur für den Betrieb des im praktischen Teil entwickelten Prototyps zu generieren. Weiterhin musste ein Konzept für die Entwicklungsumgebung geschaffen und entsprechend umgesetzt werden, um eine nachhaltige und einfache Organisation des zu erstellenden Softwareprojekts zu gewährleisten. Folglich umfasst das Ziel des praktischen Teils die Entwicklung eines Prototyps, welcher diverse Kernfunktionalitäten exemplarisch implementiert. Diese sind unter anderem die Anbindung an eine Graphdatenbank sowie die Visualisierung der angefragten Daten mittels der JavaScript-Bibliothek D3JS.

2 Grundlagen

2.1 Enterprise Architecture Management

Die folgenden Definitionen von Enterprise Architecture und Enterprise Architecture Management zeigen die konkreten Abgrenzungen voneinander auf. In der Praxis werden diese Begriffe oft als Synonym verwendet, jedoch gibt es gravierende Unterscheidungen zwischen diesen beiden Begriffen.

Definition Unternehmensarchitektur (EA):

„Eine Unternehmensarchitektur (Enterprise Architecture) schafft eine gesamthafte Sicht auf das Unternehmen. Sie legt die wesentlichen fachlichen und IT-Strukturen fest und verknüpft sie miteinander. Auf dieser Basis lassen sich das Business und die IT und ihre Zusammenhänge beschreiben. Eine gemeinsame Sprachbasis, „eine Brücke“ zwischen Business und IT, wird geschaffen. So kann die strategische Weiterentwicklung von Business und IT aktiv gesteuert werden.“ (Hanschke, 2013)

Definition EAM:

„EAM ist ein systematischer und ganzheitlicher Ansatz für das Verstehen, Kommunizieren, Gestalten und Planen der fachlichen und technischen Strukturen im Unternehmen. Es hilft dabei, die Komplexität der IT-Landschaft zu beherrschen und die IT-Landschaft strategisch und businessorientiert weiterzuentwickeln. EAM ist ein wesentlicher Bestandteil des strategischen IT-Managements und beinhaltet alle Prozesse für die Dokumentation, Analyse, Qualitätssicherung, Planung und Steuerung der Weiterentwicklung der IT-Landschaft und der Geschäftsarchitektur.“ (Hanschke, 2013)

Einfach gesagt, fungiert die Enterprise Architecture als eine Art Vermittlerrolle zwischen der IT und den entsprechenden Fachbereichen eines Unternehmens, so (BITKOM, 2011). Man könnte es auch bildlich als eine Brücke beschreiben, welche diese Bereiche verbindet.



Abbildung 1: Brücke zwischen Business und IT (Business-IT)

Durch diese Verbindung wird ein Rahmen für den Ausbau der IT-Landschaft zur Verfügung gestellt. Dieser Rahmen umfasst laut (BITKOM, 2011) strategische, konzeptionelle und organisatorische Aspekte. Weiter führt (BITKOM, 2011) aus, dass besonders die Methoden der Umsetzung und die Überprüfung im Hinblick einer Kosten-Nutzen-Betrachtung im Fokus stehen.

Das Enterprise Architecture Management wiederum „umfasst die Aufgaben zur Erstellung, Pflege und Umsetzung einer EA“ (BITKOM, 2011). Man kann also sagen, dass das EAM ein strukturierter Ansatz für die Erstellung, Verwaltung und Nutzung der von EA bereitgestellten Modelle ist.

Je größer ein Unternehmen wird, desto bedeutender ist die Rolle des EAM. Durch neue Technologien und Schnittstellen steigt die Komplexität und das Bewältigen der IT-Landschaft wird immer schwieriger. Durch fehlendes EAM können Redundanzen und inkonsistente Daten auftreten. Diese Problematik ist unter anderem die Basis für die IT-Konsolidierung. Aber auch mit EAM kann über einen längeren Zeitraum eine IT-Konsolidierung notwendig werden.

Durch das Bereitstellen diverser Hilfsmittel durch EAM, kann die Kontrolle über die IT-Landschaft wiederhergestellt werden und sie kann strategisch und businessorientiert weiterentwickelt werden.

Zudem wird EAM oft eingesetzt, da es Transparenz schafft. Diese ist maßgeblich für die Bewertung der IST-Situation und zeigt bei der IT-Landschaft schon nach kurzer Zeit, in welchem Bereich sich Kosten einsparen lassen.

Laut (Hanschke, 2013) kann in Problemfällen durch entsprechende Visualisierungen schneller Klarheit verschafft werden, z. B. welche Geschäftsprozesse von einem Ausfall des IT-Systems betroffen sind, wo die entsprechenden Verantwortlichkeiten liegen oder welche Abhängigkeiten zwischen den IT-Systemen bestehen.

Um die Leistungsfähigkeit der IT-Leistung einordnen zu können, werden in der Praxis oft relative Bewertungen wie z. B. sehr hoch, hoch, mittel und gering verwendet. Diese Bewertung ist einfach und verständlich, damit sie von Fachbereichen, Anwendern und Managern angewendet werden kann. Die Fachbereiche und die Fachfremden Personen sollten jeweils eine solche Bewertung vornehmen, um festzustellen in wie weit sich das Eigen- und Fremdbild unterscheidet und anschließend sollte entsprechend vorgegangen werden, um diese beiden Ansichten anzugleichen, führt (Hanschke, 2013) weiter aus.

2.2 Visualisierungskonzepte

Farben und Muster sprechen den Menschen schneller an als eine einfache Tabelle mit Informationen. Daher ist eine Visualisierung von Daten wichtig, um das Interesse schneller auf relevante Aspekte zu lenken. Bei der Darstellung von Inhalten ist es wichtig sich vorab zu überlegen, welcher Sachverhalt näher dargestellt werden soll und entsprechend können hierfür unterschiedliche Visualisierungsarten mit passenden Eigenschaften herangezogen werden.

Grundsätzlich sei an dieser Stelle darauf verwiesen, dass nicht jede Visualisierungsart für die Darstellung eines beliebigen Sachverhalts geeignet ist. Beispielsweise kann ein Ampel Diagramm sehr gut verwendet werden, um einen Fortschritt in einem Prozess oder die Kritikalität einer Eigenschaft darzustellen. Es ist jedoch nicht geeignet komplexere Beziehungen zwischen Personen, Prozessen oder Softwareprodukten darzustellen.

Auch für den Bereich des EAM gibt es mehrere gängige Visualisierungsarten, welche laut (Hanschke, 2013) die Aussagekraft der Informationen unterstreicht. Auch auf diesem Gebiet gestaltet sich der Einsatzbereich solcher Grafikkonzepte analog zum Einsatzbereich der Ampel Grafik. Sie sind jeweils Anhand ihrer Eigenschaften für die Darstellung gewisser Sachverhalte besser geeignet und für andere schlechter. Beispiele für Visualisierungsarten aus dem Bereich des EAMs seien gegeben durch folgende Grafikkonzepte:

- Bebauungsplan-Grafik
- Cluster-Grafik
- Portfolio-Grafik

Die drei genannten Grafikkonzepte wurden exemplarisch aus dem Best-Practice-Visualisierungskatalog gewählt. Diese zeigen durch ihre unterschiedlichen Ausprägungen die verschiedenen Möglichkeiten, welche eine Visualisierung im Bereich des EAM leisten kann. Diese drei Grafikkonzepte werden folgend näher erläutert und zur Veranschaulichung nach Vorlage von (Hanschke, 2013, 2016) nachgestellt.

- Bebauungsplan-Grafik

Diese Grafik stellt die Zusammenhänge zwischen Elementen der Unternehmensarchitektur in Form einer Matrix dar. Somit können u.a. Informationssysteme zu Geschäftsprozessen und Geschäftseinheiten in Beziehung gesetzt werden. Durch die flexible Struktur können viele Fragestellungen beantwortet werden. Der Einsatz von Farben und diversen Linientypen unterstreicht diese Visualisierung nochmals. Die Bebauungsplan-Grafik unterteilt sich in drei Unterpunkte. Für das EAM ist besonders die Ausprägung der „typischen“ Bebauungsplan-Grafik interessant, da diese eine verbreitete Form zur IT-Unterstützung des Unternehmens ist. In einem Bebauungsplan der IT-Branche werden laut (BITKOM, 2011) gegenwärtige und zukünftige Infrastrukturen, als auch Anwendungssoftware definiert. Diese sollen die Geschäftsprozesse eines Unternehmens unterstützen.

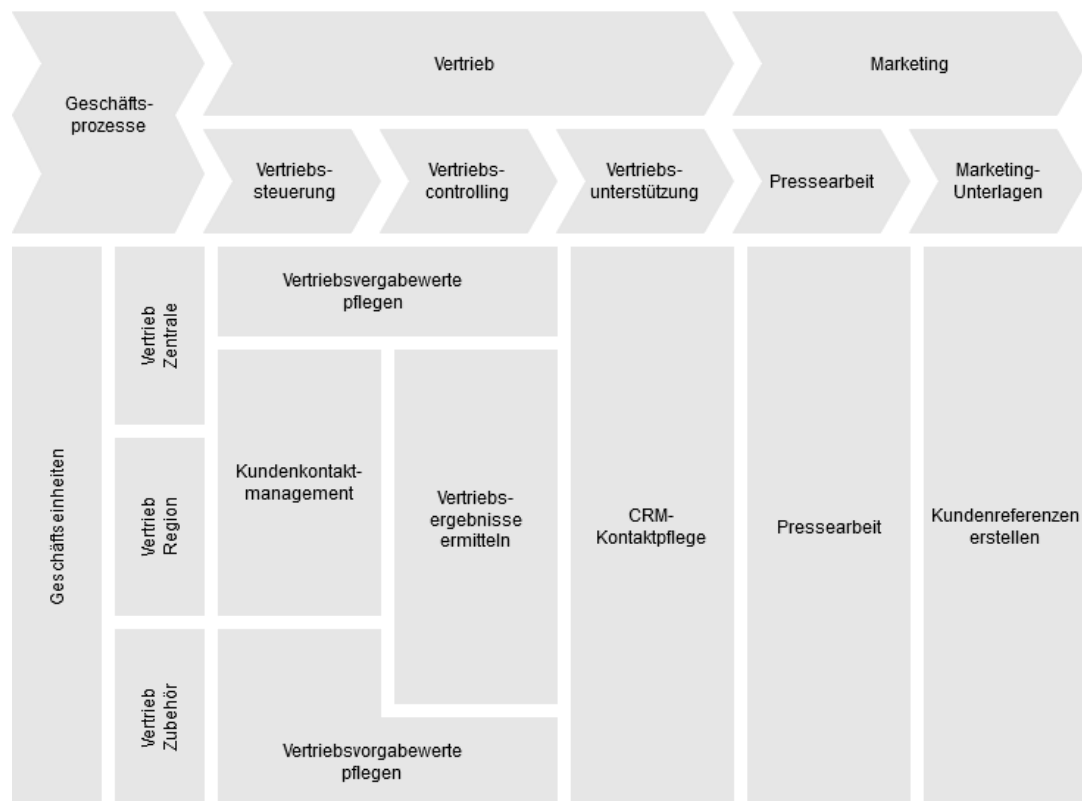


Abbildung 2: Bebauungsplan Grafik

Diese Abbildung zeigt die Geschäftsprozesse auf der X-Achse und die Geschäftseinheiten des Vertriebs auf der Y-Achse. Die einzelnen Bebauungselemente in mitten dieser Grafik geben einen guten Überblick, welche Geschäftseinheiten für welche Geschäftsprozesse zuständig sind.

Diese Art der Grafik ist jedoch nicht zwingend kompatibel zu größeren Datenmengen mit einer netzwerkartigen Struktur, da Abhängigkeiten unter den Daten deutlich schwieriger dargestellt werden können. Dies resultiert aus der blockartigen Anordnung, welche ab einer gewissen Datenmenge eine optische Clusterung nur bedingt zulässt.

- Cluster-Grafik

Das Konzept der Cluster-Grafik beschreibt das Aufteilen und Gruppieren von Bebauungselementen, anhand von Eigenschaften, Beziehungen oder definierten Kriterien. Bekannte Ausprägungen der Cluster-Grafik sind z. B. das Fachliche Domänenmodell, welches die Ausprägung einer Prozesslandkarte oder eines funktionalen Referenzmodells hat. Im Fachlichen Domänenmodell werden die Kernstrukturen der Geschäftsarchitektur festgelegt, daher gibt es laut (Hanschke, 2013) innerhalb eines Unternehmens auch nur ein Fachliches Domänenmodell.

Grafik

Abbildung 3: Funktionales Referenzmodell

Die Abbildung 3 zeigt ein funktionales Referenzmodell. Dieses Eignet sich bis zu einem gewissen Grad, Abhängigkeiten zwischen Bebauungselementen darzustellen. Dies wird in diesem Fall auf Basis einer verschachtelten Darstellung realisiert. Somit eignet sich diese Darstellungsart, um die Zusammengehörigkeit zwischen Elementen der IT-Landschaft eines Unternehmens darzustellen. Die Grenzen werden jedoch erreicht, sobald Elemente in mehreren Gruppierungen zugehörig sind, da dies maximal durch eine redundante Aufführung kompensiert werden kann. Dies kann jedoch bei größeren Darstellungen wiederum die Lesbarkeit bzw. Interpretierbarkeit der Grafik beeinträchtigen. Es kann somit auch nicht auf einfachem Wege dargestellt werden wie stark ein Softwareprodukt in ein Unternehmen integriert ist.

Portfolio-Grafik

Bei dieser Grafik können besonders gut sogenannte Wertigkeiten von Bebauungselementen oder auch Strategien für Bebauungselemente visualisiert werden. Ebenso eignet sie sich für die Abbildung von Nutzenpotenzialen oder Risiken. Hierbei kann der Ist-Zustand oder auch der Soll-Zustand bzw. deren Differenz dargestellt werden. In einer Portfolio Grafik können maximal fünf verschiedene Kriterien abgebildet werden. Diese sind jeweils die Achsen, sowie die Größe, Farbe und der Kantentyp der Füllelemente.

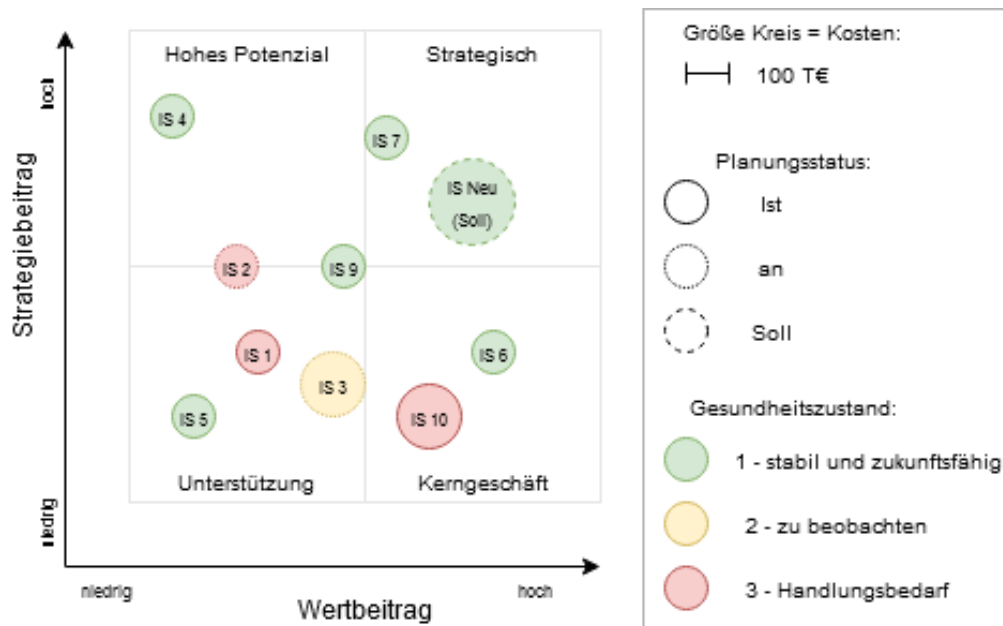


Abbildung 4: Portfolio-Grafik

Durch diese Grafik kann schnell ein Überblick anhand der Positionierung verschafft werden. Neben der Position, gibt auch die Größe der einzelnen Elemente Aufschluss über beispielsweise Kosten. Diese Grafik eignet sich besonders gut, um messbare Eigenschaften oder eine kategorische Einordnung von Elementen darzustellen. Abhängigkeiten zwischen Elementen können maximal über die kategorische Einordnung realisiert werden.

