

**Universität Heidelberg Fakultät für Mathematik
und Informatik**

Bachelorarbeit

Mario Bijeljanin
geboren in Bruchsal
Matrikelnummer:4011008
Universität Heidelberg
Angewandte Informatik

Beaufsichtigt von Daniel Schäfer und Professor Dr. Jürgen Hesser

Sommersemester 22
01.07.2022

Definition of the Requirements- and Riskanalysis and implementation of a Relational Database with a Data Ingest Pipeline of a Hospital Triage System

Abstract

When new patients enter the hospital patient intake, they need to be triaged as to assess the urgency of their condition. This results in high priority patients receiving time critical care immediately. This process is currently still done manually at the staff discretion based on personal experience. The TEDIAS (Test and Development Centre for Digital Anamnesis Systems) project aims to make this process more efficient. Under the new system, when patients enter the Mannheim University hospital neurological clinic, they are asked for their personal details and are handed a tablet. Patients fill out a questionnaire from which a rules-based algorithm determines the likely patient priority. The results are visible to the medical practitioners in the form of a dashboard. The purpose of this dashboard is to optimize the efficiency of the neurological clinic by helping the staff manage patient prioritization. Each patient is given a traffic light score which indicates the likely degree of urgency. Medical staff still have a final say and are always in the loop. This helps to decrease wait times for urgent cases, results in patients that will be referred to a non-clinic specialist to be referred out of the clinic earlier and thus increase clinic efficiency and patient satisfaction. This work presents the requirements and risk analysis conducted for this project. Patient User journeys as well as acceptance criteria were established utilizing the IREB standard. In addition, a relational database with an accompanying database schema and data ingest pipeline was developed. This transfers the evaluated patient questionnaires along with their priority flags to a database for long-term analysis.

Definition der Anforderungs- und Risikoanalyse sowie Implementierung einer relationalen Datenbank mit einer Data Ingest Pipeline für ein Krankenhaus-Triage-System

Abstrakt

Wenn neue Patienten in die Patientenaufnahme eines Krankenhauses kommen, müssen sie nach der Dringlichkeit ihres Zustandes eingeteilt werden. Dies führt dazu, dass Patienten mit hoher Priorität sofort eine zeitkritische Versorgung erhalten. Dieser Prozess wird derzeit noch manuell nach dem Ermessen des Personals und auf der Grundlage persönlicher Erfahrungen durchgeführt. Das Projekt TEDIAS (Test- und Entwicklungszentrum für digitale Anamnesesysteme) zielt darauf ab, diesen Prozess effizienter zu gestalten. Nach dem neuen System werden die Patienten beim Eintritt in die Neurologische Abteilung der Universitätsklinik Mannheim nach ihren persönlichen Daten gefragt und erhalten ein Tablet. Die Patienten füllen einen Fragebogen aus, aus dem ein regelbasierter Algorithmus die wahrscheinliche Priorität des Patienten ermittelt. Die Ergebnisse sind für die Ärzte in Form eines Dashboards sichtbar. Ziel dieses Dashboards ist es, die Effizienz der neurologischen Klinik zu optimieren, indem es dem Personal hilft, die Patientenpriorisierung zu verwalten. Jeder Patient erhält einen Ampelwert, der den wahrscheinlichen Grad der Dringlichkeit entspricht. Das medizinische Personal hat nach wie vor das letzte Wort und ist der finale Entscheidungsträger. Dies trägt dazu bei, die Wartezeiten für dringende Fälle zu verkürzen, führt dazu, dass Patienten, die an einen nicht in der Klinik ansässigen Facharzt überwiesen werden sollen, früher aus der Klinik entlassen werden und erhöht somit die Effizienz der Klinik und die Zufriedenheit der Patienten. Die vorliegende Arbeit stellt die für dieses Projekt durchgeführte Anforderungs- und Risikoanalyse vor. Unter Verwendung des IREB-Standards wurden Patient User Journeys sowie Akzeptanzkriterien festgelegt. Darüber hinaus wurde eine relationale Datenbank mit zugehörigem Datenbankschema und Dateningest-Pipeline entwickelt. Diese überträgt die ausgewerteten Patientenfragebögen zusammen mit ihren Prioritätskennzeichen in eine Datenbank zur Langzeitanalyse.

Contents

List of Figures	2
List of Tables	3
1 Einleitung	4
1.1 Motivation	4
2 Theorie and Related works	6
2.1 Requirements	6
2.2 Risikoanalyse	15
2.3 Datenanalyse	18
2.4 Docker	29
2.5 Data Ingestion Pipeline	37
2.6 CAP Theorem	42
2.7 Stand der Dinge in der Digitalisierung im Gesundheitswesen	43
2.8 TEDIAS	48
3 Material und Methoden	50
3.1 Datenbank	56
3.2 Data Ingest	60
3.3 Unit Tests	64
4 Resultate	66
4.1 Anforderungs/Risikoanalyse	66
4.1.1 User Stories	68
4.1.2 Akzeptanzkriterien	71
4.1.3 Systemanforderung	72
4.2 Technical Results	76
5 Diskussion	77
5.1 Requirements- und Risikoanalyse	77
5.1.1 User Stories	77
5.1.2 Akzeptanzkriterien	78
5.1.3 Systemanforderung	79
5.2 Datenbank	79
5.2.1 Datenschutz	80
5.3 Allgemein	80
6 Konklusion	82
7 Ausblick	83
8 Bibliography	85

9 Appendices	90
9.1 Appendix A: User Dokumentation	90
9.2 Appendix B: Ethik in dem Gesundheitswesen	94
9.3 Appendix C: User Patient Journey Überblick	96
10 Ehrenwörtliche Erklärung	97

List of Figures

1	Darstellung der Notaufnahme im TEDIAS Konzept	5
2	Darstellung des Kano Modells	10
3	Darstellung einer Requirements Analyse	12
4	Darstellung der Big Data im Gesundheitswesen	21
5	Darstellung von der Vielfalt der Daten	24
6	Darstellung der Docker Architektur	31
7	Darstellung der Architektur von der virtuellen Maschine	32
8	Container und virtuelle Maschinen zusammen	33
9	Darstellung des CPU Verhaltens in vergleich zwischen Docker und VM . .	36
10	Darstellung des Speicherleistung Verhaltens in vergleich zwischen Docker und VM	36
11	Die Struktur einer Plattform-Pipeline	39
12	Cap Theorem in der Anwendung	43
13	Darstellung der Bitkom Studie die sich mit der Digitalisierung zur Zeiten der Pandemie beschäftigt	46
14	Darstellung der Studie die sich mit der Einführung der Elektronischen Patientenkarte beschäftigt	46
15	Darstellung der B.A.H Studie die sich mit der Einführung der Elektron- ischen Patientenkarte beschäftigt	47
16	Darstellung von 20 Europäischen Ländern zum stand der Implementierung der Elektronischen Patientekarte	49
17	Das Use-Case 2 Dashboard wird über die .NET Core Middleware in die restliche TEDIAS-Landschaft integriert.	50
18	Datenbank Schema	56
19	User Story Journey Patient	68
20	User Story Journey Arzt	69
21	User Story Journey Pfleger und IT-Admin	70
22	TEDIAS Neurologische Notaufnahme Use-Case Patientendashboard. Im- plementierung auf JavaScript Basis bei Max Hirsch ^[71]	73

List of Tables

1 Patient User Journey 96

1 Einleitung

Das aus dieser Arbeit hervorgegangene Projekt TEDIAS hat sich zum Ziel gesetzt, die Patientenregistrierung in der Notaufnahme effizienter zu gestalten. also Wartezeit Minimieren und konzentrieren Sie die Fähigkeiten der Mitarbeiter auf ihre wahren Fähigkeiten. Das Problem in der Notaufnahme ist, dass der Patient auch in die Notaufnahme kommt. die keine dringende Behandlung benötigen und an ihren Haus- oder Facharzt überwiesen werden Muss werden. Manuelle Erfassung von Patientendaten und deren Weg zu Die Kenntnis, dass ein Patient an einen Haus- oder Facharzt überwiesen werden kann, ist bindend Personalressourcen sind nicht erforderlich, da diese Entscheidungen oft anhand einfacher Merkmale getroffen werden Kasten. Die vorgeschlagene Lösung besteht darin, den Prozess unter Personalaufsicht zu digitalisieren, wie etwa in Abbildung 1 zu sehen ist. Patienten, die in die Klinik kommen, finden kleine Räume vor, die Privatsphäre gewährleisten, in dem sie ihre Situation selbstständig in einen Fragebogen auf einem Smartpad eingaben. Daten werden verarbeitet und den Einsatzkräften interaktiv auf dem Bildschirm angezeigt mit einer Empfehlung, ob sie an einen Hausarzt oder einen Facharzt überwiesen werden sollen (grün markiert), wenn das System nichts Offensichtliches findet und der Fall von Mitarbeitern der Notaufnahme untersucht werden muss (gelb markiert), oder wenn der Patient notärztlich versorgt werden muss Erfordert Aufmerksamkeit (rot hervorgehoben). Außerdem medizinische Grundversorgung Kabinendaten werden automatisch erfasst, aber das ist nicht Teil des Jobs. Einer Abbildung 1. Neben der schnellen Filterung von Patienten, die keine Notaufnahme benötigen, reduziert dieser Prozess auch den Personalaufwand für das Ausgeben und Einsammeln von Papierstücken. oder sogar ihre mündlichen Fragen und Einschätzungen werden aufgezeichnet. Daten werden digital und somit zuverlässiger und platzsparender gelagert werden. Der Fragebogen wird verständlicher, weil Bedingungssätze wie „Wenn ja, wie lange haben Sie diese Schmerzen schon?“ nachher Der Rechner kann basierend auf früheren Antworten abgefragt werden. Eventuell mündlich Befragungen finden nicht mehr bei überwiesenen Patienten statt und können viel gründlicher und genauer mit anderen durchgeführt werden, da mehr Zeit und Hintergrundinformationen zur Verfügung stehen. war anwesend. Diese Arbeit befasst sich mit dem, was als nur ein Teil der Bewertung beschrieben wird Dem Personal der Notaufnahme vorlegen. Die Fragen selbst, ihre Logik und ihre Bewertungen wurden von medizinischen Fachleuten gegeben.

1.1 Motivation

Der Zweck des Dashboards besteht darin, die Effizienz einer neurologischen Klinik zu optimieren und dem gesamten medizinischen Personal bei der Verwaltung der Patientenpriorisierung zu helfen, indem Patienteninformationen nach Wichtigkeit gefiltert werden. Dies wird dem Personal helfen, eine gute allgemeine Vorstellung davon zu haben, welche Patienten priorisiert werden sollen, und den besten maßgeschneiderten Behandlungsplan entsprechend zu erstellen. Im Gegenzug werden die Wartezeiten der Patienten minimal sein, ebenso wie die Kosten für das Krankenhaus und die Versicherung. Die Verbesserung würde erleichtert, indem die Vermutungsarbeit aus dem Job der Krankenschwestern her-

ausgenommen wird und die Software einfach die Beinarbeit für sie erledigen kann. Dies gilt insbesondere für neurologische Patienten, die aufgrund ihrer Erkrankung oft nicht in der Lage sind, am besten zu kommunizieren. Die aktuelle Wartezeit des Patienten wird auf dem Dashboard angezeigt. Dies wird dem medizinischen Personal helfen, die Wartezeiten der Patienten in Bezug auf die Dringlichkeit der Symptome besser zu verfolgen. Für die Vervollständigung von dem Projekt wird eine Datenbank gebraucht wo die ganzen Daten gespeichert und gegeben-falls weiterverarbeitet werden. Durch die Speicherung der Daten in einem Datenbankspezifischen Umfeld in diesem fälle MYSQL sind die Daten sicher gespeichert und jederzeit abrufbar und für detailliertere Untersuchungen bereit. Durch die Verfügbarkeit der Daten soll den späteren Daten Analytikern die Arbeit möglich erleichtert werden und alles mögliche gemacht für eine langzeitige wissenschaftliche Auswertung der gesammelten Daten über eine gewählte Zeitangabe. Damit diese Daten möglichst gut verarbeitet werden und für jeden Rechner unter aller Konfigurationen verfügbar ist wurde Docker in diesem Projekt benutzt und wird auf einen Docke Container erstellt um einen möglichst besten bedarf der Software zu sichern, ihre Anwendung und möglichst vielen Menschen zu helfen. Das Projekt muss medizinisch zugelassen werden deshalb wurden alle Medizinischen Richtlinien befolgt und eine möglichst detaillierte Dokumentation erschaffen da es um sehr vertrauliche Daten handeln.

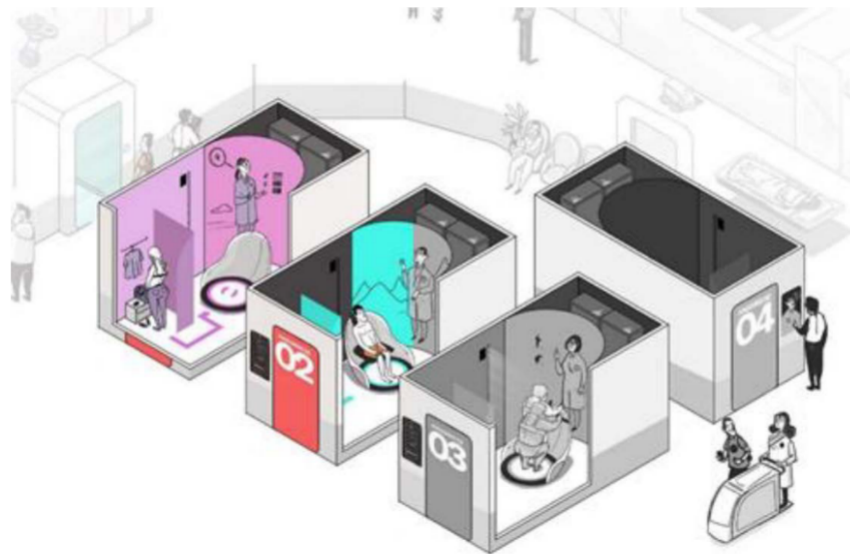


Figure 1: Darstellung der Notaufnahme im TEDIAS Konzept, das Test- und Entwicklungszentrum für digitale Anamnese-Systeme^[1].

2 Theorie and Related works

Es folgt eine Einleitung in die für diese Arbeit erforderlichen theoretischen Grundlagen.

2.1 Requirements

Um eine bestmögliche Realisierung von dem Entwicklungsprojekt zu bekommen/ zu gewährleisten, ist die wichtigste Priorität herauszufinden was sind die Anforderungen an das System, und wie sind die zu dokumentieren.^[6] Einer der wichtigsten Begriffe im Requirements Engineering ist der von dem Stakeholder(Projektbetroffener). Stakeholder repräsentieren wichtige Quellen für die Anforderungen, das Übersehen oder nicht Hineinbeziehung von Stakeholder hat sehr häufig zur Konsequenz, dass die ermittelten Anforderungen an das System fehlerhaft oder lückenhaft sind.^[5] Stakeholder repräsentieren/ beinhalten alle Personen oder Organisationen, die einen Einfluss auf die Gestaltung von Anforderungen haben. Zu den Akteuren gehören die Personen die das System nutzen werden (z.B der Nutzer der Systems oder der Administrator), dazu gehören auch Personen die nicht sicherlich am System teilnehmen aber ein Interesse haben das System zu nutzen, aber es nicht nutzen werden oder sollen(z.B das Management oder ein Hacker von dem das System geschützt werden soll), dazu gehören auch noch juristische Personen, Institutionen, Verbände usw da diese schließlich durch Personen vertreten werden, die Anforderungen von dem zu betrachteten System sind durch alle diese Faktoren beeinflusst bzw die können so definiert werden.^[4] Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Anforderungsspezifikation und -verwaltung mit den folgenden Zielen:

1. Die wichtigsten Anforderungen die damit verbunden sind herausfinden, damit erzielt man einen Konsens der Stakeholder darüber, dokumentieren gemäß festgelegten Standards und Verwaltung der Anforderungen systematisch
2. Die Vorstellungen, Verlangen und Bedürfnisse der Stakeholder einbeziehen und versuchen bestmöglich zu verstehen damit ein gutes Endprodukt entsteht dies noch entsprechen zu dokumentieren
3. Anforderungen sind da um die Wünsche und Bedürfnisse der Stakeholder soweit es möglich ist zu sammeln und die auch zu spezifizieren und zu verwalten und die an das System herüberzubringen.

Requirements Engineering im Entwicklungsprozess haben also die Aufgabe, alle möglichen Anforderungen der Stakeholder zu ermitteln, die auch richtig zu dokumentieren, die auch zu überprüfen, die auch zu priorisieren sowie über den ganzen Verlauf des Projektes hinweg zu verwalten und die Dokumentation zu aktualisieren.^[7]

Die vier damit einhergehenden Haupttätigkeiten sind:

- In dieser Tätigkeit werden verschiedene Techniken genutzt um Anforderungen von den beteiligten Stakeholder und anderer relevanten Quellen zu gewinnen, sie zu detaillieren und gegebenenfalls vereinfachen

- Dokumentieren: In dieser Tätigkeit werden die gewonnenen/ Anforderungen entsprechend beschrieben dafür steht dem Team unterschiedliche Techniken die eingesetzt werden können, um Anforderungen in natürlicher Sprache oder in verschiedenen Modellen zu dokumentieren
- Prüfen und abstimmen: Alle Anforderungen die dokumentiert sind müssen auch frühzeitig geprüft und abgestimmt werden, um sicherzustellen dass alle verlangten Qualitätskriterien garantiert sind
- Verwalten Die Anforderungsverwaltung (Requirements Management) wird gleichzeitig mit allen anderen Aktivitäten und beinhaltet alle Maßnahmen, die essenziell sind, um die Anforderungen zu strukturieren, damit unterschiedliche Rollen die in einbezogen sind die auch nutzen können dazu gehört auch die Anforderungen konsistent zu ändern und (auf dem neusten stand zu halten).