2.3 Funktionsweise einer Graph-Datenbank

Jeder ist bereits mit einem Graphen in Berührung gekommen. Sei es bei der Erstellung eines Mindmaps oder beim Skizzieren von Symbolen und Linien auf einer Tafel. Graphen sind sehr einfach, verständlich und vielseitig einsetzbar, nach (Hunger, 2014). Eine der bekanntesten Graph Datenbanken ist Neo4j. Der Name Neo4j leitet sich aus „*Network Engine for Objects*“ ab. Die Zahl „4“ vermittelt eine Aussage über die Versionsnummer und das „j“ steht dafür, dass die frühere Java-API entsprechend weiterentwickelt wurde. Zu Beginn war die Bedeutung von „4j“ „für Java“, führt (Trelle, 2017) aus. Mittlerweile kann, neben Java, auch in der Programmiersprache Scala entwickelt werden. Daher ist die Bedeutung „for Java“ nicht mehr ganz aktuell.

Eine Graph-Datenbank gehört zu der Gruppe der NoSQL Datenbanken. Dies bedeutet, dass die Datenbank in der Regel einen nicht relationalen Ansatz verfolgt. Jedoch wird die Verwendung von SQL nicht gänzlich ausgeschlossen, da NoSQL für „Not only SQL“ steht.

Eine Relationale Datenbank verwendet Tabellen, welche Spalten und Zeilen für die Speicherung der Daten nutzt. Die NoSQL Datenbank hingegen nutzt für die Organisation der Daten beispielsweise Attribut-Wert-Paare (Properties), Objekte, Dokumente oder Listen und Reihen für die Organisation der Daten. Ein großer Vorteil von NoSQL Datenbanken ist, dass sie dort ansetzen, wo SQL-basierte relationale Datenbanken an ihre Grenzen stoßen. NoSQL Systeme eignen sich besonders gut für große, exponentiell wachsende Datenmengen, um diese performant zu verarbeiten, wie beispielsweise im Bereich von Big Data. Dies resultiert aus der einfachen Unterstützung der Datenbankfragmentierung, welche durch den Verzicht auf ein Datenbankschema und auf die referenzielle Integrität ermöglicht wird. Im speziellen besagt die referenzielle Integrität, dass Datensätze nur auf bestehende Datensätze verweisen dürfen. Entsprechend kann auch nur ein Datensatz gelöscht werden, wenn auf diesen kein anderer Datensatz verweist.

Graph Datenbanken werden nicht tabellarisch geführt, sondern in Form von Labeln organisiert. Diese sind zwar vergleichbar mit den Tabellen der relationalen Datenbanken, haben jedoch den Unterschied, dass diese ohne eine feste Vorgabe der enthaltenen Attribute auskommt. Die Speicherung der Daten erfolgt somit Schemafrei. Dies erlaubt zusätzliche Informationen wie mangelnde Attribute zu einem späteren Zeitpunkt problemlos nachzutragen. Ein implizites strukturiertes Schema sei dennoch durch das Graphenmodell innerhalb einer Graphdatenbank gegeben. Das Graphenmodell kann als Äquivalent zum Relationenmodell der Relationalen Datenbanken interpretiert werden. Es resultiert analog zum Relationenmodell aus dem Entitäten-Beziehungsmodell. Dieser Zusammenhang sei mit Abbildung 5 nach (Meier et al., 2016) näher verdeutlicht.

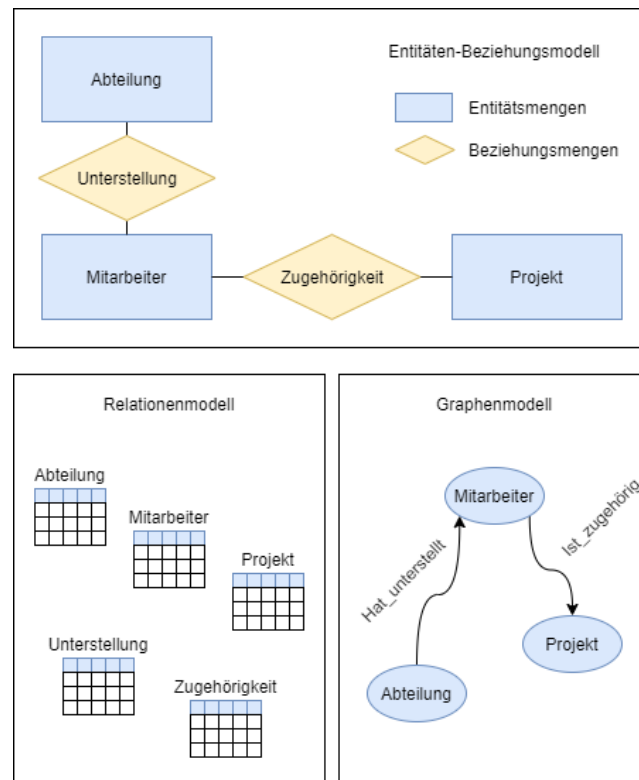


Abbildung 5: Zusammenhang Entitäten-Beziehungsmodell, Relationenmodell, Graphenmodell

Laut (Maier et al., 2016) werden die zu speichernden Daten mithilfe von so genannten Knoten und Kanten dargestellt. Hier werden Objekte in Knoten und Beziehungen zwischen Knoten in Kanten abgebildet. In einem Knoten ist die Beziehung nicht in ihrer Art oder Anzahl beschränkt, ergänzt (Matzer, 2019).

In der Literatur werden Knoten häufig auch als Vertex (V) und Kanten als Edges (E) benannt. Dies sei an dieser Stelle bezüglich der Vollständigkeit erwähnt. Folgend wird diese Terminologie jedoch nicht weiterverwendet.

Die Kante stellt eine Verbindung zwischen den Knoten dar. Diese Verbindung kann als eine Linie oder als ein Pfeil visualisiert werden. Eine Kante ist gerichtet, das bedeutet, dass sie einen Start- sowie einen Endpunkt besitzt. Die Beziehungen zwischen den Knoten können Eigenschaften besitzen. Diese Daten können analog zu den Attributen der Knoten abgefragt werden. Nach (Luber et al., 2017) verzichtet eine Graphdatenbank auf verschachtelte Beziehung, was die hohe Performance für die Speicherung, sowie Abfrage der Informationen erklärt.

Die Information, welche Beziehungen zwischen Daten vorliegen, ist unter gewissen Umständen besonders förderlich. Diese Beziehungen kann ein Graph ausgesprochen gut darstellen. Hieraus ergibt sich der hohe Nutzen der Verwendung einer Graph Datenbank für Daten, welche eine netzwerkartige Struktur aufweisen.

Eine Besonderheit der Graphdatenbanken ist nach (Maier et al., 2016), die Eigenschaft der indexfreien Nachbarschaft. Das Datenbanksystem kann somit die direkten Nachbarn eines Knotens ermitteln, ohne sämtliche Kanten berücksichtigen zu müssen. Dies ist beispielsweise in relationalen Datenbanksystemen nötig, welche sogenannte Beziehungs- oder Verknüpfungstabellen verwenden. Aus diesem Sachverhalt ergibt sich, dass die Abfrage von Beziehungen auf Knoten unabhängig von der gespeicherten Datenmenge sind und somit eine konstante Geschwindigkeit besitzen.

Laut (Matzer, 2019) nutzen bereits große Unternehmen wie z. B. Google, Facebook, Microsoft und LinkedIn die Graph Datenbank, ebenso haben auch Unternehmen wie die NASA oder die Schweizer Bank UBS die Graph Datenbank für sich entdeckt.

Die gängigsten Abfragesprachen für Graph Datenbanken sind Apache TinkerPop Gremlin, SPARQL und Cypher. Da, in dieser Arbeit mit der Graph Datenbank Neo4j gearbeitet wird, betrachtet diese Arbeit folgend nur die Abfragesprache Cypher näher. Cypher wurde von Neo4j entwickelt und erfüllt somit die besten Voraussetzungen was Kompatibilität angeht, nach (Matzer, 2019). Zudem ist Cypher sehr verbreitet und in seiner Struktur der Abfragesprache SQL von Relationalen Datenbanken recht ähnlich. Durch diese ähnliche Struktur ist eine Umstellung von SQL auf Cypher nicht zu zeitaufwendig und komplex. In Kapitel 2.4 wird auf die Abfragesprache Cypher näher eingegangen.

2.4 Cypher

Cypher gehört zu der Gruppe der „deklarativen Abfragesprachen“. Die bekannteste Abfragesprache dieser Gruppe ist SQL. Der Fokus der deklarativen Abfragesprachen ist die Beschreibung eines Problems. Gegenüber den deklarativen Abfragesprachen steht die Gruppe der „imperativen Sprachen“, wie beispielsweise Java oder C++. Nach (Böhm, 2005) ist die imperative Programmierung ein Programmierstil, nach welchem „ein Programm aus einer Folge von Anweisungen besteht, die vorgeben, in welcher Reihenfolge was vom Computer getan werden soll“.

Wird folgend Cypher als problembeschreibende Sprache in Bezug auf Graphen betrachtet, kann von der Beschreibung von Mustern innerhalb von Graphen gesprochen werden. An folgendem Beispiel wird dieser Sachverhalt verdeutlicht. Es wird ein Graph angenommen, bestehend aus zwei Klassen von Knoten und einer gerichteten Beziehung zwischen den Knoten. Exemplarisch seien dies für die Knoten „Filme“ und „Schauspieler“ und für die Beziehung „spielt_in“. Entsprechend beschreibt der exemplarische Graph, welcher Schauspieler in welchem Film spielt. Die Beschreibung des Musters des dargelegten Graphen mittels Cypher stellt sich wie folgt dar.

```
Match (s:Schauspieler) – [spielt_in] -> (f:Film) Return s, f
```

Die entsprechende Rückgabe sind alle Knoten und deren Verbindungen untereinander, wie in Abbildung 6 dargestellt. Filme, zu denen keine Schauspieler in der Datenbank hinterlegt sind, werden nicht ausgegeben, da sie dem abgefragten Muster nicht entsprechen.

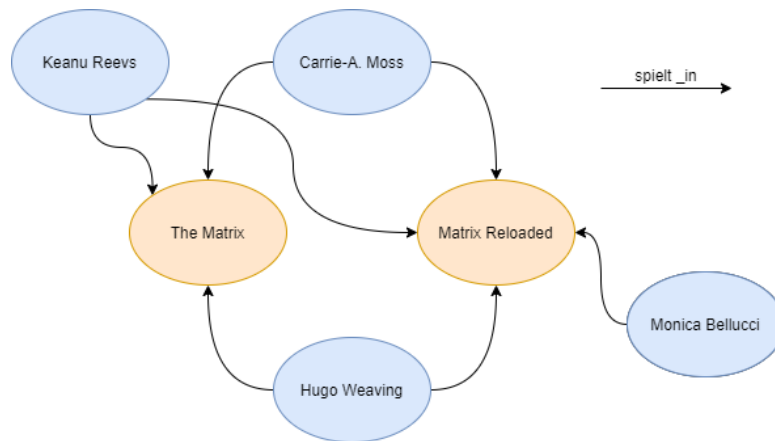


Abbildung 6: Ergebniss Relationenabfrage

Analog zu SQL bietet Cypher die Option, anhand von Attributen der Knoten und Kanten mittels einer Where-Bedingung das Muster zu Filtern. Folgend sei dies mittels Cypher dargestellt.

```
Match (s:Schauspieler) – [spielt_in] -> (f:Film)
Where (s.name = „Carrie-A. Moss“)
OR (s.name = „Monica Bellucci“)
Return s, f
```

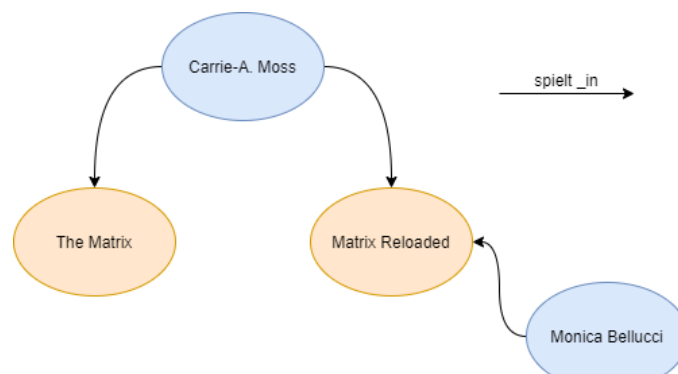


Abbildung 7: Ergebniss Relationenabfrage mit Einschränkungen auf Attributwerte

2.5 Node.js

Node.js ist eine plattformübergreifende Laufzeitumgebung basierend auf JavaScript. Verwaltet wird das Open-Source-Projekt NodeJS durch die OpenJS Foundation mit Unterstützung der Linux Foundation. Die ursprüngliche Hauptaufgabe von NodeJS besteht in der Ausführung von JavaScript-Code außerhalb eines Browsers. Es wird somit möglich serverseitig dynamische Webseiteninhalte zu erstellen, bevor die Seite an den Benutzer übertragen wird. Entsprechend wird deutlich, dass NodeJS primär für die Entwicklung hoch performanter Webserver verwendet wird. Nativ wird von NodeJS nur JavaScript unterstützt. Es gibt jedoch Compiler, die es ermöglichen andere Sprachen, wie zum Beispiel CoffeeScript oder TypeScript in JavaScript zu übersetzen und somit für NodeJS nutzbar zu machen.

Auf der funktionalen Ebene ist NodeJS bezüglich der dynamischen Erstellung von Webseiteninhalten am ehesten mit PHP zu vergleichen, welches dies analog zu NodeJS ebenso unterstützt. Auf der technischen Ebene unterscheidet sich NodeJS jedoch deutlich von PHP. NodeJS führt Funktionen in der Regel parallel aus und verwendet sogenannte „Callbacks“ um die Fertigstellung oder das Scheitern einer Funktion zu signalisieren. PHP hingegen blockiert die meisten Funktionen bis zur Fertigstellung der vorherigen.

Neben der beschriebenen Hauptaufgabe wird NodeJS jedoch auch sukzessive als Werkzeug für diverse andere Aufgaben eingesetzt. In dieser Arbeit wurde NodeJS beispielsweise indirekt als Bestandteil der Entwicklungs- und Deploymentumgebung verwendet. Dies wird ermöglicht, durch den Node Package Manager (npm), welcher ein integrierter Bestandteil von NodeJS ist. Der npm ist ein Paketmanager für die Laufzeitumgebung von NodeJS, was letztlich als freies Repository für Paketerweiterungen für NodeJS interpretiert werden kann. Am Beispiel dieser Arbeit wurde eine Reihe von Paketerweiterungen wie beispielsweise „webpack“ oder „babel“ für die Entwicklung und das Deployment eingesetzt. Diese Erweiterungen reichen von Compilern bis hin zu Webservern für die lokale Entwicklung. Auf die im Rahmen dieser Arbeit eingesetzten Paketerweiterungen wird [in Kapitel 4](#) spezifischer eingegangen.

2.6 Webserver

Ein Webserver ist schlicht eine Software, welche serverseitig installiert wird. Die Hauptfunktion eines Webserver ist nach [\(Killelea, 2002\)](#) das Speichern, Verarbeiten und Übermitteln von Webseiten an Clients. Er operiert somit im sogenannten „Client-Server Modell“, welches in Abbildung 8 näher veranschaulicht ist. Ein Webserver kann im Allgemeinen eine oder mehrere Webseiten enthalten. Bei den bereitgestellten Seiten handelt es sich am häufigsten um HTML-Dokumente, die neben dem Textinhalt auch Bilder, Stylesheets und Skripte enthalten können.

Für Clientanfragen und Serverantworten ist das Hypertext Transfer Protocol (HTTP) von zentraler Bedeutung. Der Client stellt eine HTTP-Anfrage mit einem Uniform Resource Locator (URL), durch welchen die angeforderte Resource serverseitig identifiziert werden kann. Anschließend antwortet der Webserver nach der Anfragenverarbeitung dem Client mit einer entsprechenden Nachricht. Diese enthält wiederum Informationen zum Status der Verarbeitung (erfolgreich oder gescheitert) und gegebenenfalls den angeforderten Inhalt beziehungsweise die angeforderte Resource.

Während die Hauptfunktion darin besteht, Inhalte bereitzustellen, umfasst eine vollständige Implementierung von HTTP auch Möglichkeiten zum Empfangen von Inhalten. Diese Funktionalität wird zum Senden von Webformularen verwendet bis einschließlich des Hochladens von Dateien.

Viele generische Webserver unterstützen wie schon in Kapitel 2.4 aufgegriffen das serverseitige Skripting mit beispielsweise PHP (Hypertext Preprocessor) oder anderen Skriptsprachen. Dies bedeutet letztlich, dass das Verhalten des Webserver mittels separater Skripte erweitert beziehungsweise definiert werden kann, während die tatsächliche Serversoftware komplett unberührt bleibt. Die dynamische Generierung von Webseiteninhalten wird überwiegend zum Abrufen oder Ändern von Informationen aus Datenbanken verwendet.

Die aktuell am weitesten verbreiteten Webserver nach (w3techs.com, 2020) sind der Apache HTTP Server und der Nginx Webserver.

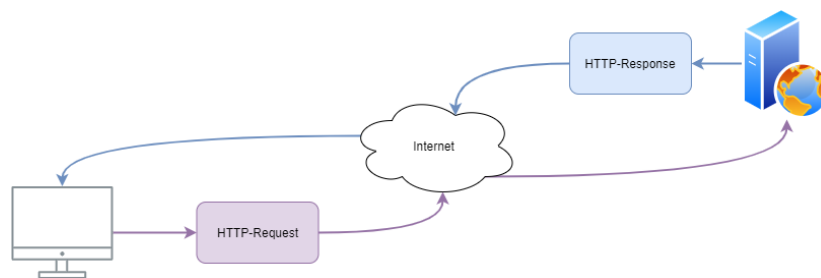


Abbildung 8: Client-Server Modell

2.7 Webtechnologien zur Visualisierung (Überdenken ob passend)

Der Sinn einer Visualisierung ist, dass diese auch gesehen wird und der schnellste und einfachste Weg eine Visualisierung für mehrere Menschen zugänglich zu machen ist über eine Webseite. Ein Webbrowser hat laut (Murray, 2017) den Vorteil, dass er nicht an ein Betriebssystem wie beispielsweise Windows, Mac OS, Linux, Android oder iOS gebunden ist, sondern für die Anwendung nur einen Internetzugang und einen Browser benötigt. Ein weiterer Vorteil ist, dass diese Variante kaum Lizenz- und Schulungskosten mit sich bringt. Jedoch muss hierbei auch die Benutzerfreundlichkeit im Fokus liegen, da der Anwender nicht zwingend technisch begabt ist, sollte die Webansicht einfach, verständlich und intuitiv sein. Laut (Wirtschaftslexikon Onpulsion) müssen, um die Benutzerfreundlichkeit und die Erwartungen der Zielgruppe aufeinander abzustimmen, die Usability-Kriterien Effektivität, Effizienz und Zufriedenheit mit dem Content, Design und der Struktur der Webseite harmonisieren. Mit Effektivität ist hier gemeint, dass der Anwender auf dem richtigen Weg ist, um etwas zu finden, das bedeutet, dass eine Seite so aufgebaut sein muss, dass man sich einfach zurechtfindet und auch zu jeder Zeit weiß wo man sich aktuell befindet. Effizient steht hier für die Schnelligkeit. Der Nutzer möchte schnell sein gewünschtes Ziel erreichen können. Und die Zufriedenheit sagt aus, ob sich der Anwender in einem für ihn ästhetischem Umfeld befindet also wie ihn die Seite optisch anspricht.

Dazu gibt es auch kommerzielle Tools für eine Visualisierung von Graphen. Diese sind z. B. KeyLines und Linkurious. Diese bieten zwar einige Analysemöglichkeiten, jedoch werden diese nicht weiter in Betracht gezogen, da das Unternehmen ISB AG eine kostengünstige Lösung anstrebt. Aus diesem Grund wird eine Visualisierung in Form einer Webansicht selbst erstellt.

d3js

3 Konzeption

Dieses Kapitel befasst sich mit den theoretischen Überlegungen, welche in dieser Arbeit erarbeitet wurden. Konzepte bezüglich der Infrastruktur, Datenmodell, Entwicklungsumgebung und Visualisierung werden thematisiert. Es bildet somit die theoretische Basis, auf welcher der praktische Teil dieser Arbeit aufsetzt. Dieser wird in Kapitel 4 näher betrachtet.

3.1 Infrastrukturaufbau

Die folgende Grafik zeigt den Aufbau der Infrastruktur für diese Arbeit. Als Server kann hier beispielsweise ein Windows Server verwendet werden. Innerhalb dieses Servers befinden sich die Datenbank und der Webserver. Als Datenbank wurde hier Neo4j angedacht und als Webserver beispielsweise der Apache HTTP-Server. Der Webserver beinhaltet den entsprechenden Code mit den dazugehörigen Skripten.

Um auf diesen Inhalt zuzugreifen, muss zunächst der User an seinem Computer einen Browser öffnen. Dies kann wie hier abgebildet der Firefox sein. Anschließend wird über das Internet bzw. ein Intranet auf die Webseite zugegriffen.

Die einzelnen Komponenten sind in dieser Grafik nur beispielhaft aufgeführt. Diese können durch vergleichbare Komponenten ersetzt werden. Hier ein kleiner Auszug, welche gängigen Komponenten dies sein könnten:

- Server: Windows, Linux, Unix
- Datenbank: Neo4j, MySQL, PostgreSQL, CouchDB
- Webserver: Apache HTTP-Server, Tomcat, Nginx
- Internet / Intranet
- Browser: Firefox, Google Chrome, Internet Explorer, Apple Safari

Grafik

Abbildung 9: Infrastrukturkonzept

3.2 Datenmodellkonzept

Ein Datenmodell ist bei der Erstellung einer Datenbank sehr wichtig, da dies die einzelnen Verbindungen und Abhängigkeiten abbildet. Die folgende Grafik zeigt ein Beispiel für ein Datenmodell einer abstrahierten IT-Landschaft. Das Datenmodell wurde so erweitert, dass es möglichst realistisch ist und viele eventuellen Szenarien abdecken kann, welche die Entscheidungsfindung unterstützen.

Der Aufbau des Datenmodells ist in drei Spalten gegliedert. Die linke Spalte, welche in der Farbe Blau dargestellt ist, zeigt die Unternehmensstruktur. Bei der mittleren Spalte handelt es sich um das Informationssystem. Dieses ist die Verbindung zwischen der Unternehmensstruktur und der in der Farbe Grün dargestellten rechten Spalte, dem technischen Aspekt. In manchen Fällen kommt es vor, dass der Name der Relation sich wiederholt. Dies ist der Fall, wenn Verbindungen dieselbe Art der Beziehung aufweisen. Beispiele hierfür können der Abbildung 10 entnommen werden. Beziehungen zwischen Elementen innerhalb der Kategorie der Unternehmensstruktur (blau) sowie der Kategorie des technischen Aspekts (grün) sind zulässig. Des Weiteren werden auch Beziehungen innerhalb einer Objektklasse (Knoten) erlaubt. Als Beispiel sei hierfür eine Beziehung zwischen zwei Informationssystemen an dieser Stelle genannt. Diese könnte beispielsweise durch eine implementierte Schnittstelle hervorgerufen werden. Weiterhin sei erwähnt, dass eine Objektklasse sowie eine Relation mit Attributen erweitert werden kann. Dies hängt in der Realität von der entsprechenden Datengrundlage ab.

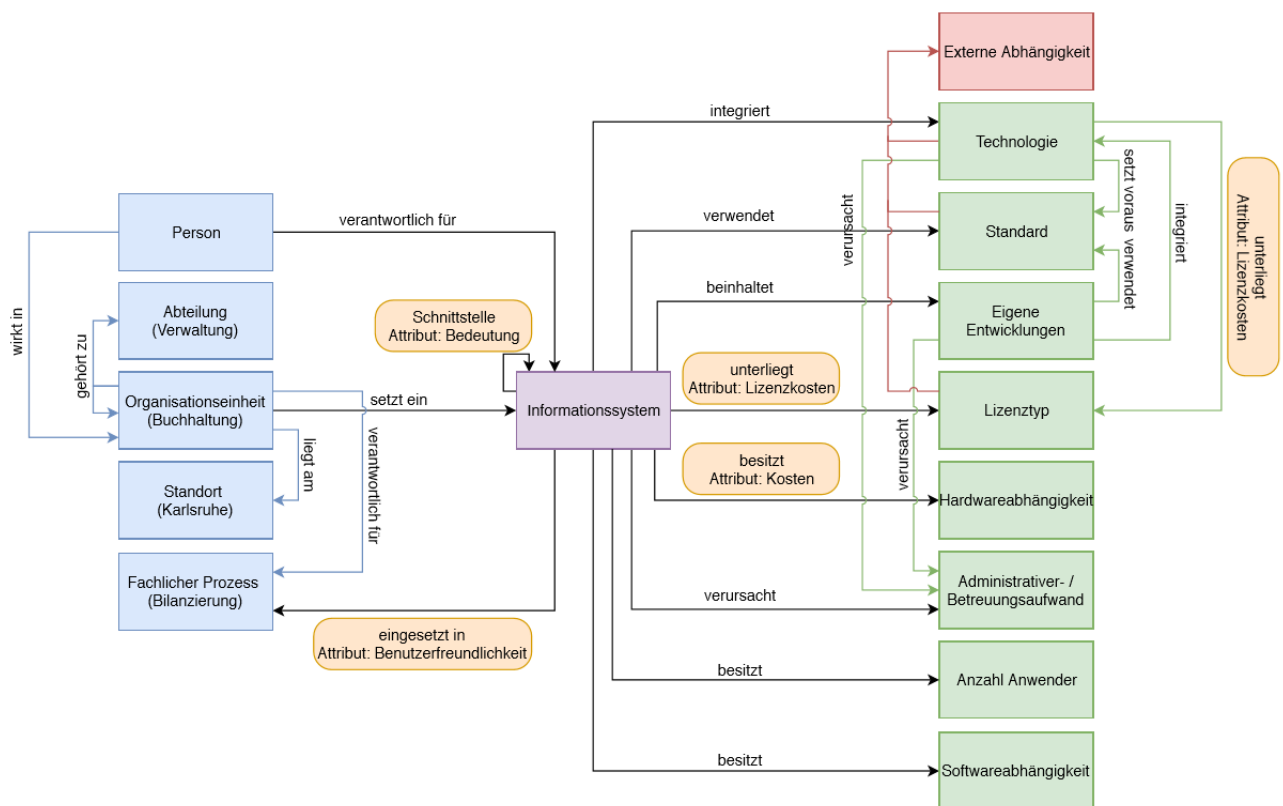


Abbildung 10: Datenmodell

Datenmodelle sind sehr aussagekräftig. Durch das visuelle Verständnis fungiert das Modell als eine Art Kommunikationstool zwischen Entwickler und Endanwender. Wenn man ein konzipiertes Datenmodell sieht, ist es zumeist sehr selbsterklärend und man kann die Zusammenhänge des Modells schnell erkennen, ohne thematisch tiefe Kenntnisse des Projektes zu haben.

Laut (Wirtschaftslexikon Gabler, 2018) beschreibt ein Datenmodell die zu verarbeitenden Daten eines Anwendungsbereichs mittels Grafik. Zudem zeigt dieses die Beziehung, welche die Daten untereinander aufweisen.

Durch ein Datenmodell können Redundanzen aufgedeckt werden. Ebenso wird dadurch ersichtlich, ob die Daten eindeutige Informationen aufzeigen. Unter Berücksichtigung dieser Punkte, können Rückschlüsse über die Datenqualität getroffen werden. Ein qualitativ hochwertiges Datenmodell ist somit ein essenzieller Bestandteil eines Softwareprojekts, welches Datenbanken als Ressource einbindet. Aus diesem Grund ist es wichtig sich vor der Implementierung mit dem Datenmodell auseinander zu setzen. Zusätzlich unterstützt der visuelle Entwurf der Datenstruktur, den Prozess der Implementierung.

3.3 Visualisierungskonzept für Graphen

Eine weitere Best-Practice-Visualisierung ist nach (Hanschke, 2013) zudem der Graph. Ein Graph stellt die Beziehung zwischen mehreren Informationen visuell dar. Diese Verbindungslinien, auch Kanten genannt, können durch verschiedene Farben und Lienen typen beschaffen sein. Außerdem können diese auch beschriftet werden und somit weitere Informationen liefern. Die dargestellten Knoten zeigen Beziehungen und mögliche Abhängigkeiten untereinander auf, welche visuell schneller wahrgenommen werden als durch eine Liste, so (Hanschke, 2013) weiter.

Ein Graph ist im Gegensatz zu der Bebauungsplan Grafik deutlich agiler in der Darstellung. Die Bebauungsplan Grafik hat eine vorgegebene Struktur und lediglich der Inhalt kann flexibel befüllt werden. Beim Graph hingegen ist die gesamte Darstellung sehr anpassungsfähig. Ebenso ist die Cluster Grafik nicht besonders beweglich in ihrer Darstellung. Bei den Ausprägungen der Prozesslandkarte oder dem funktionalen Referenzmodell gibt es eine vordefinierte Darstellung welche flexibel befüllt werden kann aber das Konstrukt an sich ist eher steif. Es ist schwer hier Abhängigkeiten untereinander darstellen zu können, besonders bei größeren Datenmengen. Die Portfolio Grafik ist, wie auch der Graph, sehr aussagekräftig und kann flexibel befüllt werden. Allerdings lassen sich Beziehungen und Anhängigkeiten kaum darstellen und bei einer großen Datenmenge wird dieses Konstrukt schnell unübersichtlich.

Aus diesem Grund wurde zunächst die Visualisierung mittels Graphen gewählt, da dieser einen guten Überblick der Beziehungen und Abhängigkeiten liefert und die Darstellung sehr performant ist. Da jedoch die Darstellung des gesamten Graphen bei einer großen Datenmenge schnell unübersichtlich werden kann und somit für eine Entscheidungsgrundlage nicht dienlich ist, empfiehlt es sich den Graphen in Subgraphen zu unterteilen. Die folgenden Mockups stellen mögliche Subgraphen dar. Beispielhaft wurden vier Mockups erstellt, welche in der Realität vorkommen könnten und somit eine hilfreiche Entscheidungsbasis liefern. In Anbetracht des enormen Umfangs der Arbeit wurden diese lediglich konzipiert und nicht praktisch umgesetzt. Doch zum besseren Verständnis sind diese sehr wichtig, da dadurch das große Potenzial von Graphen besser nachvollziehbar ist.

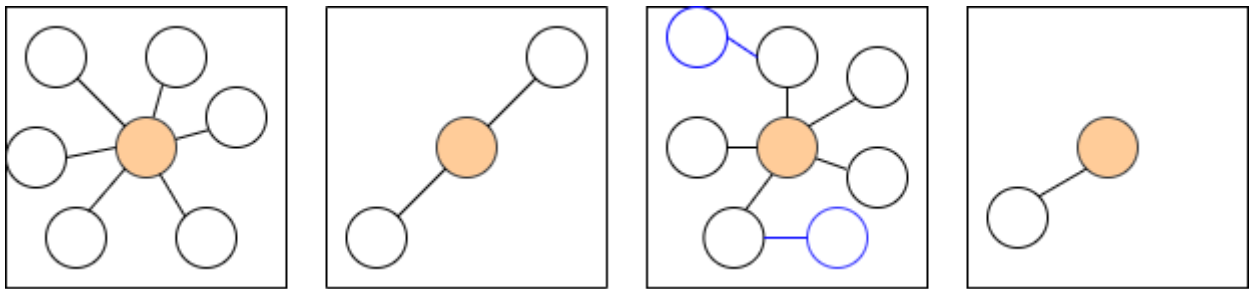


Abbildung 11: Mockup Grad der Vernetzung

Dieses Mockup stellt den Grad der Vernetzung dar. Es zeigt auf, wie viele Beziehungen ein Knoten besitzt. Entsprechend kann dieses Kriterium herangezogen werden, um zu beurteilen, wie stark ein Bestandteil der IT-Landschaft in ein Unternehmen eingebunden ist. Durch diesen Grad der Vernetzung wird schnell ersichtlich, ob ein Knoten einfach eliminiert werden kann oder ob er durch eine hohe Vernetzung ein wichtiger Bestandteil ist.