Diese Schlüsseloperationen können unter Verwendung von Verfahren wie denen, die in ISO/IEC/IEEE 291 8:2011 empfohlen werden, sequenziert werden Schlüsseloperationen können für Anforderungen auf verschiedenen Ebenen durchgeführt werden, wie etwa System-, Software- oder Stakeholder-Anforderungen. Da viele Projekte eine unterschiedliche Bedingungen haben deshalb werden das Requirements Engineering jeweils anders beeinflusst. So können z.B die Menschen die an dem Projekt beteiligt sind , fachliche Faktoren wie z.B die Branche für das das Projekt gemacht wird oder organisatorische Randbedingungen (z.B Zeitliche Verfügbarkeit der Projektbeteiligten oder die gesetztes Lage der Branche in der das Projekt gemacht wird)eine große Auswirkung auf die Auswahl der geeigneten Techniken haben und damit auch den Ausgang des Projektes beeinflussen. Die wachsende Bedeutung von Systemen hat einen erheblichen Anteil an Software in Industrieprojekten und die Notwendigkeit innovativere, individualisierte und besser skalierbare Systeme schneller, besser und mit höchster Qualität auf den Markt zu bringen, anspruchsvolles Engineering erfüllt Effizienzanforderungen. Vollständige und fehlerfreie Anforderungen sind die Grundlage für eine erfolgreiche Systementwicklung Einer der ersten Nutzen von Requirements Engineering ist es potenzielle Risiken zu entdecken sie zu identifizieren und gegebenenfalls auch beheben oder eine mögliche Strategie entwickeln um sie zu umgehen damit ein erfolgreichen Projektablauf möglich ist. Fehler und Lücken in den Anforderungen müssen rechtzeitig entdeckt werden um Komplikationen in späteren verlauf des Projekts zu verhindern. In der Requirements-Spezifikation werden nicht alle Anforderungen in einem Projekt gleich detailliert bearbeitet deshalb ist es notwendig für alle Granularitätsstufen entsprechenden Techniken der Spezifikationen bereitzustellen. Epics sind eine Möglichkeit der Spezifikation für noch nicht ganz beschriebene oder ganz grob beschriebene und dokumentierte Anforderungen. Ihre Nutzen ist auch darin mögliche Ideen und Themen festzuhalten, von denen das Team noch nicht genau weiß, ob diese auch in einem Zeitpunkt des Projektes auch umgesetzt werden da es sehr häufig ist das im Projekt verlauf oft Pläne geändert werden.

[6]

Arten von Anforderungen

Im Allgemeinen unterscheidet man zwischen drei Arten von Anforderungen:

- Funktionale Anforderungen sind dazu da um die Funktionalität genau festzulegen, die das geplante System unterschätzen. Sie werden typischerweise in Funktions-, Verhaltens- und Strukturanforderungen unterteilt
- Qualitätsanforderungen definieren die gewünschten Eigenschaften des zu entwickelnden Systems und beeinflussen die Gestaltung der Systemarchitektur oft stärker als funktionale Anforderungen. Qualitätsanforderungen beziehen sich im Allgemeinen auf die Leistung, Verfügbarkeit, Zuverlässigkeit, Skalierbarkeit oder Portabilität des betrachteten Systems. Auch Anforderungen dieser Art werden häufig der Klasse „nichtfunktionale Anforderungen“ zugeordnet.
- Randbedingungen (auch: Rahmenbedingungen) sind von den Projektbeteiligten nicht beeinflussbar. Randbedingungen können sich sowohl auf das betrachtete System beziehen (z. B. „Das System muss über einen Webdienst implementiert werden“) als auch auf den Systementwicklungsprozess (z. B. „Das System muss vorhanden sein“), das spätestens Mitte 2010 auf dem Markt ist“). Im Gegensatz zu funktionalen und qualitativen Anforderungen werden Randbedingungen nicht implementiert, sondern schränken die Auswahlmöglichkeiten für die Implementierung ein, d. h. den Lösungsraum während der Entwicklung.

Neben der Unterscheidung zwischen funktionalen Anforderungen, Qualitätsanforderungen und Randbedingungen werden in der Praxis noch einige andere Klassifizierungen von Anforderungen verwendet. Dies gilt beispielsweise für die in verschiedenen Standards (z. B. CMMI [SEI 2006], SPICE [ISO/IEC 1550 5]) definierten Arten von Anforderungen oder für die Klassifizierung von Werten, die zur Art der Anforderungen gehören, wie z. B. Granularität, Priorität oder Rechtsverbindlichkeit der Anforderungen.

Die Anforderungen werden auf Basis der Vision und der Projektziele erfasst. Es ist essentiell zu verstehen das Requirements Engineering, ein Requirement ist kein bestimmtes Artefakt auf einer definierten Granularitätsebene.

Ein Requirement beinhaltet alle Anforderungen an das System, unabhängig davon, auf welcher Abstraktionsebene oder in welcher Detailliertheit diese beschrieben ist, dazu gehören: User Stories, Epics, Features, Use Cases und auch übergeordneten Ziele gleichermaßen als Requirements zu bezeichnen.^[6]

Deshalb ist die Zentrale Aufgabe des Requirements Engineering relevante von nicht relevanten Stakeholdern zu teilen und die relevanten Stakeholder zu identifizieren.^[8]

Stakeholder stellen für den Requirement Engineer eine sehr wichtige Informationsquelle die dazu dient mögliche Anforderungen an das System zu identifizieren. Die Aufgabe von einem Requirement Engineer ist es die teils widersprüchlichen Ziele und Anforderungen der unterschiedlichen Stakeholder zu sammeln zu bearbeiten und filtern und die entsprechend zu dokumentieren dazu noch und mit allen Beteiligten zu konsolidieren.^[9]

Falls ein Requirement Engineer die Stakeholder nicht berücksichtigt oder werden wichtige Stakeholder nicht identifiziert, hat das zu Folge das mit sehr hoher Wahrscheinlichkeit erheblich negative Auswirkungen auf dem gesamten Projektverlauf verbuchen, da hierdurch Anforderungen nicht erkannt werden. Im Betrieb des System treten diese übersehenden Anforderungen als Änderungsanträge und dies Resultiert in zusätzlichen Verspätungen und wesentlich mehr Aufwand von dem Team als gedacht ,deren nachträgliche Integration hohe Zusatzkosten verursacht.

Deshalb ist es von höchster Wichtigkeit, das der Requirement Engineer alle Stakeholder identifiziert und effektiv die Stakeholder in die Anforderungsermittlung einzubeziehen. Einer der bekanntesten und meist verbreiteten Hilfsmittel zur Identifikation von Stakeholdern ist eine Checkliste, mit der relevante Stakeholder gezielt und systematisch ermittelt werden können. Eine inhaltlich unvollständige oder zeitlich zu spät vervollständigte Stakeholderliste resultiert dazu , dass wichtige Aspekte des Systems nicht identifiziert werden, das Projektziel verfehlt wird oder durch spätere Anpassungen signifikante Mehrkosten anfallen. Für die Identifikation der relevanter Stakeholder dienen meist Vorschläge bzgl. relevanter Stakeholder, die z.B. aus dem Management oder von Fachexperten stammen. Von dieser Basis können in späteren Verlauf weitere relevante Stakeholder ermittelt werden. [6]

Damit die Anforderungsermittlung best möglich erfüllt wird ist es wichtig die Zufriedenheit der Stakeholder hin einzubeziehen durch den Gewinn diese Kenntnisse hat der Requirement Engineer wichtiges Zusatzwissen was sehr hilfreich ist in der Anforderungsermittlung Diese Zufriedenheit von den Stakeholdern wird mit den jeweiligen Merkmalen eines Produkts, von denen sie abhängen, nach dem Modell von Kano in drei Kategorien eingeteilt: [10]

- Basisfaktoren Selbstverständlich vorausgesetzte Systemmerkmale (unterbewusstes Wissen). Muss vom System vollständig erfüllt sein, sonst stellt sich beim Stakeholder massive Unzufriedenheit ein. Vollständig erfüllte Basisfaktoren erzeugen keine positive Stimmung, sondern vermeiden lediglich, dass starke Unzufriedenheit entsteht.
- Leistungsfaktoren sind die explizit geforderten Systemmerkmale (bewusstes Wissen) und
- Begeisterungsfaktoren sind Systemmerkmale, die der Stakeholder nicht kennt, Faktoren die der Stakeholder sich nicht explizit gewünscht hat und erst während der Benutzung als angenehme und nützliche Überraschungen entdeckt (unbewusstes Wissen).

In der Laufe der Zeit ist ein Phänomen entstanden weil aus Begeisterungsfaktoren Leistungsfaktoren und schließlich Basisfaktoren denn der Nutzer gewöhnt sich an Merkmale eines Systems. Vor einer gewissen Zeit war es nicht selbstverständlich das jedes Hotel einen WLAN Anschluss hat aber heute ist es ein Basisfaktor geworden. Bei der Ermittlung der Anforderungen sind alle drei Anforderungskategorien zu berücksichtigen.

Basisfaktoren(unterbewusste Anforderungen) Selbstverständlich vorausgesetzte Systemmerkmale die auf jedem Fall vom System vollständig erfüllt sein müssen, sonst stellt sich

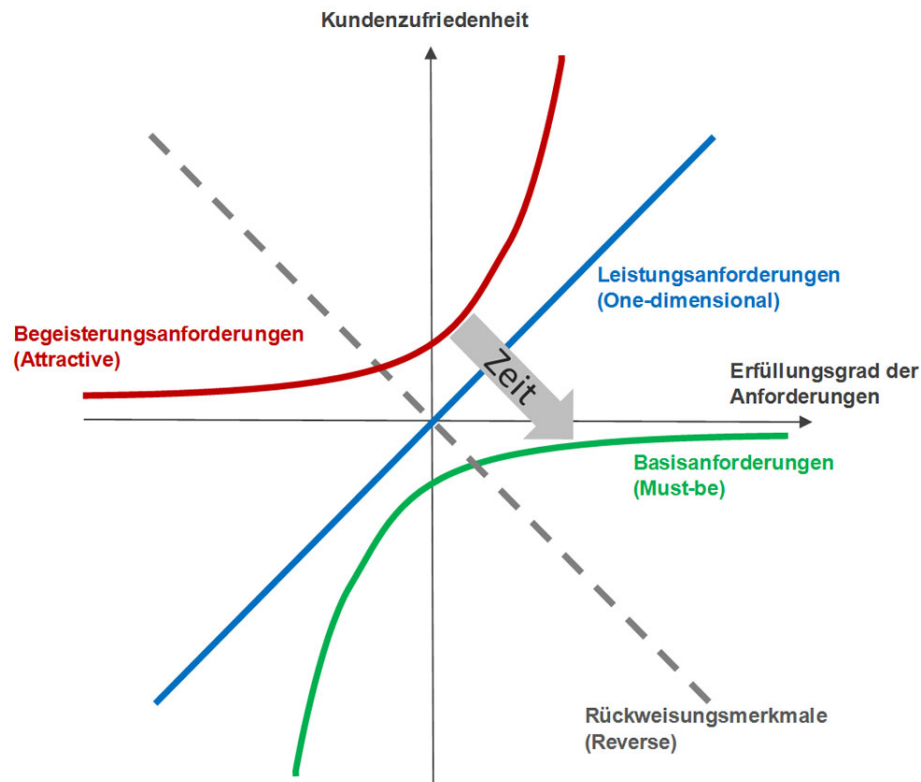


Figure 2: Darstellung von dem Kano-Modell der Kundenzufriedenheit [10].

beim Stakeholder massive Unzufriedenheit ein. Vollständig erfüllte Basisfaktoren erzeugen keine positive Stimmung, sondern vermeiden lediglich, dass starke Unzufriedenheit entsteht. Basisfaktoren orientieren sich meistens an bereits verfügbaren Systemen und durch die sind die Basisfaktoren sehr geprägt. Für deren Ermittlung eignen sich deshalb besonders Beobachtungstechniken und dokumentenzentrierte Techniken. Leistungsfaktoren (bewusste Anforderungen) Die explizit geforderten Systemmerkmale (bewusstes Wissen). Die Erfüllung dieser Merkmale erzeugt Stakeholderzufriedenheit und ist erstrebenswert. Falls einiger dieser geforderten Merkmalen fehlt, akzeptiert der Stakeholder das Produkt vermutlich, doch seine Zufriedenheit sinkt mit jedem fehlenden Leistungsfaktor. Leistungsfaktoren lassen sich gut durch Befragungstechniken ermitteln wie in Abbildung zu sehen 2. Begeisterungsfaktoren (unbewusste Anforderungen) Systemmerkmale, die der Stakeholder nicht kennt und erst während der Benutzung als angenehme und nützliche Überraschungen entdeckt (unbewusstes Wissen). Deren Wert erkennt der Stakeholder erst, wenn er sie selbst ausprobieren kann, oder sie vom Requirements Engineer vorgeschlagen bekommt. Für die Ermittlung solcher Begeisterungsfaktoren sind insbesondere Kreativitätstechniken geeignet.[10]

Anforderungen haben auch ihre rechtliche Relevanz : Auftraggeber und Auftragnehmer können auch rechtlich verbindlich Anforderungen in dem Projekt fordern. Durch

die schriftliche Dokumentation der Anforderungen können im Streitfall rechtliche Konflikte zwischen den Beteiligten zügig geklärt werden. In bestimmten Projekten oder Branchen (z.B sicherheitskritischen Umfeld wie Medizin) ist es notwendig, gewisse Dokumentationsvorschriften einzuhalten (z.B der Nachweis von spezifischen Anforderungen und Testfällen für ein System sowie deren Zusammenhang), um sich rechtlich abzusichern und die Haftungen zu reduzieren,^[4] Neben dem Zweck der Dokumentation für rechtliche Zwecke, ist die auch wichtig für die Nachhaltigkeit. Gewisse Informationen über ein System sind es wert, erhalten zu bleiben, damit diese für spätere Aufgaben und Modifikationen von dem Projekt verfügbar sind und die diese Informationen können sich gut in eine Datenbank speichern damit dadurch auch neue Informationen daraus gewinnen werden können, deshalb dienen sie auch als gemeinsamen "Wissensspeicher" der dem Team oder der Organisation hilft. Dadurch können Diskussion über früher getroffene Entscheidungen zu vereinfacht werden und Entscheidungen in neuen Projekten erleichtern.^[4] Es existieren drei unterschiedlichen Perspektiven um alle Anforderungen an ein System zu dokumentieren. In der Praxis wird sowohl die natürliche Sprache als auch konzeptuelle Modelle eingesetzt und die werden auch miteinander kombiniert. Die Anforderungen an das System können in unterschiedlichen Perspektiven auf das zu entwickelnde System dokumentiert werden. In der Strukturperspektive wird eine statisch-strukturelle Perspektive auf die Anforderungen an das System eingenommen.

- In dieser Perspektive basiert die Dokumentation auf die die Struktur von Ein und Ausgabedaten sowie die statisch-strukturellen Aspekte von Nutzungs und Abhängigkeitsbeziehungen des Systems dazu zählen auch Klassendiagramme und ER – Diagramme
- In der Funktionsperspektive werde die Informationen (Daten) aus dem Systemkontext, die durch das entwickelnde System manipuliert werden, die vom System in den Systemkontext fließen. Dokumentation der Systematik von Funktionsausführung zur Verarbeitung der Eingabedaten. Dazu werden auch die sogenannten Aktivitätsdiagramme benutzt
- Die Verhaltensperspektive beinhaltet Zustandsorientierte Dokumentation des Systems und dessen Einbettung in den Systemkontext dazu gehören die Reaktion des Systems auf Ereignisse im Systemkontext, Bedingungen eines Zustandswechsels sowie Effekte die dabei entstehen. Dazu werden auch die sogenannten Zustandsdiagramme benutzt ^[4]

Sehr viele Projekte scheitern, und Produkte erreichen die Marktziele nicht. Mangelhaftes Requirements Engineering ist immer unter den Top-3 Ursachen. Eine aktuelle Studie der Standish Group zeigt, dass ein gutes Drittel aller Projekte erfolgreich abgeschlossen wird. Ein Fünftel wird abgebrochen, und der Rest kommt zwar zu einem Abschluss, aber nur unter Aufgaben von ursprünglichen Zielen.^[13] Fehler in den Anforderungen haben die größten Effekte daher hat Requirements Engineering eine der wichtigsten Aufgaben.^[12] Die meisten abgebrochenen Projekte hatten nur ungenügend geklärte initiale Anforderungen und konnten Änderungen der Anforderungen nicht beherrschen ^[11]

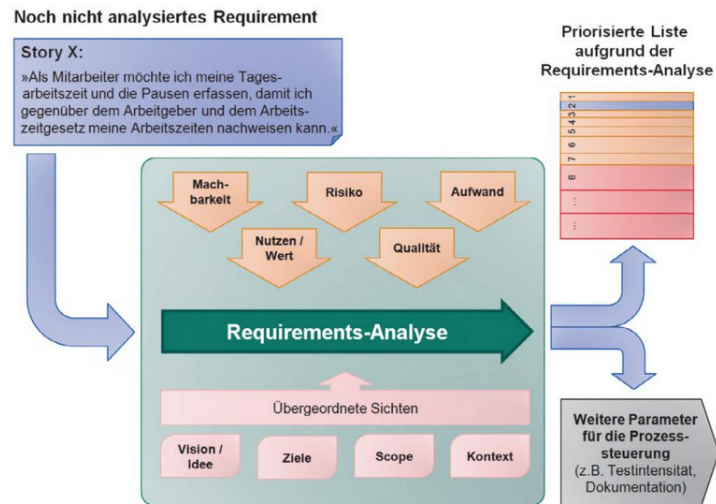


Figure 3: Darstellung einer validierung und Abstimmung einer Requirements [4].