Der dunkelgraue Knoten könnte beispielsweise ein Informationssystem darstellen, welches mehrere Technologien verwendet. Möchte man nun das Informationssystem durch ein anderes ersetzen, kann man durch solch eine Darstellung die Beziehungen sehen und daraus schließen wie machbar bzw. wie aufwendig diese Veränderung ist. Ein Vorteil durch diese Übersichtlichkeit ist, dass bei einer Änderung von Systemen alle vernetzten Knoten sichtbar sind und folglich keins in Vergessenheit gerät. Dies ist besonders bei sehr komplexen Sachverhalten ein erheblicher Mehrwert.

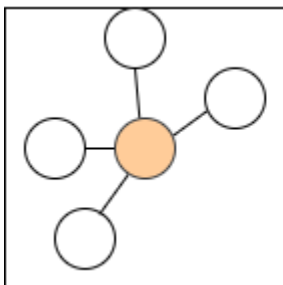


Abbildung 12: Mockup Beziehung zu

Graphen können auch zur Analyse genutzt werden. Das zweite Mockup zeigt den Fall, dass beispielsweise ein Mitarbeiter aus einem Unternehmen ausscheidet, z. B. durch Rente. Daher müssen seine Aufgaben auf andere Mitarbeiter umgelegt werden. Mittels Graphen kann abgebildet werden, für welche Systeme der Mitarbeiter verantwortlich war. Ebenso kann abgefragt werden, ob es Redundanzen gibt. Durch diese Analysemöglichkeit des Graphen kann man sehen, welche Mitarbeiter dieselben Verantwortlichkeiten aufweisen und entsprechend handeln.

Ein weiterer Fall kann hier sein, dass die Technologie, welche hier grau dargestellt ist, veraltet ist. Diese sollte aus Sicherheitsgründen zeitnah ersetzt werden. Die weißen Knoten sind alle Informationssysteme, welche diese Technologie verwenden. Durch diese Abfrageart, wer alles eine Beziehung zur Technologie hat, kann schnell erkannt werden, welche Informationssysteme davon betroffen sind.

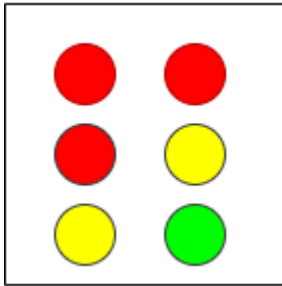


Abbildung 13: Mockup Ampelsystem

Dieses Mockup stellt ein Ampelsystem dar. Es könnte beispielsweise zum Einsatz kommen, wenn es um das Risiko im Bereich von Support geht. Die Knoten stellen hier Technologien dar, welche ein definiertes Datum für ein Supportende aufweisen. Sobald kein Support mehr für die Technologie zur Verfügung steht, entsteht ein großes Sicherheitsrisiko für Unternehmen. Daraus folgt, dass Unternehmen vor Ablauf von Supportenden auf eine neuere Version oder ein anderes Produkt umstellen sollten. Daher zeigt diese Darstellung den aktuellen Stand an.

Rot steht für Datum des Supports ist bereits abgelaufen. Gelb steht für Support läuft in z. B. 200 Tagen ab. Abschließend steht grün dafür, dass der Support noch gewährleistet ist. Durch diese Darstellung kann man schnell einen Überblick über die Lage bekommen, ohne jede Technologie im Detail anzuschauen. Daher ist diese farbige Darstellung eine erhebliche Zeitersparnis, was besonders bei einer großen Anzahl an verschiedenen Systemen sein Vorteil zeigt.

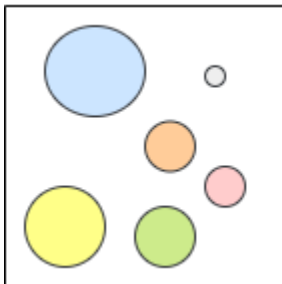


Abbildung 14: Mockup Nutzeranzahl

Das letzte Mockup kann für eine Anzahl der Benutzer eines Informationssystems stehen. Je größer ein Knoten ist, desto mehr Nutzer verwenden ein Informationssystem. Wenn man aufgrund von Kosten ein Informationssystem eliminieren bzw. ersetzen möchte spielt die Anzahl an Nutzer eine große und wichtige Rolle. Verwenden wenige Personen ein Informationssystem ist das Ersetzen von diesem nicht so aufwendig, wie wenn mehrere hundert Personen dieses verwenden. Trotzdem sollte man nicht vor viel benutzten Systemen zurückschrecken. Denn das Eliminieren eines viel benutzten Systems kann schnell Kosten einsparen. Diese Ersparnis kann in Schulungen für Mitarbeiter investiert werden, um auf ein kostenloses oder kostengünstigeres System umzusteigen.

Dieses Mockup kann sehr gut mit anderen gezeigten Mockups kombiniert werden. Beispielsweise bietet sich die Kombination mit dem Grad der Vernetzung oder der Abfrage, zu welchen dieser Knoten eine Beziehung darstellt, an. Dadurch kann sehr gut der reale Umfang dieses Knoten verdeutlicht werden. Wenn ein Knoten durch viele Personen verwendet und ist der Grad der Vernetzung sehr hoch, sollte diese Technologie besonders sorgfältig gepflegt werden.

3.4 Umsetzungsmöglichkeiten für künftige IT-Systementwürfe

Eine künftige Visualisierung kann ein Tech Radar sein. Dieses ordnet die Knoten in einzelne definierte Phasen ein. Dies kann eine gute Entscheidungsgrundlage darstellen, wenn es beispielsweise darum geht welche Systeme Risiken aufweisen oder welche Technologien für neue Projekte verwendet werden sollen. Unternehmen wie Zalando haben bereits eine eigene Visualisierung ihrer Daten im Web veröffentlicht. Dies wurde beispielsweise mittels Technologie Radar umgesetzt, wie Abbildung 15 zeigt. Der Aufbau und die Struktur dieses Technologie Radars wird nach der folgenden Grafik näher beschrieben.

Zalando Tech Radar — 2019.12

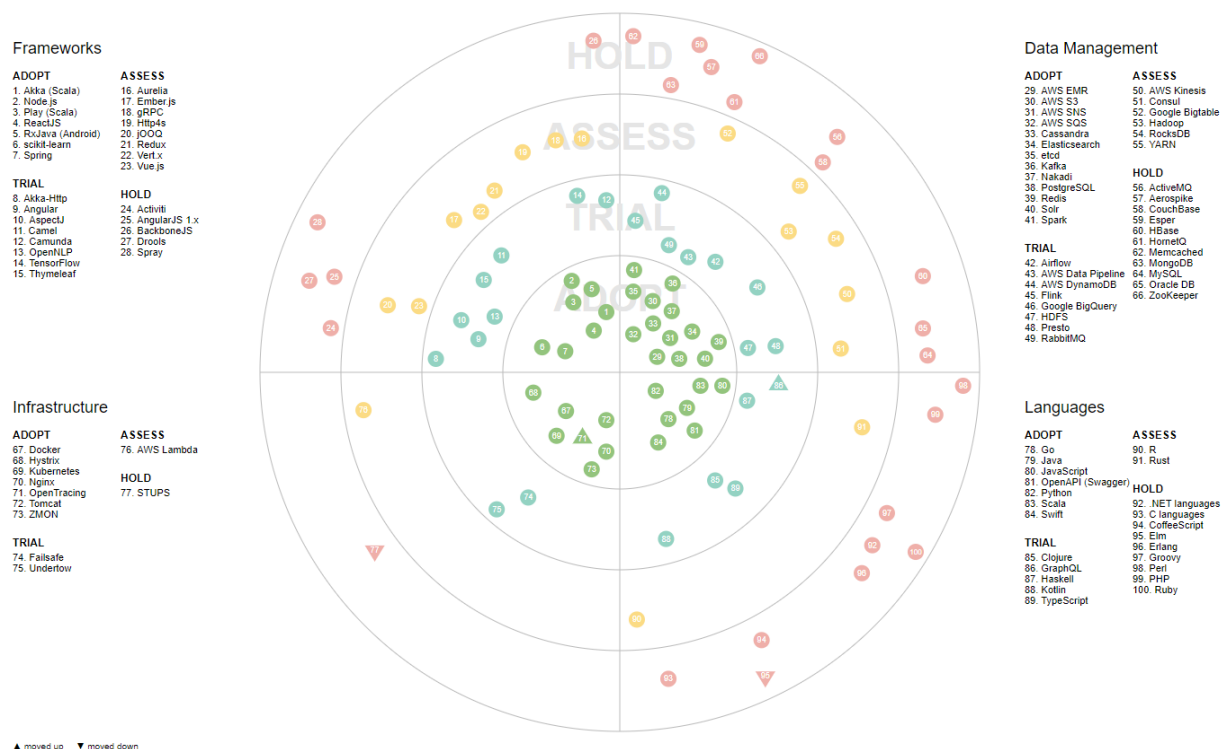


Abbildung 15: Technologie Radar

Dieses Radar kann z. B. wie bei Zalando in folgende Punkte untergliedert werden:

- **Adopt = Übernehmen**

In diesem Feld befinden sich Technologien, auf welche großes Vertrauen gesetzt wird, um langfristig mit diesen zu arbeiten. Ebenso sind diese weitestgehend risikofrei und sind recht verbreitet. Ein Auszug dieser am Beispiel von Zalando sind u.a.:

- Frameworks: Node.js, ReactJS, Spring
- Infrastructure: Docker, Nginx, Tomcat
- Data Management: Cassandra, Kafka, PostgreSQL, Spark
- Languages: Java, JavaScript, Python, Scala

- **Trial = Test**

Technologien werden in diesem Bereich getestet, u. a. in Projekten. Die Technologie hat sich teilweise bereits in der Problemlösung bewährt. Testtechnologien sind riskanter, da sie unvorhersehbare Auswirkungen haben können. Ein Auszug dieser am Beispiel von Zalando sind u.a.:

- Frameworks: Angular, Camunda, Thymeleaf
- Infrastructure: Failsafe, Underlow
- Data Management: Airflow, AWS DynamoDB, Flink, Google BigQuery, Presto
- Languages: GraphQL, Haskell, Kotlin, TypeScript

- **Assess = Bewertung**

Technologien mit Potenzial für das Unternehmen, sowie für die sich Forschung und Prototypen lohnen. Das Risiko ist bei einer Wertung recht hoch, da die Technologien ggf. brandneu und vielversprechend sind, sich aber im Unternehmen noch nicht bewährt haben. Ein Auszug dieser am Beispiel von Zalando sind u.a.:

- Frameworks: Aurelia, Ember.js, Redux, Vert.x, Vue.js
- Infrastructure: AWS Lambda
- Data Management: Consul, Google Bigtable, Hadoop, RocksDB, YARN
- Languages: R, Rust

- **Hold = Halten**

In diesem Bereich befinden sich Technologien, welche nicht für neue Projekte präferiert werden. Diese sind lediglich für aktuell bestehende Projekte notwendig. Daher sind diese Technologien keine weiteren Investitionen mehr wert. Ein Auszug dieser am Beispiel von Zalando sind u.a.:

- Frameworks: Activiti, AngularJS 1.x, BackboneJS, Drools, Spray
- Infrastructure: STUPS
- Data Management: ActiveMQ, CouchBase, Esper, MongoDB, MySQL, Oracle DB
- Languages: .NET languages, Elm, Groovy, Perl, PHP, Ruby

Diese Auflistungen sind beispielhaft von Zalando aufgeführt. Für die Auflistung in dieser Arbeit wurden diese Komponenten gewählt, da sie recht bekannt sind und man sich somit von der Einordnung von Zalando einen besseren Einblick verschaffen kann. Besonders aufgefallen ist im Bereich des Frameworks, dass das vorherige AngularJS 1.x noch im roten Bereich gehalten werden muss und die neuere Version Angular sich bereits im Testbereich des blauen Bereichs befindet.

In der Kategorie der Infrastructure ist aufgefallen, dass sich fast nur Technologien im grünen Bereich befinden. In der Testphase befinden sich nur zwei und in der Bewertungs- und Haltephase lediglich eine Technologie. Das erweckt den Eindruck, dass Zalando im Bereich der Infrastructure auf dem aktuellen Stand ist was langfristig zuverlässige Technologien angeht.

Ganz im Gegensatz zum Data Management. In dieser Kategorie sind die Bereiche sehr ähnlich stark befüllt. Vergleichbar ist es auch in der Kategorie der Languages. In diesen beiden Kategorien herrscht gelegentlicher Wechsel was neue Technologien angeht, im Vergleich zum Framework und der Infrastructure. Diese sind eher konstant da eine Änderung hierbei deutlich aufwendiger umzusetzen ist wie beispielsweise der Umstieg in eine andere Sprache.

Zudem sind innerhalb der visuellen Darstellung der Kategorien die Veränderungen der einzelnen Punkte mit Dreiecken versehen, was bedeutet ob sie auf- oder abgestiegen sind. So ist eine Änderung des Bereichs sehr schnell zu erkennen. Anhand des Beispiels von Zalando ist dies in der Kategorie Infrastructure die Technologie STUPS. Diese befindet sich nun in der Haltephase, jedoch wird hier nicht ersichtlich aus welchem Bereich diese Technologie in die rote Phase gekommen ist. Dies ist auch in der Kategorie Languages zu erkennen. Einmal ist die Technologie GraphQL in die Testphase gekommen und die Technologie Elm ist in der Haltephase. Auch hier kann man nur mutmaßen aus welchen Bereichen diese Veränderung kommen kann.

Eine Erstellung solcher Technologien ist in Unternehmen sehr sinnvoll und hilft den Mitarbeitern, besonders in großen Unternehmen einen Überblick zu behalten. Außerdem ist diese Übersicht sehr hilfreich für die Mitarbeiter, um für neue Projekte die besten Technologien auszuwählen zu können. Ebenso zeigt es Veränderungen von Technologien an und die Mitarbeiter können sich so schnell und selbstständig auf den neusten Stand bringen.

Durch diese Visualisierung kann erkannt werden, welche Produkte das Unternehmen besitzt bzw. verwendet oder welche für bestehende Projekte notwendig sind und nicht ohne weiteres eliminiert werden dürfen. Abhängig davon was man aussagen möchte ist diese Visualisierung ggf. geeignet. Bei der vorliegenden Datenbasis macht diese Visualisierung nur Sinn, wenn man nicht alle Daten abbildet. Wenn es um eine rein technische Konsolidierung der Daten geht, sprich die Informationssysteme und Technologien nur im Fokus stehen, wäre dieses Radar eher geeignet.

3.5 Konzeption der Entwicklungsumgebung

Bei der Erstellung des Konzeptes für die genutzte Entwicklungsumgebung, war neben den technischen Anforderungen dieser Arbeit auch die Nachhaltigkeit, die einfache Verwaltung wie auch die einfache Portierung des Systems auf andere Geräte von Bedeutung. In der Abbildung 16 sind die einzelnen Komponenten der Entwicklungsumgebung und deren Verknüpfungen untereinander visualisiert.