Für die Verschiedenen Validierung und Abstimmung sind tagelange Meetings notwendig. Es reicht eine Diskussion und erfahrungsbasierte Einschätzung des Teams, um zu einer gute Bewertung zu kommen. Nach der Validierung der Anforderung müssen die Anforderung auf Machbarkeit, Nutzen und Geschäftswert, Risiko, Qualitätsanforderung, Aufwand und Priorität untersucht werden wie in der Abbildung zu sehen 3

Testgetriebene Entwicklung (Test Driven Development)

Testgetriebene Entwicklung (Test Driven Development) basiert auf der Idee, die jeweiligen Test zuerst zu schreiben, bevor das entsprechende Verhalten des Systems programmiert wird. Entwicklung ist auf jeder Ebene möglich (Akzeptanztests, Systemtests und auch Unit Tests). Grundsätzlich hat die testgetriebenen Entwicklung das Ziel eine möglichst hohe Testautomatisierung zu erreichen, dadurch wird versucht bei der Testausführung die Aufwände für die Regressionstests zu reduzieren.^[14]

Die Praxis beinhaltet die Implementierung eines Systems, das von den Komponententestfällen eines Objekts ausgeht. Das Schreiben von Testfällen und das Implementieren dieses Objekts oder dieser Objektmethoden löst den Bedarf an anderen Objekten/Methoden aus.^[15] Eine wichtige Regel in TDD lautet: "Wenn Sie keinen Test für das schreiben können, was Sie programmieren möchten, sollten Sie nicht einmal über das Programmieren nachdenken."^[16] TDD bietet die Gelegenheit kontinuierliche Regressionstests zu machen damit die Codequalität verbessert wird. Informatiker können besorgt sein über den Mangel an Upfront-Design in TDD und die Notwendigkeit, Designentscheidungen in jeder Phase zu treffen. Durch diese Sorgen entsteht das Bedürfnis, die Wirksamkeit dieser Praxis empirisch zu analysieren und zu quantifizieren

Die wesentlichen Vorteile von TDD

- Der Ablauf des Projektes ist sicherer da es weniger Risiko bei Änderungen am System gibt, weil die Änderungen durch kleine Schritte entlang bestandener Tests erfolgen und der bestehende Code so bereits durch Tests abgesichert ist.
- Die Wichtigkeit der Testautomatisierung ist viel größer als bei anderen Vorgehensweisen
- Die erstellten Unit Tests und Systemtests dokumentieren auch gleich das System. Dies verringert mögliche Risiken da es eine bessere Dokumentation darstellt als einfacher Quellcode

Einige Risiken beim Einsatz TDD

- Ein Testgetriebenes Vorgehen alleine macht noch keine guten Requirements es gehört viel mehr dazu
- Es existieren viele verschiedenen Aspekte des Systems die nicht automatisiert getestet und in den Build-Prozess eingebunden werden können
- Das TDD ist limitiert und dadurch können nicht alle Aspekte eine guten Spezifikation abgedeckt werden z.B grafische Prozessmodellierung , Visualisierung
- Durch die Testfälle musst aufgepasst werden,dass sie nicht zu schnell und zu tief in die Details abgleiten und Sonderfälle etc. beschreiben (evtl dies auch eine wichtige Aufgabe der Spezifizierer) ^[4]

Akzeptanz und Abnahmekriterien

Anforderungen müssen validiert, geprüft oder getestet werden können. Sonnst ist die Anforderung nur von beschränktem Nutzen bzw. Führt meist zu Mehraufwänden, weil dann typischerweise noch Missverständnisse oder Unklarheiten auftreten die das Vorgehen verlangsamen können Deshalb braucht jede Anforderung eine Reihe von bestimmten Kriterien, die getestet werden können, um festzustellen, ob die Anforderung auch erfüllt ist.

Zu den ursprünglichen Anforderungen werden passende Testfälle definiert, die Testfälle ergänzen sich und sind gemeinsam eine sinnvolle Gesamtspezifikation. Ein wichtiger Faktor ist die Granularität der Anforderung abhängig davon müssen entsprechende Akzeptanzkriterien definiert werden:^[6]

- Auf höheren Ebenen (Vision, Ziele und Epics) werden Normalerweise Erfolgskriterien definiert,da das Team öfters nur feststellen kann, ob die benötigte Funktionalität/Fähigkeit vorhanden ist oder nicht – was bedeutet, dass die Anforderung erfüllt wurde.
- Auf niedrigeren Abstraktionsniveaus können Akzeptanzkriterien verwendet werden, um schlussfolgern, wie die Lösung getestet werden soll, damit sie schließlich vom Kunden akzeptiert wird.

Durch die modellbasierten Dokumentation von Anforderungen im Requirements Engineering werden drei Ausprägungen /Darstellungen? von Anforderungen dokumentiert, die im Requirements Engineering ergänzend/zugleich zueinander eingesetzt werden:

- Ziele beschreiben Absichten von Stakeholdern oder Stakeholdergruppen. Ziele können gegebenenfalls auch in Konflikt zueinander stehen.
- Use Cases und Szenarien die dazu dienen beispielhafte Abläufe der Systemnutzung zu dokumentieren. Szenarien werden in Use Cases gruppiert.^[6]

Use Cases (Anwendungsfälle) wurden erstmals von vorgestellt, um die Funktionalität eines geplanten oder existierenden Systems auf der Basis einfacher Modelle untersuchen und dokumentieren zu können. Der Use-Case-Ansatz basiert dabei auf zwei Konzepten, die kombiniert eingesetzt werden: ^[17]

- Use-Case-Diagramme
- Use-Case-Spezifikationen

Nach OMG 2007 sind Use-Case-Diagramme der UML leicht verständliche Modelle, die dazu dienen, aus einer Nutzungssicht die Funktionalitäten des betrachteten Systems, dazu deren Beziehungen untereinander und die Beziehungen des Systems zu dessen Umgebung überblicksartig zu dokumentieren ^[18]

- Systemanforderungen (allgemein als Anforderungen bezeichnet) beschreiben detaillierte Funktionalitäten und Qualitäten, die das zu entwickelnde System umsetzen soll und die möglichst vollständig und präzise als Eingabe für die weiteren Entwicklungsschritte dienen.

Ein Modell steht eine abstrahierte Abbildung von einer existierenden Realität oder Vorbild für eine zu erschaffende Realität. Elemente der Abbildung können sowohl materielle wie auch immaterielle/gedankliche/ theoretische Objekte einer existierenden oder zu schaffenden Realität sein.

^[19] User Stories unterscheiden sich von Use Cases durch folgende wesentliche Punkte :

- Use Cases beschreiben eine einfache Ablaufsicht mit einem Hauptblauf, Varianten und evtl. Auch Negativfällen
- User Stories beschreiben eine einfache Anforderung an das System mit primär funktionalem Charakter
- Use Cases(in Form der Use-Case-Beschreibung bzw. - Szenarien) sind eine detailliertere Beschreibung als User Stories

Sowohl User Stories als auch Use Cases sind aus Kundensicht erstellt und sollen jeweils einen Mehrwert für den Kunden darstellen. User Stories können bzw. Sollen die grafisch modellierten Prozesse weiter beschreiben und so “zerlegen”, dass alle funktional wichtigen Teile aus den Prozessen herausgelöst und im weiteren Entwicklungsprozess verarbeitbar beschreiben werden. Prozesse selbst können nicht gut im Entwicklungsprozess

als Requirements weiterverarbeitet werden. Ein User Story muss klar aus Benutzer- oder Kundensicht geschrieben sein, im Idealfall vom User bzw. Kunden selbst. Gemäß [Cohn 2010] setzt sich eine User Story aus drei Bestandteilen zusammen :

- Schriftliche Beschreibung der Story
- Gespräche über die Story-Details
- Tests, die die Details und die Umsetzung dokumentieren

Ist der Ablauf der Aktion einer User Story nicht selbsterklärend, so kann dieser Ablauf Schritt für Schritt in einer oder mehreren Use Cases beschrieben werden. Die Use Cases beschreiben so die Anwendungsfälle einer in der Story abgebildeten Funktionen. [55]

2.2 Risikoanalyse

In Projekten und Aufgaben, wo Anforderungen nicht detailliert beschrieben sind und nicht genau definiert sind, da nicht immer jedem bewusst ist wie das Projekt weitergeht und wie die Anforderungen gestellt sind oder wie sie sich im Laufe von dem Projekt ändern ist es wünschenswert die Risiken zu definieren und zu beschreiben damit das Team einen guten Überblick bekommt und nicht den roten Pfaden verliert und mögliche Unsicherheit zu minimieren. In Softwareprojekten gibt es zahlreiche Risiken, die den Projekterfolg gefährden können, beispielweise:

- Das verwendete Framework für den Zugriff auf die Datenbank passte sich nicht an die erwartete Anzahl von Benutzern an
- Eine Anforderung in Modul X verursacht mehrere Fehler in Modul Y, die nicht sofort sichtbar sind
- Eine einzelne Anforderung ist so komplex, dass die aktuelle Architektur dafür komplett modifiziert werden muss.