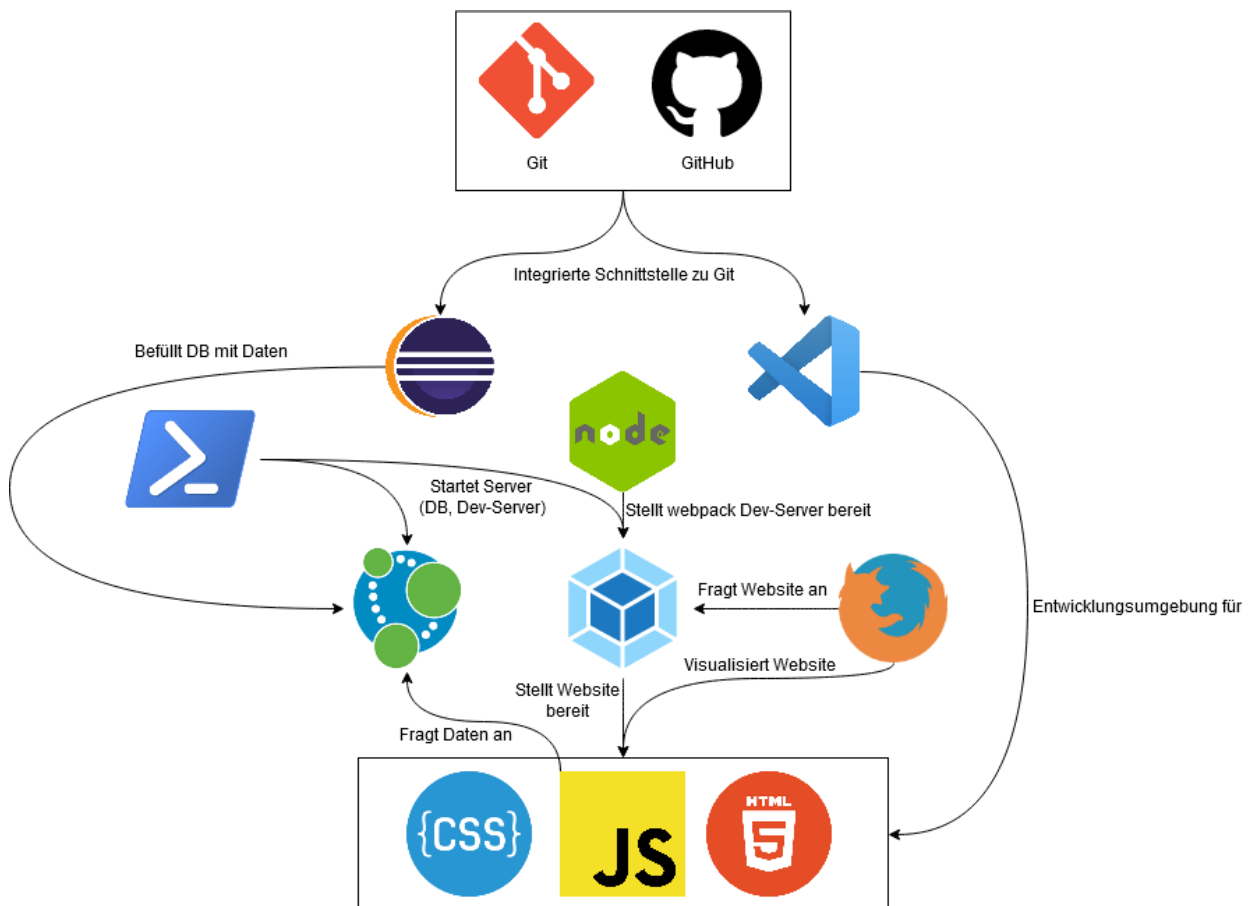


Abbildung 16: Konzept der Entwicklungsumgebung

Als zentraler Ablageort für den Code wurde GitHub verwendet. Hinter GitHub steckt das Versionsverwaltungssystem Git, welches sich auch in der Namensgebung von GitHub wiederfindet. Die Wahl fiel auf Git, da es diverse Schnittstellen zu populären integrierten Entwicklungsumgebungen (IDE) bietet und zudem den aktuellen Standard im Bereich der Versionsverwaltungssysteme definiert. Der Einsatz eines Repositories innerhalb eines Softwareprojekts bietet neben der Datensicherheit auch noch den Vorteil der einfachen Verteilung des Systems auf andere Geräte.

Als IDE wurde Visual Studio Code für die Entwicklung für CSS, JavaScript und HTML5 und für die Entwicklung für Java wurde Eclipse gewählt. Visual Studio Code zeichnet sich durch eine sehr schlanke IDE aus, welche modular durch zahlreiche Erweiterungen ergänzt werden kann. Erweiterungen sind für diverse Sprachen vorhanden. Eclipse ist eine sehr weitverbreitete IDE für Java und aufgrund ihrer langjährigen Historie sehr bewährt.

Der Code des Softwareprojekts wird mittels eines Webservers bereitgestellt und über einen Browser visualisiert. Der Webserver wird durch die Paketerweiterung `webpack` zur Verfügung gestellt, welche wiederum aus der Laufzeitumgebung `NodeJS` stammt. Die Installation von `webpack` erfolgt über den in `NodeJS` integrierten `Node Package Manager`. Dieser bietet auch die Option in einer zentralen Konfigurationsdatei (`package.json`) Abhängigkeiten des Softwareprojekts zu anderen Paketerweiterungen zu definieren. Diese können dann wiederum mittels des `Node Package Managers` und der Konfigurationsdatei auf einem anderen System automatisiert installiert werden. Entsprechend trägt der `Node Package Manager` erheblich zur einfachen Verteilung des Systems auf anderen Geräten bei.

Die eigentliche Hauptaufgabe von `webpack` liegt jedoch nicht darin einen Webserver für die Entwicklung bereitzustellen, sondern in dem Bereich des Deployments. Ist ein lauffähiger Zustand des Softwareprojekts erreicht, kann mittels `webpack` ein Build erzeugt werden. Hierbei extrahiert `webpack` den gesamten projektrelevanten Quellcode, Ressourcen und Stylesheets und führt dies in entsprechenden Dateien zusammen. Dies führt beispielsweise dazu, dass eine komplette Bibliotheksdatei von mehreren Megabyte in Abhängigkeit, des verwendeten Quellcodes auf mehrere Kilobyte reduziert werden kann. Der Build kann somit mit einer Art Standaloneimplementierung verglichen werden.

Als abschließende Komponente der Entwicklungsumgebung sei an dieser Stelle der Datenbankserver von `Neo4J` erwähnt. Dieser ist für die Datenspeicherung und Datenbereitstellung verantwortlich. Er kommuniziert direkt mit der Website unter Zuhilfenahme spezieller Datenbanktreiber für `JavaScript`. Auf eine Middleware zwischen Datenbank und Website wurde bewusst verzichtet, da es das Gesamtsystem spürbar komplexer gemacht hätte, jedoch im aktuellen Zustand des Prototyps keinen erheblichen Mehrwert liefert. Das System kann jedoch ohne größeren Aufwand aufgrund des modularen Aufbaus zu einem späteren Zeitpunkt um eine Middleware nachgerüstet werden.

Eine Java Entwicklungsumgebung ist für Programmierer sehr von Vorteil, da es einem das Programmieren deutlich erleichtern kann. Das Syntax-Highlighting ist besonders hilfreich, da es einem schnell einen Überblick verschaffen kann, in welchem Bereich sich der Code wiederholt. Aber auch die Codevervollständigung ist ein erheblicher Vorteil, da es den Zeitaufwand reduziert.

Die Entwicklungsumgebung in dieser Arbeit setzt sich aus mehreren Bestandteilen zusammen. Für die Programmierung wurde das Programm Visual Studio Code verwendet. Innerhalb dieses Programms wurde mit der Sprache Java, JavaScript, HTML und CSS programmiert. Innerhalb von Visual Studio Code wurde die JavaScript Bibliothek D3.js verwendet. Laut (Webseite D3.js, 2019) ist diese besonders hilfreich im Umgang mit Daten in der Kombination mit HTML, da sie leistungsstarke Visualisierungskomponenten und einen datengesteuerten Ansatz kombiniert.¹

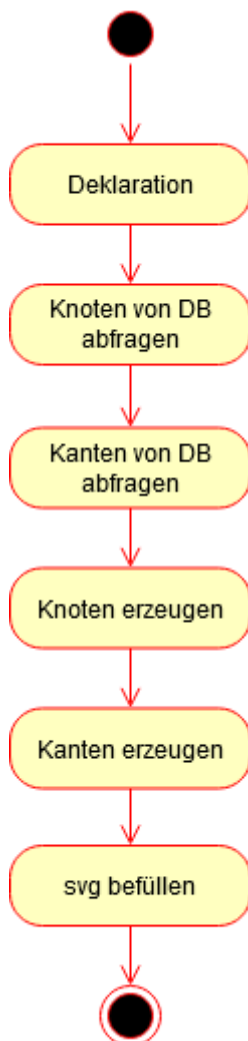
Um die Datenbank anzubinden wurde in Visual Studio Code eine Verknüpfung für Neo4j eingearbeitet. Allerdings wurde ein zusätzliches Programm benötigt, um die Daten in die Datenbank laden zu können. Das wurde in Eclipse realisiert. Dies war notwendig, da die Datenbank nur mit Cypher angesprochen werden kann. Aus diesem Grund mussten die Daten zunächst umgewandelt werden. Da dies manuell umzuwandeln ein zu großer Aufwand - und langfristig auch nicht effizient wäre, wurde in Eclipse ein Programm geschrieben, welches dies umwandelt. Nachdem der Output generiert wurde, war eine Ansicht im Programm Notepad++ erforderlich, da durch Sonderzeichen der Daten dies zu Fehler in der Darstellung führen kann und dies anhand der Datenmenge nicht schnell innerhalb der Datenbank ersichtlich wird. Die Ansicht in Notepad++ war somit eine zusätzliche Kontrolle zur Fehlervermeidung.

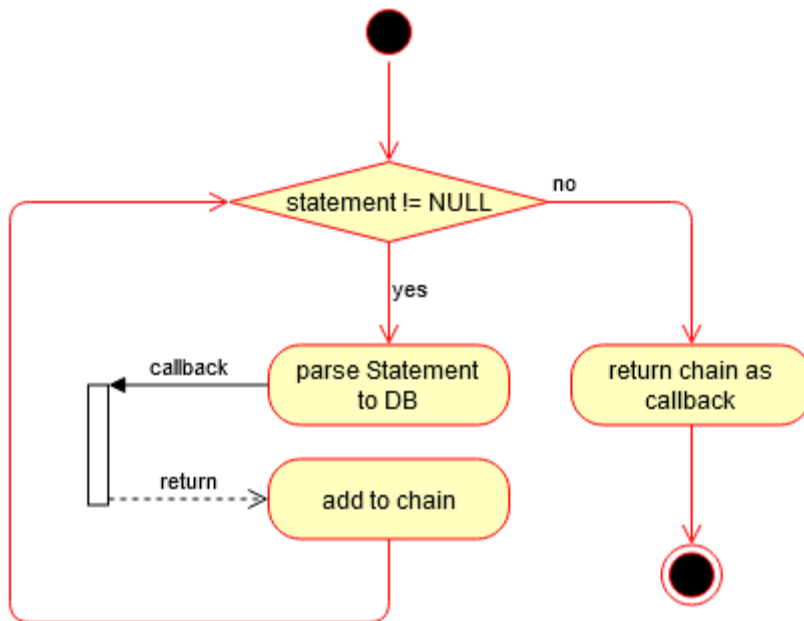
Die Darstellung in der Web-Ansicht erfolgte hier über den Browser Google Chrome, ebenso sollte dies auch bei einem anderen Browser konform funktionieren. Bei der von der ISB AG zur Verfügung gestellten Hardware, handelt es sich um einen Laptop mit dem Betriebssystem Windows 10 Enterprise.

¹ <https://d3js.org/>

4 Implementierung

In diesem Kapitel wird gezeigt, was für das Vorgehen bei der Umsetzung der technischen Seite der Arbeit erforderlich ist. Dieser Punkt ist bei einer möglichen Nachbildung der Arbeit sehr wichtig. Hier kann man nochmal genau nachvollziehen was für eine Umsetzung zwingend notwendig ist und wie das Ergebnis entsprechend aussehen soll.





4.1 Verwendete Technologien

Um die in der Arbeit verwendeten Technologien in ihrer Gänze bereitzustellen, sind diese Tools nachfolgend aufgelistet und mit den entsprechend verwendeten Versionsnummern versehen:

- Betriebssystem: Windows 10 Enterprise
- D3.js Version 5
- Eclipse Java 2019-09
- GitHub Desktop Version 2.2.4
- Google Chrome Version 79.0.3945.130 (64-Bit)
- Java™ SE Development Kit 13.0.1.0 (64-Bit)
- Microsoft Visual Studio Code (User) Version 1.41.1
- Neo4j Desktop Version 1.2.1
- Node.js Version 10.16.3
- Notepad++ Version 6.5.4
- XAMPP Version 7.3.9-0

Einbauen: jquery, PowerShell, Webpack?

Eine Anforderung an den Anwender ist, dass ein aktueller Browser verwendet werden sollte. Mit einer veralteten Version kann es zu Schwierigkeiten in der Darstellung kommen. Zudem wird eine stabile Internetverbindung grundsätzlich vorausgesetzt.

4.2 Vorgehen bei der Visualisierung

Es war zu Beginn wichtig, die in die Datenbank eingespielten Daten, direkt zu Visualisieren. Dadurch konnten Unstimmigkeiten oder Verbesserungsmöglichkeit schneller festgestellt werden. Ebenso konnte dadurch abgeschätzt werden, ob diese Visualisierung eine Entscheidungsgrundlage bieten kann oder ob es zu unübersichtlich anhand der Datenmenge ist. Zum einen wurden die Daten innerhalb der Datenbank Neo4j visuell dargestellt und zum anderen auf der programmierten Webansicht im Browser.

Zudem wurde bei der Visualisierung der Daten festgestellt, dass einige Knoten sehr dominant wirken und die ganze Aufmerksamkeit auf sich ziehen, obwohl diese nicht relevanter sind als andere Knoten. Daher wurden die Farben der Knoten harmonisch angepasst, im Sinne von Farbintensität. Je heller und kontrastierender, desto stärker werden einzelne Farben hervorgehoben.

4.3 Interpretation und Ergebnisse

Die vorliegende Datenmenge ist restlos in die Datenbank eingebunden. Anhand der Datenvisualisierung konnte die Darstellung weiter optimiert werden. Eine Erkenntnis war, dass bei einer großen Datenmenge, welche viele Abhängigkeiten und Beziehungen aufweist, eine Visualisierung in Form eines Graphen, eine Herausforderung war. Die Herausforderung war hierbei, dies gut sichtbar darzustellen, ohne die Datenmenge zu reduzieren. Ein Graph ist für eine umfangreiche Datenmenge konzipiert und bleibt zu jeder Zeit sehr performant.

Bei der Visualisierung der Daten hat sich gezeigt, dass man bei einer hohen Datenmenge schnell den Überblick verliert. Durch Veränderungen der Darstellung, wie beispielsweise wurde die Anzahl der Anwender nicht mehr als eigener Knoten dargestellt, sondern haben diese Informationen die Größe der Informationssystem-Knoten verändert. Somit wurden einige Knoten aus der Darstellung entfernt und die Größe der Knoten gibt Auskunft über die Anzahl der Anwender. Als Beispiel auf der Datenbasis bedeutet dies, wie viele Personen ein Informationssystem verwenden. Je größer der Knoten, desto mehr Anwender gibt es bei diesem Informationssystem. Auch die Farbgebung der Knoten hat sich erst bei der Visualisierung gezeigt, welche Farben dominanter wirken als anderen. Dementsprechend wurden hier auch Änderungen vorgenommen.

→ Gehört zu Kapitel 4 Implementierung

Das Internet kann durch ein Intranet ersetzt werden, wenn es sich beispielsweise um eine geschlossene Anwendung handelt. Das Intranet ist ein vom Internet unabhängiges Rechnernetz, welches nicht öffentlich zugänglich ist.

Bei solch einem Projekt ist ein guter Infrastrukturaufbau essenziell. Aus diesem Grund wurde hier zu Beginn das Tool GitHub installiert, um dieses einerseits als Repository, also als Datenspeicherungstool, zu verwenden. Zum anderen war dieses Tool sehr gut, um von mehreren Geräten am selben Projekt arbeiten zu können. Das Besondere an GitHub ist auch, dass im Falle eines erheblichen Fehlers auf eine vorherige Version zurückgegriffen werden kann, da es eine Versionsverwaltung bietet.