Eine Großzahl der Risiken sind vermeidbar oder lassen sich verkleinern/ verringern, wenn das Team sich früh genug Gedanken macht und schon Pläne vorbereitet was zu tun falls diese Probleme und Risiken eintreten damit im Falle eines Eintretens von diesen Risiken das Development Team eine gute und schnelle Antwort parat hat Genau das ist das Ziel von Risikomanagement.[4]

Risiko identifizieren und bewerten

Risiken identifizieren bedeutet, gezielt zu untersuchen, welche Risiken im Projekt auftreten können. Risiken können aus unterschiedlichen Quellen entstehen.[20]

- Anforderungen
Die Risiken die mit den Anforderungen zusammenhängen entstehen aus dem Grund

das Anforderungen oftmals zu grob sind und unklar formuliert sind und daher benötigen sie viele Änderungen um das gewünschte Ziel/druga rijec zu erreichen. Der Kunde ist mit diesem stand der Dinge unzufrieden und das Produkt erfüllt seinen Zweck nicht. Aufwands und Zeitziele können nicht ganz eingehaltet werden und dies führt zu einem nicht erwünschten Ausgang.

- Vorgehensmodell und Methoden
Das Vorgehensmodell ist nicht geeignet, ein Projekt dieser Art erfolgreich abzuwickeln. Planung und Kontrolle sind sehr schwierig
- Team
Das Team was für das Projekt verantwortlich ist hat nicht die genügende Erfahrung in dem organisatorischen oder fachlichen Aspekt von dem Projekt dieser Art und Größe.
- Stakeholder
Die Arbeitsweise des Kunden passt nicht zum agilen Vorgehen. Der Kunde und die Ansprechpartner sind neu oder haben kaum Zeit, sich in das Projekt einzubringen und deshalb entstehen Probleme die mit einer hohen Wahrscheinlichkeit den Endprodukt schaden zufügen können ein gute Kommunikation zwischen den Kunden und dem Team ist erforderlich um das Projekt gut abzuschließen.
- Komplexität
Das Projekt bildet viele strenge Prozesse ab, viele voneinander abhängige Funktionen werden benötigt, die Technik ist komplex, es gibt viele Wechselwirkungen mit anderen Systemen. Qualitätssicherung ist für ein solches Projekt nicht geeignet.
- Technologie
Die eingesetzten Frameworks, Produkte und Technologien erlauben die geforderte Erfüllung nicht oder nur mit großem Aufwand.

Sofort nach der Erstellungsphase der Anforderungen macht sich der Produkt Leiter an das kritischen denken ran und überlegt sich welche Porenzielen Risiken diese Anforderungen mit sich bringen.Dabei ist es sehr wichtig sich unterstützung und Hilfe von dem ganzen Team, damit auch die Technischen Risiken und unerwünschte „Side effects“ gefunden werden und die auch analysiert werden.Risiken werden auch im der Aufwandsschätzung und bei der Planung von Anforderungen beleuchtet hierdurch erhofft man sich eine noch bessere Elimination von unerwünschten “Side effects”. Risikoidentifikation und Analyse passiert also immer nebenher, wenn die Anforderung beschrieben, diskutiert oder verplant wird damit haben wir einen ständigen Bearbeitungsprozess.^[21]

$$\mathbf{R(A)} = \mathbf{P(A)} * \mathbf{C(A)}$$

$\mathbf{P(A)}$ = Wahrscheinlichkeit des Eintretens von Ereignis A

$\mathbf{C(A)}$ = Schaden bei Eintritt von Ereigniss A

$\mathbf{R(A)}$ = Risikowert für Ereignis A

Risiko maßnahmen planen

Nachdem die wesentlichen Risiken identifiziert und bewertet wurden, sollten Maßnahmen für die Risiken mit hohem Risikowert identifiziert werden. Grundsätzlich gibt es verschiedene Arten von Maßnahmen, mit denen auf Risiken reagiert werden kann

- **Risikovermeidung** Das Team versucht das Produkt oder den Entwicklungsprozess so zu verändern damit dieses Risiko nicht mehr vorkommt damit wird das Problem umgehen und das Risiko das mit dem Produkt verbunden ist existiert nicht mehr.
- **Risikoeliminierung** Die Faktoren die mit dem Risiko verbunden sind , die dazu führen das das Risiko eventuell eintreten kann werden beseitigt.
- **Risikoverminderung** Das Team versucht den Risikowert möglichst zu verringern hierbei werden Maßnahmen getroffen, die als Ziel haben die Eintrittswahrscheinlichkeit und/oder den entstehenden Schaden zu reduzieren
- **Risikoabwälzung** Die gemeinsam im Team beschlossenen risikoreduzierenden Maßnahmen werden als Stories oder Tasks erfasst, ins Backlog aufgenommen und wie andere Projektaufgaben auch umgesetzt^[21]

Bewertung der Qualität der Anforderungen

Priorisierung Eine sehr wichtige Aufgabe des Produktowners ist es die Anforderungen entsprechend zu priorisieren. Bei der Iteration werden die Anforderungen ihrer Priorität nach abgearbeitet damit steht die Reihenfolge der Arbeitsschritte fest . Es ist aber sehr wichtig eine Laufende Aktualisierung aller Anforderungen zu haben damit diese aberbauungsReihenfolge möglichst best verläuft. Dies gilt als eine wichtiger milestone der vor dem Release und der Iterationsplanung müssen die Prioritäten für die erstellten Anforderungen gesetzt werden um eine möglichst gute Bearbeitung zu ermöglichen, dies ist eine Entscheidung des Produktowners die mit aus der Zusammenarbeit mit dem Team und den richtigen Stakeholdern entsteht. Wichtig ist aber auch hier der Grundsatz “ So viel wie nötig, aber nicht mehr.” Bei der Priorisierung muss entschieden werden, welche Skala man verwendet, um mögliche Prioritäten festzulegen die für die Iterationsplanung festgelegt werden sollen. Zusätzlich muss geklärt werden, auf Basis welcher Daten die Priorität einer Anforderung bestimmt wird **Prioritätsskala** Die umstandslose Verfahrensweise der Priorisierung der Anforderungen ist dessen Aufteilung/Anordnung in festgelegte Prioritätsklassen. Es gibt viele verschiedene Vorgehensweisen um diese Klassen zu bilden eine der bekanntesten ist die Einteilung nach Kano in Basis-, Leistungs-, Begeisterungsfaktoren, High Medium Low(HML)Priorisierung oder 1-10 Priorisierung. Die folgenden Requirements-Attribute sind verbreitete Basiswerte für die Priorisierung:

- **Geschäftswert für den Kunden**
In dieser Vorgehensweise hängt Grad der Priorisierung von dem Wert der Story,

desto höher ist der Wert dieser Story , desto höhere Priorität bekommt die Story dadurch wird die Story Früher umgesetzt

- Risiko
In dieser Vorgehensweise hängt Grad der Priorisierung von dem Risikowert der Story, desto höher ist der Risikowert dieser Story , desto höhere Priorität bekommt die Story dadurch wird die Story Früher umgesetzt
- Abhängigkeiten
In dieser Vorgehensweise hängt Grad der Priorisierung von dem Abhängigkeit der Story, desto höher ist der Wert dieser Story , desto höhere Abhängigkeit bekommt die Story dadurch wird die Story Früher umgesetzt
- Aufwand/Kosten
In dieser Vorgehensweise entscheidet das Team ob die aufwendigsten Anforderungen zuerst umgesetzt wird oder die einfachsten

Use-Case-Modell(Anwendungsfallmodellierung): Mit dem Use-Case wird zunächst die Außensicht auf das zu erstellende System erfasst und dokumentiert. Zwischen dem auslösenden Ereignis und dem vom System erwarteten Ergebnis wird ein Anwendungsfall definiert, der vom System unterstützt werden muss. Dies ist ein Bestandteil von den sogenannten Unterstützenden Techniken. Der Zweck der Unterstützenden Techniken ist es bei der Anwendung von Ermittlungstechniken als Ergänzung und versuchen mögliche Schwächen der gewählten Ermittlungstechniken aufzuheben.^[4]

2.3 Datenanalyse

Datenanalyse ist ein weit gefasster Begriff, der verschiedene Arten der Datenanalyse umfasst. Alle Informationen können analytische Techniken angewendet werden, um neue Einsichten und Einsichten zu gewinnen. Fertigungsunternehmen erfassen häufig Arbeitsstunden, Ausfallzeiten und Wartezeiten für die verschiedenen Maschinen, die sie besitzen, und analysieren diesen Datensatz dann, um Arbeitsvolumen besser zu planen, damit die Maschinen nahe ihrer maximalen Kapazität arbeiten und die Nutzungseffizienz höher ist. Durchführbarkeit. Neben der Erkennung eines problematischen Teils und Produktionsengpässen kann die Datenanalyse Unternehmen zugute kommen. Spieleunternehmen verwenden Analysen, um Kundenprämien einzustufen, um sicherzustellen, dass eine große Anzahl von Benutzern aktiv ihre Spiele spielt. Content-Marketing-Unternehmen verwenden Analysen, um Benutzer auf ihren Plattformen zu halten. Der Prozess, der die Datenanalyse begleitet, besteht aus mehreren verschiedenen Schritten.