Als Datenbank wurde hier das Neo4j verwendet. Neo4j ist eine NoSQL Datenbank. Genauer gesagt eine Graph-Datenbank und unterscheidet sich zu einer Relationalen Datenbank insofern, dass diese die Möglichkeit bietet große Datenmengen, welche stark vernetzt sind, effizient zu analysieren. Dies könnten beispielsweise Routenberechnungen oder Online-Einkäufe sein.² Die Darstellung dieser Datenbank ist so, dass die Neo4j Datenbank aus einer Menge von Knoten besteht, welche die Objekte darstellen. Die Verbindungen zwischen den Knoten sind Kanten. Diese Kanten spiegeln die Beziehungen zwischen den Knoten wider.

Um die Datenbank im Web aufrufen zu können, musste zunächst ein **Webpack gestartet** werden. Hier wurde das Windows PowerShell verwendet. **WARUM** Nachdem das Programm geöffnet wurde, wählt man zunächst den entsprechenden Ordner aus. Die Steuerung ist so, dass man durch den Befehl „cd..“ einen Ordner zurück springt und durch „cd Ordnername“ nach vorne zum entsprechenden Ordner springt. Anschließend startet man den Vorgang durch den Befehl „bin\neo4j console“. Dies wird durch die nachfolgende Grafik nochmal visuell veranschaulicht.

² <https://www.saracus.com/blog/vergleich-graphdatenbank-neo4j-mit-relationalen-datenbanken/>, 2017 → ersetzen Neo 2.0

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\kschem1> cd..
PS C:\Users> cd..
PS C:\> cd .\neo4j-community-3.5.9\
PS C:\neo4j-community-3.5.9> .\bin\neo4j console
2019-12-20 09:13:49.856+0000 WARN You are using an unsupported version of the Java runtime. Please use Oracle(R) Java(TM)
Runtime Environment 8, OpenJDK(TM) 8 or IBM J9.
2019-12-20 09:13:49.866+0000 INFO ===== Neo4j 3.5.9 =====
2019-12-20 09:13:49.874+0000 INFO Starting...
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.bouncycastle.jcajce.provider.drbg.DRBG (file:/C:/neo4j-community-3.5.9/lib/bcp
rov-jdk15on-1.60.jar) to constructor sun.security.provider.Sun()
WARNING: Please consider reporting this to the maintainers of org.bouncycastle.jcajce.provider.drbg.DRBG
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2019-12-20 09:13:53.489+0000 INFO Bolt enabled on 127.0.0.1:7687.
2019-12-20 09:13:55.262+0000 INFO Started.
2019-12-20 09:13:56.421+0000 INFO Remote interface available at http://localhost:7474/

```

Abbildung 17: Windows PowerShell

Anschließend kann im Webbrowser durch *Localhost:7474* die Webseite der Datenbank aufgerufen werden. An der folgenden Grafik ist deutlich zu erkennen, dass die Knoten die Objekte darstellen. Ebenso wird innerhalb der Datenbank der Inhalt dieser Knoten aufgezeigt. Die Verbindung zwischen den Objekten, also die Kanten, haben jeweils einem entsprechenden Namen der Relation, welcher direkt in der Datenbank und auf der Kante angezeigt wird.

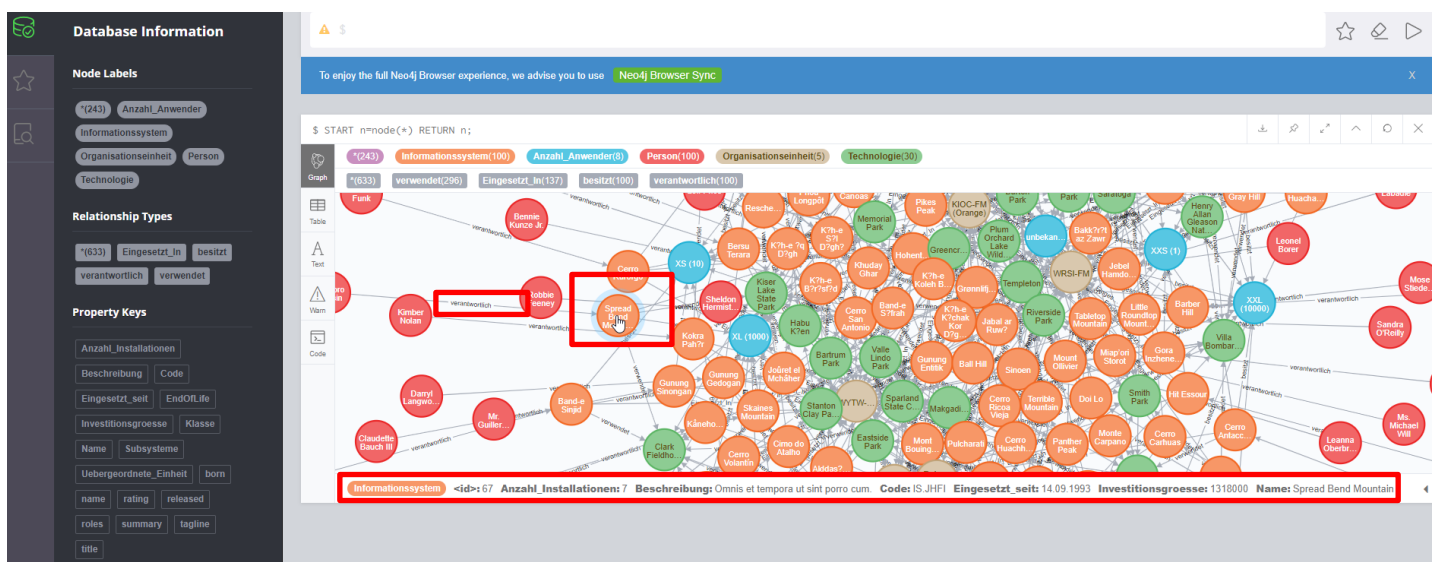


Abbildung 18: Neo4j Datenbankauszug

Besonders von Vorteil ist bei Neo4j, dass innerhalb der Datenbank das Abfrageergebnis direkt visualisiert wird, wie auf der obigen Grafik zu erkennen ist. Somit ist direkt ersichtlich, ob die definierte Abfrage auch das gewünschte

Ergebnis erzielt. Abgefragt wurde hier anhand des Befehls „*START n=node(*) RETURN n,*“. **Durch diese Abfrage gibt die Datenbank ihren gesamten Inhalt aus.**

Für das Tool zum Programmieren wurde sich für Visual Studio Code entschieden. Zunächst standen verschiedene Programme zur Auswahl wie beispielsweise Notepad++ und Sublime Text 3. Jedoch hat sich schnell herausgestellt, dass diese nicht so komfortabel sind wie Visual Studio Code, daher wurde sich für dieses entschieden. Das bei Visual Studio Code die farbliche Zusammengehörigkeit des Codes übersichtlicher dargestellt ist als bei Sublime Text 3, war ein Grund für die Wahl für Visual Studio Code. Diese farbliche Zusammengehörigkeit ist auch als Syntax-Highlighting bekannt. Zudem hat dieser Editor einen zusätzlich integrierten Debugger und läuft mit verschiedenen Betriebssystemen wie z. B. Windows, Linux und Mac OS.³ Ein weiterer Grund für Visual Studio Code ist, dass die Hintergrundfarbe schwarz deutlich angenehmer beim Programmieren ist als die Hintergrundfarbe weiß wie bei Notepad++. Zudem befindet sich in Visual Studio Code rechts im Fenster eine optische Übersicht des Codes in Miniatur. Dadurch erhält man einen kleinen Überblick über die Struktur des Codes. In Abbildung 5 und Abbildung 6 sind diese optischen Unterschiede visuell dargestellt.

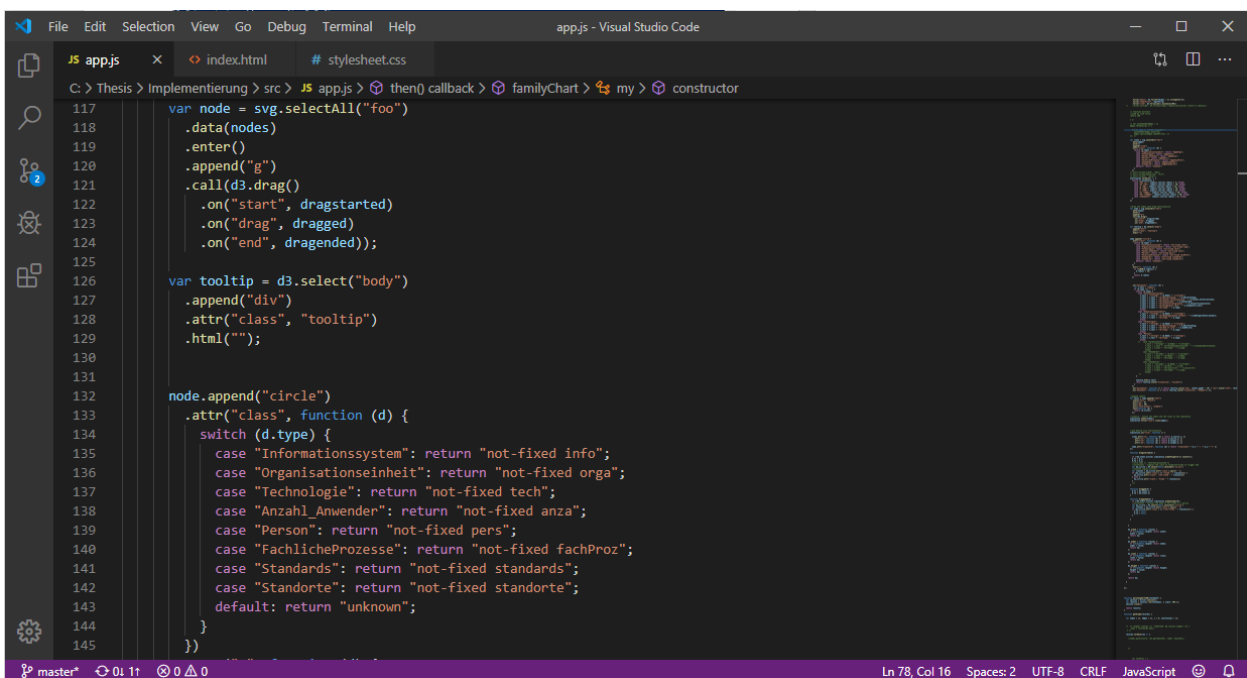
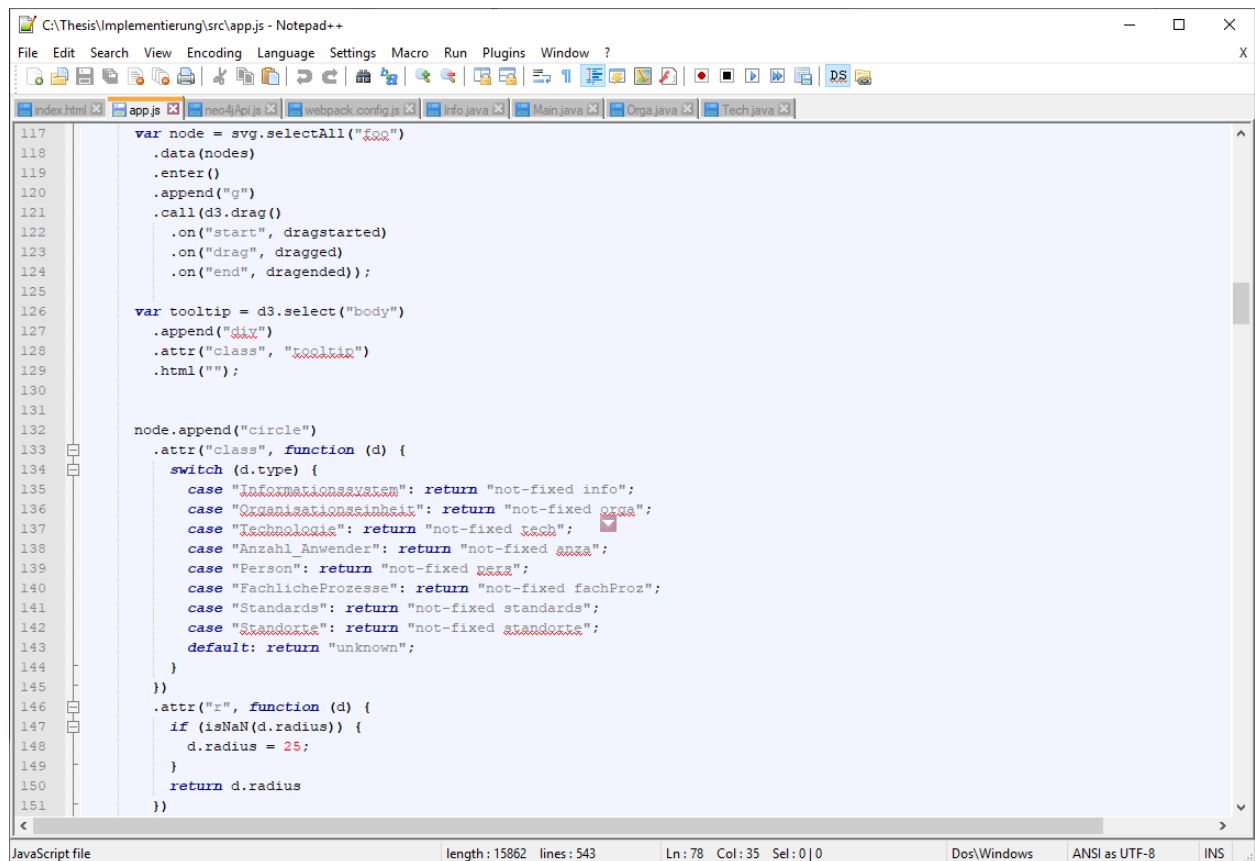


Abbildung 19: Visual Studio Code

³ <https://www.heise.de/newsticker/meldung/Test-Microsoft-Visual-Studio-Code-ein-Editor-mit-integriertem-Debugger-4026553.html>, 2018 → ggf. ersetzen