- Definition der richtigen Fragen - In der Organisations- oder Geschäftsanalyse ist es notwendig, mit gewaschenen Fragen zu beginnen. Die Fragen sollten klar, prägnant und messbar sein. Es müssen Fragen entwickelt werden, um potenzielle Lösungen für ein bestimmtes Problem entweder zu qualifizieren oder zu disqualifizieren.
- Klare Messprioritätensetzung - Dieser Schritt kann in zwei Schritte unterteilt werden:

1. Bestimmen, was gemessen wird - Es ist notwendig zu überlegen, welche Metriken wir benötigen, um unsere Schlüsselfragen zu beantworten, die wir zuvor definiert haben. Dieser Satz von Metriken wird als Hauptleistungsindikatoren (Key Performance Indicators, KPIs) bezeichnet.
 2. Bestimmen, wie es gemessen wird - Wir müssen darüber nachdenken, wie wir die gesammelten Daten messen werden. Ein Beispiel dafür kann eine Entscheidung darüber sein, welchen Zeitrahmen wir verwenden werden (täglich, monatlich, vierteljährlich usw.) oder eine Entscheidung, in welcher Währung wir die Einnahmen berechnen (Euro, Dollar usw.).
- Datenerfassung - Wenn wir klar definierte Probleme und Prioritäten für Metriken haben, ist es an der Zeit, eine Strategie für die Datenerfassung zu entwickeln. In diesem Schritt müssen wir damit beginnen, die richtigen Daten aus den richtigen Quellen zu sammeln, um unsere Schlüsselfragen zu beantworten und bereits definierte Leistungsindikatoren zu messen.
 - Datenanalyse - In diesem Schritt müssen verschiedene mehr oder weniger ausgefeilte Softwaretools und Algorithmen angewendet werden, um eine angemessene Analyse durchzuführen. Ein guter Ausgangspunkt ist es, Daten auf verschiedene Arten zu manipulieren, z. B. das Finden von Korrelation und Interdependenz, das Untersuchen von Datenbereichen und anderen Merkmalen. Wenn wir Daten manipulieren, haben wir vielleicht tatsächlich alle Daten, die wir brauchen, aber der häufigere Fall ist, dass wir an diesem Punkt erkennen, dass wir das ursprüngliche Problem überarbeiten oder zusätzliche Daten sammeln müssen. In jedem Fall hilft uns die erste Analyse von Trends, Korrelationen, Variationen und Ausnahmen, uns darauf zu konzentrieren, genauere und bessere Antworten auf Schlüsselfragen zu geben.
 - Interpretation der Daten - Nach der Analyse der Daten und der möglichen Durchführung weiterer Untersuchungen ist es endlich an der Zeit, die Ergebnisse zu interpretieren. Bei der Interpretation der Analyse ist es notwendig zu wissen, dass wir niemals beweisen können, dass die Hypothese tatsächlich richtig ist, nur werden wir in der Lage sein, die Hypothese als falsch abzulehnen. Bei der Interpretation der Ergebnisse sind folgende Fragen zu beachten:
 1. Beantworten die Ergebnisse erste Fragen und wie?
 2. Helfen die Ergebnisse, sich gegen verschiedene Einwände zu verteidigen und wie?
 3. Gibt es bestimmte Einschränkungen für unsere Schlussfolgerungen, die wir möglicherweise nicht berücksichtigt haben?

Es gibt vier grundlegende Arten der Datenanalyse. Im Folgenden werden sie von einfacher bis anspruchsvoller ausgestellt. Je komplexer die Art der Analyse, desto komplexer die Analyse, aber auch der Wert, den die Analyseergebnisse bringen.

1. Deskriptive Analyse - Beantwortet die Frage was passiert ist und gilt als einfachste Form der Analyse. Die enorme Datenmenge, die Unternehmen heute verwenden, zerlegt große Datensätze in verständliche Teile. Der Zweck dieser Art von Analyse besteht darin, die Erfindungen zusammenzufassen und zu verstehen, was in der Vergangenheit wirklich passiert ist. Die wichtigsten verwendeten Techniken sind: Ad-hoc-Berichterstattung, Data Mining, Datenaggregation und aggregierte Statistiken.
2. Diagnostische(Diagnostic) Analyse - Beantworten Sie die Frage, warum etwas passiert ist. Diagnostische Analysen versuchen, die Hauptursache bestimmter Ereignisse besser zu verstehen. Es ist nützlich zu bestimmen, welche Faktoren und Ereignisse zum Ergebnis beigetragen haben. Es verwendet hauptsächlich Wahrscheinlichkeit, Signifikanz und Verteilung der Ergebnisse für die Analyse. Die wichtigsten verwendeten Techniken sind: Hauptkomponentenanalyse, Sensitivitätsanalyse und Regressionsanalyse. Machine-Learning-Algorithmen wie Klassifizierung und Regression gehören ebenfalls zu dieser Art von Analyse.
3. Prädiktive(Predictive Analytics) - Beantwortet die Frage, was passieren könnte, wenn bestimmte Bedingungen erfüllt sind. Predictive Analytics wird verwendet, um zukünftige Ergebnisse zu bestimmen, aber es ist wichtig zu beachten, dass es nicht wirklich sicher sagen kann, ob ein Ereignis in der Zukunft stattfinden wird, sondern nur die Wahrscheinlichkeit vorhersagt, dass das Ereignis tatsächlich stattfinden wird. Vorhersagemodelle werden auf der Grundlage von Vorphasen der deskriptiven Analyse erstellt. Die wichtigsten verwendeten Techniken sind: quantitative Analyse, prädiktive Modellierung, maschinelle Lernalgorithmen.(machine learning)
4. Präskriptive (Prescriptive) - Beantwortet die Frage, welche Aktionen in Abhängigkeit von den gewünschten Ergebnissen am besten geeignet sind. Die Grundlage dieser Phase ist Predictive Analytics, geht jedoch über die oben genannten drei Kategorien hinaus, indem es zukünftige Lösungen und Maßnahmen vorschlägt. Es kann alle günstigen Ergebnisse gemäß einer bestimmten Vorgehensweise vorschlagen und kann auch verschiedene Handlungsweisen vorschlagen, um ein bestimmtes Ergebnis zu erzielen. Es verwendet tatsächlich ein leistungsstarkes Feedback-System, das die Beziehung zwischen Aktionen und Ergebnissen kontinuierlich lernt und aktualisiert. Die wichtigsten verwendeten Techniken sind: Simulationsanalyse, Systemempfehlung, künstliche Intelligenz, neuronale Netze.

Das Gesundheitswesen ist eine der Branchen mit dem größten Potenzial für Big Data. Nach herkömmlicher Definition bezieht sich Big Data auf die Tatsache, dass die heutigen Datensätze oft zu groß, zu heterogen und zu schnell wachsen, um von herkömmlichen Technologien gespeichert, analysiert und verwendet zu werden. Big Data wird durch drei Technologietrends getrieben: Geschäftsprozesse werden zunehmend elektronisch erledigt, Einzelpersonen generieren immer mehr Daten – z. B. in sozialen Netzwerken – und die weitere Digitalisierung, durch Smartphones und Apps im Alltag. Auch