```

117 var node = svg.selectAll("foo")
118   .data(nodes)
119   .enter()
120   .append("g")
121   .call(d3.drag()
122     .on("start", dragstarted)
123     .on("drag", dragged)
124     .on("end", dragended));
125
126 var tooltip = d3.select("body")
127   .append("div")
128   .attr("class", "tooltip")
129   .html("");
130
131
132 node.append("circle")
133   .attr("class", function (d) {
134     switch (d.type) {
135       case "Informationssystem": return "not-fixed info";
136       case "Organisationseinheit": return "not-fixed orga";
137       case "Technologie": return "not-fixed tech";
138       case "Anzahl_Anwender": return "not-fixed anza";
139       case "Person": return "not-fixed pers";
140       case "FachlicheProzesse": return "not-fixed fachProz";
141       case "Standards": return "not-fixed standards";
142       case "Standorte": return "not-fixed standorte";
143       default: return "unknown";
144     }
145   })
146   .attr("r", function (d) {
147     if (isNaN(d.radius)) {
148       d.radius = 25;
149     }
150     return d.radius
151   })

```

JavaScript file length: 15862 lines: 543 Ln: 78 Col: 35 Sel: 0 | 0 Dos/Windows ANSI as UTF-8 INS

Abbildung 20: Notepad++

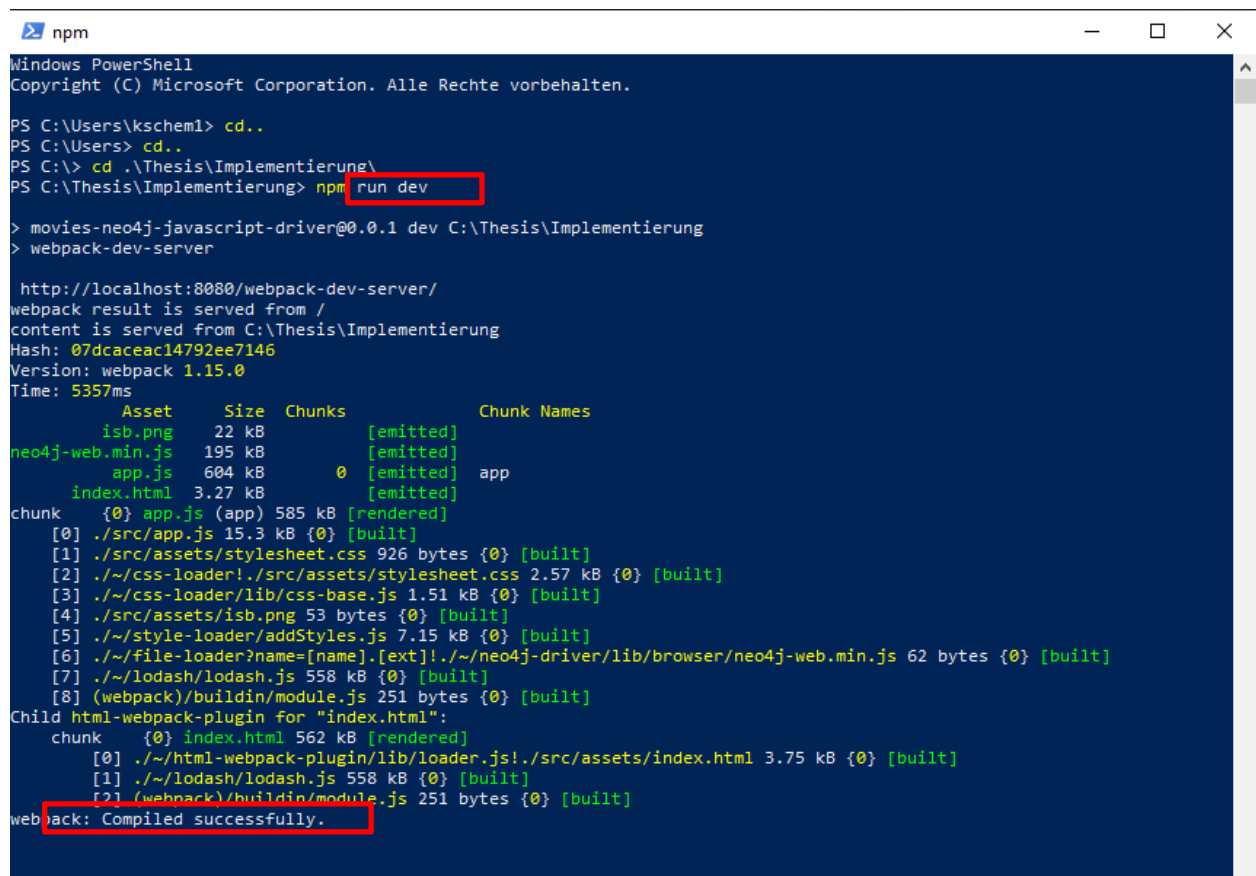
Doch um eine Datenbank sinnvoll befüllen zu können, benötigt man selbstverständlich auch Daten. Die Testdaten wurden hier per Excel-Liste vom Betreuer der ISB AG zur Verfügung gestellt. Die Daten sind aus Datenschutzgründen hier fiktiv aufgeführt. Diese Liste umfasst sechs Arbeitsblätter mit mehreren Spalten und Zeilen Inhalt. Die Arbeitsblätter haben folgende Namen: Fachprozesse, Informationssysteme, Organisationseinheiten, Technologien, Standards und Standorte. Um die Daten jedoch in Datenbank zu bekommen, muss zunächst jedes Arbeitsblatt einzeln als eine csv-Datei exportiert werden.

	A	B	C	D	E	F	G	H	I
	Code	Name	Beschreibung	Verantwortlich	Anzahl Installationen	Anzahl Anwender (Max.)	Subsysteme	Investitionsgröße	Technologien
1	IS_SMWQ	Pehu	Dolorem est expedita quaeat qui esse totam aliquam.	Euna Walsh	6	S (50)		\$412000	TECH.JKWY;TECH.ZLUU;TECH.CILF;TECH.DECH
2	IS_GNLQ	Cerro Jabali	Reiciendis odio tempora nostrum ad ea vel.	Rickey Rogahn MD	10	M (100)		\$540000	TECH.EOTL;TECH.CILF;TECH.DECH
3	IS_RUAO	Mount Allimono	Blanditiis error quis ut culpa repellat est.	Gaston Bauch	5	XS (10)		\$301000	TECH.TXVR;TECH.JKWY;TECH.JPWM;TECH.XSOI
4	IS_VYNA	Küh-e Chap Darreh	Libero optio excepturi natus rerum repellat cum aliquam fugit.	Timothy Smith	7	L (500)		\$1851000	TECH.NOGI;TECH.WNZT;TECH.EOTL
5	IS_QKQO	Mount Pickthorne	Rerum districtio numquam sed maiores consectetur repudiandae.	Elliott Auer	5	M (100)		\$636000	TECH.NOGI;TECH.XSOI
6	IS_LJZV	Küh-e Sar Gachineh	Excepturi quo suscipit aperiam facilis officia suscipit maiores qui.	Elliott Auer	10	L (500)		\$77000	TECH.GPPC;TECH.CILF;TECH.XSOI
7	IS_URTX	El Collar	Saepe nam quasi id magnam ea optio assumenda.	Elliott Auer	5	S (50)		\$4068000	TECH.XSOI;TECH.WPDR;TECH.CTPD
8	IS_HUWM	Band-e Gandajih	Molestias architecto nam aut eum adipisci nobis labore.	Euna Walsh	5	XS (10)		\$1378000	TECH.TXVR;TECH.JKWY;TECH.DECH
9	IS_UCLZ	Djebel Choumkar	Culpa repellendus porro voluptas deleniti optio quia id.	Ollie Keeling	10	unbekannt		\$724000	TECH.WPDR
10	IS_RUDV	Dolot Pandolan	Possumus corporis culpa labore sint magnam voluptas.	Gaston Bauch	10	S (50)		\$4596000	TECH.XSOI;TECH.DECH;TECH.CTPD
11	IS_GQRL	Rangitane	Assumenda est vero necessitatibus unde porro excepturi eum.	Charlotte Maggio	5	XXS (1)		\$1374000	TECH.GCAQ;TECH.CILF
12	IS_CDLS	Nuuf Usu	Enim enim eveniet saepe voluptatibus at velit.	Timothy Smith	5	S (50)		\$884000	TECH.EOTL;TECH.XSOI;TECH.OJXS;TECH.DECH
13	IS_AAKV	Cerro Pillone	Ratione similique sint et dolores voluptatum deserunt expedita aut tempore.	Ollie Keeling	4	XXS (1)		\$2339000	TECH.XYQE;TECH.JKWY;TECH.NOGI;TECH.WPDR
14	IS_SENP	Gora Onorhek	Ab ea alias consequatur rerum deserunt error sit suscipit.	Timothy Smith	1	L (500)		\$270000	TECH.XYQE;TECH.CTPD
15	IS_BRZV	Nongrak	Ut quo delectus ad ea quo neque.	Euna Walsh	4	XS (10)		\$1869000	TECH.DJIX;TECH.XSOI;TECH.DECH
16	IS_PORS	Küh-e Arreh Khvâr	Voluptatibus itaque magni libero et ut totam voluptatem.	Gaston Bauch	5	XL (1000)		\$2786000	TECH.GCAQ;TECH.QICM;TECH.DECH;TECH.CTPD
17	IS_KKKX	Cerro San Felipe	In et vitae doloribus provident velit nam sunt.	Timothy Smith	1	S (50)		\$26000	TECH.JPWM;TECH.DJIX;TECH.XSOI;TECH.CTPD;TECH.DECH
18	IS_HQNG	Black Hammer Bluff	Ut aliquid et nesciunt nulla commodi sed magni voluptatem illo.	Armadina Grant	4	XXL (10000)		\$1211000	TECH.GCAQ;TECH.CILF
19	IS_WCWC	Cerro Oqueloma	Rerum ipsam fugit dicta assumenda assumenda voluptatum perspicatis.	Euna Walsh	3	XXL (10000)		\$989000	TECH.CILF;TECH.NWNE
20	IS_FJXN	Bukit Palong	Modi aliquam vero neque quo dolores qui tempora rerum voluptatibus.	Ollie Keeling	10	M (100)		\$1749000	TECH.JPWM;TECH.GPPC;TECH.CTPD;TECH.DECH
21	IS_ABLW	Die W-Koppe	Laborum voluptates rerum quos velit officis in modi.	Ollie Keeling	5	M (100)		\$1566000	TECH.JKWY;TECH.EOTL
22	IS_VODQ	Frodin Mount	Ipsum non unde voluptas impedit labore aut.	Elliott Auer	1	S (50)		\$1341000	TECH.XYQE;TECH.NOGI;TECH.JPWM;TECH.OJXS;TECH.WPDR
23	IS_OSVF	Jabal Badr	Corrupti quae quibusdam id eius dignissimos nostrum omnis officia.	Charlotte Maggio	5	M (100)		\$4885000	TECH.NOGI;TECH.XSOI
24	IS_UGJP	Sugarloaf Mountain	Illum sunt velit molestias rerum consectetur est ullam dolore dolorum.	Timothy Smith	8	XXL (10000)		\$2778000	TECH.JKWY;TECH.NOGI;TECH.EOTL;TECH.OJXS
25	IS_ZUPW	Örnekruten	Quaerat aut a vel molestias quia assumenda ullam.	Bret Schinner	5	M (100)		\$158000	TECH.JKWY;TECH.CILF;TECH.XSOI
26	IS_LQZG	Bronc Mule Hill	Aspernatur rerum voluptas et maxime id aut est aspernatur odit.	Timothy Smith	5	XS (10)		\$4404000	TECH.WPDR;TECH.XSOI;TECH.CTPD;TECH.IETR
27	IS_MQFO	Hines Mountain	Accusamus sed tenetur itaque in dignissimos dignissimos a incidunt.	Charlotte Maggio	5	unbekannt		\$1502000	TECH.EOTL;TECH.CILF;TECH.CTPD
28	IS_CKFX	Djebel Sridja	Voluptatem culpa sit quasi quis quia est molestiae numquam.	Ollie Keeling	6	L (500)		\$1586000	TECH.JKWY;TECH.GCAQ;TECH.CILF;TECH.XSOI;TECH.WPDR
29	IS_QESY	Gora Nikitina	Excepturi adipisci molestiae ipsum qui hic enim illo.	Charlotte Maggio	4	XL (1000)		\$218000	TECH.NOGI;TECH.CILF;TECH.DECH
30	IS_XORG	Thyskop	Aut a sapiente est quaerat et totam fuga et omnis.	Timothy Smith	6	XXS (1)		\$4003000	TECH.XYQE;TECH.JPWM;TECH.ZLUU;TECH.GPPC;TECH.OYED;TECH.OJXS
31	IS_CZNP	Cerro Jengibre	Adipisci aliquid quisquam fuga similique hic error molestias.	Timothy Smith	5	unbekannt		\$8599000	TECH.XYQE;TECH.ZLUU;TECH.EOTL;TECH.XSOI
32	IS_BKSH	Jabal Rôb Dâniêb	Odit ex corporis consequuntur nulla harum reiciendis quis laudantium.	Elliott Auer	2	XXL (10000)		\$2255000	TECH.XYQE;TECH.NOGI;TECH.CILF;TECH.DECH
33	IS_WMYM	Küh-e Dasht Khambû	Nihil magnam consequatur nulla voluptatibus ab reiciendis.	Gaston Bauch	9	XL (1000)		\$4729000	TECH.NOGI;TECH.CILF
34	IS_OBBK	Kyrfjelllet	Excepturi dignissimos exercitationem consequatur laborum preferendis cupiditate.	Gaston Bauch	4	L (500)		\$1248000	TECH.NOGI;TECH.EOTL;TECH.DECH
35	IS_XLXC	Biýagäney Tepesi	Sint atque possumus sint pariatut qui nisi labore.	Elliott Auer	6	XXS (1)		\$550000	TECH.NOGI;TECH.MTWP;TECH.XSOI;TECH.CTPD;TECH.IETR
36	IS_RAPL	Hartberg	Dolorum nihil et totam eum nobis deleniti sed facere.	Ollie Keeling	6	XS (10)		\$224000	TECH.JPWM;TECH.WPDR
37	IS_GEYE	Katjälve	Nisi eius qui hic unde exercitationem architecto nobis eveniet.	Rickey Rogahn MD	6	XS (10)		\$1881000	TECH.WPDR;TECH.XSOI;TECH.CTPD
38	IS_TXSE	Küh-e Nâvesar	Aperiam minima maiores aspernatur adipisci incidunt soluta harum sunt.	Elliott Auer	6	XS (10)		\$4262000	TECH.JKWY;TECH.MTWP;TECH.EOTL;TECH.CILF;TECH.IETR
39	IS_EFHB	Lingsetefjellet	Qui quia et vel dolorem hic repellat sint.	Euna Walsh	4	S (50)		\$2465000	TECH.EOTL;TECH.CILF;TECH.DECH;TECH.IETR
40	IS_HWKK	Boz Dağ	Est assumenda explicabo ea corporis sed ducimus pariatut.	Euna Walsh	6	XS (10)		\$1080000	TECH.MTWP;TECH.CILF;TECH.OJXS;TECH.CTPD
41	IS_MMGB	Sex Peak	Accusamus eaque delectus veritatis consectetur nihil ad libero.	Euna Walsh	6	XXL (10000)		\$758000	TECH.NOGI;TECH.WNZT;TECH.XSOI;TECH.DECH;TECH.CTPD
42	IS_MAEF	Bettafjellet	Cupiditate vel beatae id ea culpa officis.	Timothy Smith	7	unbekannt		\$9000000	TECH.JKWY;TECH.CTPD;TECH.IETR
43	IS_TEIC	Küh-e Soljân Zînâl	Eos esse earum sit ex consequatur sint animi voluptatum.	Gaston Bauch	5	S (50)		\$112000	TECH.JKWY;TECH.GPPC;TECH.DECH;TECH.CTPD
44	IS_AVJY	Goudjimo	Eveniet repellendus et et nihil odio dolores aut.	Gaston Bauch	5	L (500)		\$1063000	TECH.XYQE;TECH.NOGI;TECH.EOTL;TECH.QICM
45	IS_YUJK	Riederberg	Aut et perferendis excepturi libero ut est.	Elliott Auer	5	XXS (1)		\$1398000	TECH.NOGI;TECH.JKWY;TECH.KCQO;TECH.ZLUU;TECH.WPDR;TECH.DECH
46	IS_AHTR	FA-Riedel	Pro auctoritate perferendisibus eum cum voluptatibus ut.	Timothy Smith	5	unbekannt		\$578000	TECH.XYQE;TECH.JKWY;TECH.GPPC;TECH.WPDR

Abbildung 21: Excel Daten

Anschließend wird diese Datei in das Programm Eclipse eingelesen. Innerhalb der Import Datei wurde der Code durch die Abfragesprache Cypher angesprochen. Cypher ist die Abfragesprache für die Graph-Datenbank Neo4j. Durch diese Übermittlung des Inhalts konnte eine Output Datei erzeugt werden, welche die Graph-Datenbank aufrufen und auslesen kann. Somit sind die Daten aus der Excel-Datei in die Graph-Datenbank Neo4j gelangt.

Um Einblick in die Daten zu bekommen, wurde ein **Webserver aufgebaut**, welcher eine Webansicht ermöglicht. Die folgende Grafik zeigt das hier verwendete Windows PowerShell. Nach dem Start des Programms, muss der korrekte Ordner ausgewählt werden und der Befehl „`npm run dev`“ löst den Start aus.



```

npm
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\kschem1> cd..
PS C:\Users> cd..
PS C:\> cd .\Thesis\Implementierung\
PS C:\Thesis\Implementierung> npm run dev

> movies-neo4j-javascript-driver@0.0.1 dev C:\Thesis\Implementierung
> webpack-dev-server

  http://localhost:8080/webpack-dev-server/
  webpack result is served from /
  content is served from C:\Thesis\Implementierung
  Hash: 07dcaceac14792ee7146
  Version: webpack 1.15.0
  Time: 5357ms

   Asset      Size  Chunks             Chunk Names
neo4j-web.min.js  195 kB          0  [emitted]
   app.js    604 kB          0  [emitted]  app
  index.html  3.27 kB          0  [emitted]

chunk {0} app.js (app) 585 kB [rendered]
  [0] ./src/app.js 15.3 kB {0} [built]
  [1] ./src/assets/styleSheet.css 926 bytes {0} [built]
  [2] ./~/css-loader!./src/assets/styleSheet.css 2.57 kB {0} [built]
  [3] ./~/css-loader/lib/css-base.js 1.51 kB {0} [built]
  [4] ./src/assets/isb.png 53 bytes {0} [built]
  [5] ./~/style-loader/addStyles.js 7.15 kB {0} [built]
  [6] ./~/file-loader?name=[name].[ext]!./~/neo4j-driver/lib/browser/neo4j-web.min.js 62 bytes {0} [built]
  [7] ./~/lodash/lodash.js 558 kB {0} [built]
  [8] (webpack)/buildin/module.js 251 bytes {0} [built]
Child html-webpack-plugin for "index.html":
  chunk {0} index.html 562 kB [rendered]
    [0] ./~/html-webpack-plugin/lib/loader.js!./src/assets/index.html 3.75 kB {0} [built]
    [1] ./~/lodash/lodash.js 558 kB {0} [built]
    [21] (webpack)/buildin/module.js 251 bytes {0} [built]
webpack: Compiled successfully.
  
```

Abbildung 22: Windows PowerShell

Wenn es gestartet ist, kann durch den Webseitenaufruf *Localhost:8080* auch schon die Webansicht erfolgen. Die folgende Grafik stellt diesen Webaufruf dar. Die Knoten spiegeln die Objekte wider und die Kanten die Relationen. Die Größe der Knoten ist unterschiedlich, da diese aussagt, wie viele Anwender dieses Produkt verwenden, also die Anzahl der Anwender. Je größer der Knoten, desto mehr Anwender. Je kleiner, desto weniger Anwender. Die Farbe der Knoten ist unterschiedlich, da jede Farbe einen anderen Typ abbildet. Die Typen sind entsprechend der Excel Arbeitsblätter abgeleitet. Durch ein Mouse Over Event können die Informationen zu den entsprechenden Knoten abgefragt werden. Ebenso können die Knoten durch einen Klick bewegt und anders positioniert werden.

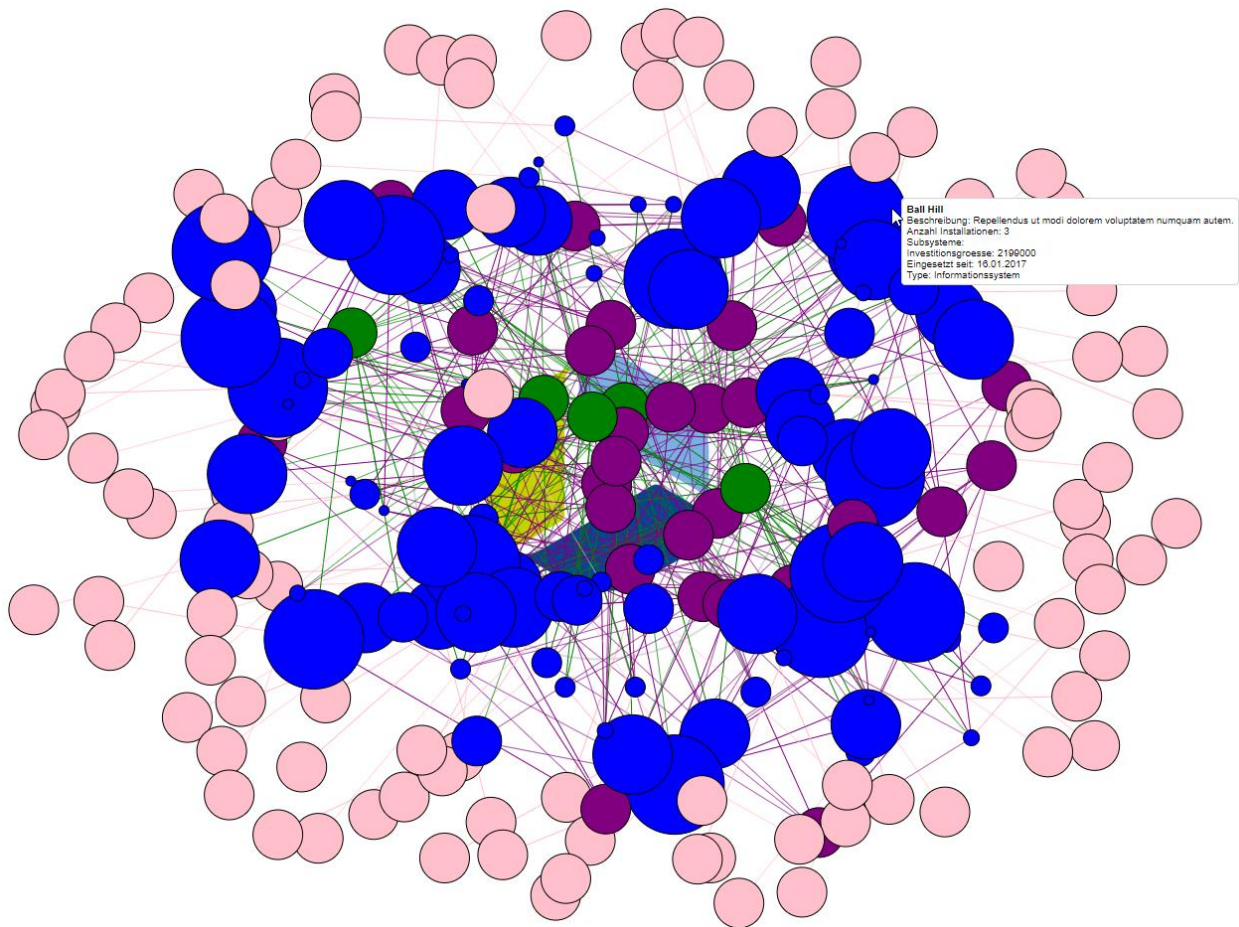


Abbildung 23: Daten Webansicht

Im Zuge des Infrastrukturaufbaus wurde ein „Build“ Ordner angelegt. **WAS IST DAS, WAS MACHT DAS.** Innerhalb dieses Ordners werden automatisiert alle Komponenten abgelegt, welche notwendig sind, um diesen Stand auf einem anderen Gerät zu erzeugen. Daraus folgt, dass keine essenziellen Bestandteile veraltet bzw. unvollständig sind.

5 Evaluation (Fazit)

Bei der Evaluation wird die Arbeit in ihrer Gänze reflektiert. Dabei werden die wichtigsten Ergebnisse prägnant präsentiert und die gesamte Vorgehensweise wird konstruktiv hinterfragt und bewertet.

Der Beginn der Zusammenarbeit mit dem Unternehmen verlief reibungslos. Zunächst wurde diverse Literatur bereitgestellt, um einen Einblick in die Thematik zu erhalten. Bei den wöchentlichen Abstimmungen mit dem Betreuer des Unternehmens wurde auch sehr auf die theoretischen Zusammenhänge eingegangen und durch das Besprechen ein grobes Modell der Datengrundlage erstellt. Mitte Oktober wurden vorläufige Daten erzeugt, mit welchen das Einspielen von Daten in die Datenbank ermöglicht wurde. Die vorab programmierten Zusammenhänge konnten dadurch korrigiert und erweitert werden. Da die Thematik mit Graph Datenbanken kein Bestandteil des Studiums war, war hier sehr viel Einarbeitung und „learning by doing“ notwendig. Die Datenbasis wurde seitens des Betreuers regelmäßig geändert und erweitert, um möglichst praxisnahe Datenbestände zu generieren. Dies hatte zur Folge, dass der Aufbau der Datenbank und des Imports, als auch die optische Darstellung abgeändert und teilweise auch neu strukturiert werden musste.

Im Laufe der praktischen Arbeit sind von Zeit zu Zeit einige Schwierigkeiten im Bereich des Programmierens aufgetreten, da teilweise das Know-How noch nicht ausreichend vorhanden war. Hier hieß es wieder tiefer in die Thematik einlesen, um eine Lösung für das vorliegende Problem zu finden. Zu jeder Zeit stand der Betreuer des Unternehmens mit Rat zur Seite, jedoch war der eigene Ehrgeiz da, es selbst lösen zu können.

Der Betreuer hat regelmäßig an Zeitplan und an die Risikoanalyse erinnert, um nicht zu sehr in Verzug zu geraten. Im Zuge dessen wurde ab Dezember auch mehr Energie in den schriftlichen Teil gelegt. Das sah so aus, dass zu Beginn viel praktisch gearbeitet wurde und an einem Wochentag geschrieben und dies nun umgekehrt wurde - sprich an einem Wochentag praktisch gearbeitet und den Rest der Woche an der Arbeit geschrieben.

Nachdem die aktuelle Version der Datenbestände eingespielt war, musste die optische Darstellung des Graphen angepasst werden. Durch die Anzahl der verschiedenen Typen war die Übersichtlichkeit nicht automatisch gegeben, was eine schlechte Entscheidungsgrundlage für den Kunden bedeuten würde. Daher mussten hier noch diverse Änderungen im Bereich der Visualisierung vorgenommen werden.

Durch das Endergebnis des Graphen ist es nun einem Berater möglich mittels Präsentation dem Kunden durch diese Darstellung einen visuellen Einblick in den Datenbestand zu ermöglichen. Anhand dieser Entscheidungsgrundlage wird es möglich sein eine Entscheidung zwecks IT-Konsolidierung zu treffen.

6 Zusammenfassung und Ausblick

In diesem Kapitel wird die Abschlussarbeit kurz zusammengefasst und legt die wichtigsten Erkenntnisse der Arbeit dar. Der abschließende Ausblick zeigt auf, welches Potenzial noch in der Arbeit steckt und wie diese Arbeit in Zukunft noch weiter ausgebaut werden kann.

6.1 Zusammenfassung

Diese Arbeit ist einerseits in einen konzeptionellen Teil, als auch in einen praktischen Teil untergliedert. Daraus resultiert ein prototypisches Werkzeug zur IT-Konsolidierung. Dieses stellt ein Planungswerkzeug für den Bereich der IT-Konsolidierung dar. Sein Zweck ist es, eine Entscheidungsgrundlage darzustellen. Diese basiert auf einer Datengrundlage, welche zum besseren Verständnis visuell dargestellt wird.

Ein Mensch kann bei einer Anzahl von mehreren 100 Systemen kaum einen Überblick behalten, welche Systeme noch aktuell sind oder welche irrelevant geworden sind. Durch eine visuelle Darstellung kann recht schnell eine Beziehung oder Abhängigkeit zwischen den Daten festgestellt werden und man erhält wieder einen Überblick. Doch von Zeit zu Zeit sollte ein Unternehmen eine IT-Konsolidierung durchführen, um sich von kritischen oder irrelevanten Technologien zu lösen. Durch eine Fusion können Irrelevanzen entstehen und durch veraltete Technologien können Sicherheitsrisiken entstehen.

Genutzt wird dieses Tool voraussichtlich von Beratern und Software Architekten, um den Kunden gewonnene Erkenntnisse visuell zu präsentieren. Anhand dieser Präsentation ist die IT-Konsolidierung für das Unternehmen verständlicher und eine Entscheidung anhand der Visualisierung kann getroffen werden.

Zu Beginn des technischen Teils wurde zunächst eine gut strukturierte Infrastruktur erstellt, welche das Arbeiten erheblich erleichtert hat. Hierbei war besonders hilfreich das verwendete Repository, welches neben einer Versionsverwaltung auch das Arbeiten an verschiedenen Geräten ermöglicht. Während der Einbindung der Daten in die

Datenbank wurde das Datenmodell erstellt und nach Datenupdates aktualisiert und erweitert. Das Datenmodell war während des Programmierens hilfreich, um die Beziehungen und Anhängigkeiten der Daten schnell erkennen zu können.

Während der Implementierung war es notwendig die Daten, welche sich in einer Excel-Datei befinden zunächst mittels Importdatei in eine csv Datei umzuwandeln. Ohne diese Umwandlung ist es der Datenbank nicht möglich die Daten einzulesen.

Innerhalb der Graph Datenbank Neo4j konnte direkt visualisiert werden. Das hatte den Vorteil, dass direkt sichtbar wurde, ob die Datenbestände korrekt in ihrer Art und Menge eingespielt wurden. Die zusätzliche Visualisierung per Webseite hat durch die ansprechende Darstellung eventuelle Probleme oder Unstimmigkeiten in der Optik aufgezeigt. Durch diese beiden visuellen Kontrollen war es möglich gezielte Anpassungen vorzunehmen.

6.2 Ausblick

Bezug auf künftige Visualisierung nehmen

Literaturverzeichnis

Klein, Manfred. 2017. eGovernment Computing. *egovernment-computing.de*. [Online] 13. August 2017.
<https://www.egovernment-computing.de/was-ist-it-konsolidierung-in-der-oeffentlichen-hand-a-741563/>.

Böhm, Jungkunz: *Grundkurs IT-Berufe*. Vieweg-Verlag

Patrick, Killelea (2002). *Web performance tuning* (2nd ed.). Beijing: O'Reilly.
p. 264. [ISBN 059600172X](#). [OCLC 49502686](#).

¹ Inge Hanschke S. 615.

¹ Inge Hanschke S. 144 / 592.

¹ <https://www.bitkom.org/sites/default/files/file/import/EAM-Enterprise-Architecture-Management-BITKOM-Leitfaden.pdf>

¹ <https://www.business-it.link/unternehmen/vision-leitbild-strategie>

¹ Inge Hanschke, S. 144f.

¹ Inge Hanschke, S. 27.

¹ Inge Hanschke S. 238ff.

¹ Inge Hanschke S. 246ff.

¹ BITKOM, Enterprise Architecture Management – neue Disziplin für die ganzheitliche Unternehmensentwicklung, 2011, S.22.

¹ Inge Hanschke S. 241.

¹ Inge Hanschke S. 253.

¹ Neo4j 2.0: Eine Graphdatenbank für alle, entwickler-press, 2014, Paderborn.

¹ <https://blog.codecentric.de/2017/06/graphen-visualisierung-mit-neo4j/>, Tobias Trelle, 2017

¹ NoSQL-Datenbanken, Andreas Maier, Michael Kaufmann, 2016, Springer Vieweg, 8. Auflage, S. 237ff.

¹ Ebook Big Data Insider, Graph-Datenbanken, Michael Matzer.

¹ <https://www.bigdata-insider.de/was-ist-nosql-a-615718/>, Stefan Luber / Nico Litzel, 2017.

¹ Ebook Big Data Insider, Graph-Datenbanken, Michael Matzer.

¹ Ebook Big Data Insider, Graph-Datenbanken, Michael Matzer.

¹ Interactive Data Visualization for the Web S. 3.

¹ <https://www.onpulson.de/lexikon/benutzerfreundlichkeit/> Wirtschaftslexikon.

¹ <https://wirtschaftslexikon.gabler.de/definition/datenmodell-28093/version-251730> , 2018, Richard Lackes,

Markus Sieper

¹ Inge Hanschke, S. 239.

¹ Inge Hanschke, S. 195.

¹ <https://opensource.zalando.com/tech-radar/>

Glossar

Begriff	Definition / Erklärung
Ampel	
Bebauung	
Best Practice	
Enterprise Architecture (EA)	
Enterprise Architecture Management (EAM)	
Fachliches Domänenmodell	
Funktionales Referenzmodell	
Geschäftsarchitektur	
Geschäftsprozess	
Informationssystem	
IT-Governance	
IT-Konsolidierung	
IT-Management	
Organisationseinheit	
Prozesslandkarte	
Schnittstelle	
Strategie	
Unternehmensarchitektur	

Die Erklärungen sind dem Strategisches Management der IT-Landschaft: Ein praktischer Leitfaden für das Enterprise Architecture Management, als auch **XX** entnommen.

Anhang

In der angehängten Daten-CD befinden sich folgende Punkte:

- schriftliche Arbeit
- kompletter Quellcode
- aktuelle Datenbestände