Quellen für Big Data im Gesundheitswesen – einige Beispiele

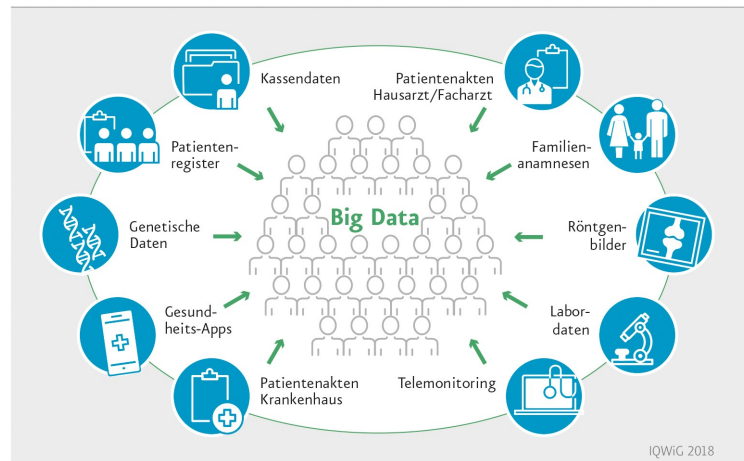


Figure 4: Darstellung der Quellen die in der Big Data in dem Gesundheitswesen eingesetzt werden^[22]

im Bereich Medizin und Gesundheitswesen zeichnen sich neue Trends hin zu neuen und spannenden Datenquellen und innovativen Möglichkeiten der Datenanalyse ab. Das gilt einerseits für die Forschung. Hier braucht beispielsweise die Omics-Forschung ganz klar Big-Data-Technologien. In der medizinischen Praxis bieten elektronische Patientenakten, frei verfügbare Daten und quantifizierte selbstgesteuerte Tendenzen, also die Messung des eigenen Verhaltens, neue Möglichkeiten der Datenanalyse. In Bezug auf die Analytik wurden in der Vergangenheit erhebliche Fortschritte bei der Extraktion von Informationen aus Textdaten erzielt, wodurch riesige Datenmengen aus der medizinischen Literatur für die Analyse zugänglich gemacht werden. Gleichzeitig ist aber aufgrund technischer, rechtlicher und ethischer Rahmenbedingungen insbesondere in der Medizin und im Gesundheitswesen die Nutzung von Big Data hier noch weniger ausgeprägt als in anderen Bereichen. Erste spannende Best-Practice-Beispiele in der Medizin und im medizinischen Bereich lassen jedoch innovative Ansätze und Ergebnisse erwarten. Dieser Abbildung4 gibt einen Überblick über das Potenzial von Big Data in der Medizin und im Gesundheitswesen und die Quellen die noch hinkommen. ^[22]

Die Methoden der Computer Vision werden in der Zukunft viele Arbeitsbereiche verändern, da es damit möglich ist, sehr arbeitsaufwendige Tätigkeiten zu unterstützen. Zusätzlich können automatische Bildererkennungssysteme, auch bei der Entdeckung von Kontaminationen oder anderen Unreinheiten, hilfreich sein. Ein Bereich, bei dem diese Technik in der Automobilbranche bereits Anwendung findet, ist das autonome Fahren ^[24]

Herkömmliche Datenspeicherung hat ihre Grenzen und Beschränkungen. Wir werden uns mit den Problemen befassen, mit denen diese traditionellen Architekturen angesichts der heutigen enormen Mengen unterschiedlicher Daten konfrontiert sind, die blitzschnell generiert werden. Die Architektur herkömmlicher Datenspeicherlösungen umfasste in der

Regel zwei verwandte Komponenten: Speicherung und Verarbeitung. Warehouses wurden in der Regel mit einigen Tools ETL(Extrakttransformationslast) kombiniert, die Daten innerhalb von Tabellen in einem Datenspeicher in einem vordefinierten Layout verarbeiteten und bereitstellten. Alle Komponenten einzelner Datenspeicher werden typischerweise auf einem gemeinsam genutzten Computer ausgeführt, was bedeutet, dass die Speicherkapazität an die Rechenleistung gebunden ist und nicht individuell skaliert werden kann. Eine weitere auferlegte Einschränkung besteht darin, dass die Datenverarbeitung auf diese bestimmte speicherverwendende Sprache (normalerweise SQL) beschränkt ist. Diese eingeschränkte Unterstützung für neue Datenformate und neue Verarbeitungsszenarien.

Datenvielfalt

Diversity-(Sorten-)Daten, die verarbeitet werden, sind für die Analytik sehr wichtig. Herkömmliche Data Warehouses sind in erster Linie für die Arbeit mit strukturierten Daten gedacht. Dies funktioniert gut, wenn die meisten Daten aus einer relationalen Datenbank gesammelt werden, aber mit der Explosion von SAAS-Diensten, verschiedenen IoT-Sensoren und sozialen Netzwerken werden die Daten benötigt, um immer vielfältigere und immer anspruchsvollere Formen unstrukturierter Daten wie Text und Audio zu analysieren, Videos oder Bilder. Mit der Entwicklung von SAAS-Anwendungen und der Microservice-Architektur tritt auch der Aufstieg des JSON-Formats (Javascript Object Notation) ein. Obwohl es eine bestimmte Struktur hat, ist diese Struktur nicht garantiert und kann sich ändern. Es kommt häufig vor, dass der SAAS-Anbieter die API (Application Programming Interface) ändert, die er seinen Kunden anbietet, und JSON reagiert auf Änderungen seines Schemas. Herkömmliche Datenspeicher sind anfällig für dieses Problem und so konzipiert, dass die Daten eindeutig einem bestimmten Schema entsprechen. Der Mangel an Flexibilität und die Fokussierung nur auf strukturierte Datentypen haben Organisationen eingeschränkt und die Entwicklung von Analysesystemen verhindert.^[22]

Datenmenge

In traditionellen Implementierungen von Datenspeicherung, -speicherung und -verarbeitung sind sie eng vorbereitet, was die Flexibilität und Skalierbarkeit stark einschränkt [1]. Bei der Erhöhung des Datenverkehrs und der Datenmenge ist es notwendig, neue und stärkere Server mit mehr Arbeitsspeicher, Prozessorleistung und mehr Festplatten zu kaufen. Da wir nicht mehr Speicherplatz erhalten können, ohne die Prozessorleistung zu erhöhen, bedeutet dies, dass wir am Ende mehr für unnötige Prozessor- und Speicherleistung bezahlen. Diese Tatsache führte dazu, dass die Verarbeitung großer Datenmengen mit traditioneller Datenspeicherung nur für Unternehmen mit großen Budgets für Informationstechnologien gedacht war.

Moderne Analyseplattformen

Im vorherigen Abschnitt wurden mehrere traditionelle Lösungen zur Implementierung eines Analysesystems besprochen, von einem einfachen Aufbau mit einer Produktionsanlage im Zentrum des Systems bis hin zu einem Schwerpunkt auf Datenspeicherung. Es wurde gezeigt, dass die Entwicklung von Big Data große Probleme mit sich bringt, so dass Unternehmen, die in diesem neuen Zeitalter führend sein wollen, sich an das schnelle Datenwachstum anpassen und ihre Analyseplattformen weiterentwickeln müssen. Die Entwicklung des Cloud Computing bringt neue Innovationen und neue Tools mit sich, die Unternehmen bei der Lösung aufkommender Probleme unterstützen. Eine der wichtigsten Innovationen war die Entwicklung moderner Cloud Data Warehouses wie Google BigQuery, Amazon Redshift, Azure SQL Data Warehouse und Snowflake. Diese modernen Data Warehouses haben einen Umbruch im Bereich Analytics und Business Intelligence geschaffen. Teure, unflexible und ineffiziente traditionelle Lösungen wurden durch neue Dienste ersetzt, bei denen Unternehmen die Möglichkeit haben, nur für die Menge an Daten zu bezahlen, die sie speichern und verarbeiten, sowie für die Ausfallsicherheit, die es ihnen ermöglicht, die Systemleistung an ihre Geschäftsanforderungen anzupassen. In diesem Abschnitt wird erklärt, warum speicherbasierte Analysesysteme manchmal nicht die beste oder ausreichende Lösung sind, auch wenn moderne Cloud-Technologien verwendet werden. Wir stellen eine Alternative vor, eine Analytics-Plattform, ihre Grundbausteine und die Vorteile, die dieser Ansatz bietet. [22]

Datenerfassung und Zentralisierung

Hier werden die Ebenen der Datenerfassung und -speicherung genauer untersucht. Der erste Schritt, der bei der Implementierung der Cloud-Analyseplattform erforderlich ist, ist die Implementierung der Erfassungsschicht, damit die Daten auf der Plattform ankommen können. Eines der Hauptmerkmale der Plattform ist ihre Fähigkeit, mit heterogenen Quellen umzugehen, die Daten in verschiedenen Formen und Formaten generieren. Die Vielfalt der Datenquellen muss vor der Implementierung sorgfältig berücksichtigt werden, und in diesem Kapitel werden wir uns mit den Besonderheiten der Datenerfassung aus verschiedenen Quellen wie Datenbanken, Dateien, Anwendungsprogrammierschnittstellen (APIs) und Streams befassen. Nachdem eine angemessene Unterstützung geleistet wurde, um Daten aus verschiedenen Quellen zu sammeln, ist es notwendig, die Daten zu zentralisieren und in einem gemeinsamen Speicherbereich zu speichern. [22]

Datenquellen

Jede der oben genannten Quellen hat ihre eigenen spezifischen Eigenschaften. Beispielsweise ist relationalen Datenbanken immer ein Typ zugeordnet, der einer bestimmten Spalte in einer Tabelle zugeordnet ist, während in CSV-Dateien (kommagetrennte Werte) mit Tabellenformat keine zugeordneten Datentypen vorhanden sind, aber es ist die Aufgabe einer Anwendung, die eine Datei liest, dies zu bestimmen. Schnittstellen, die von SAAS-Anwendungen angeboten werden, sind in der Regel so eingerichtet, dass sie eine Antwort im JSON-Format zurückgeben. Dieses Format ist semi-strukturiert, so

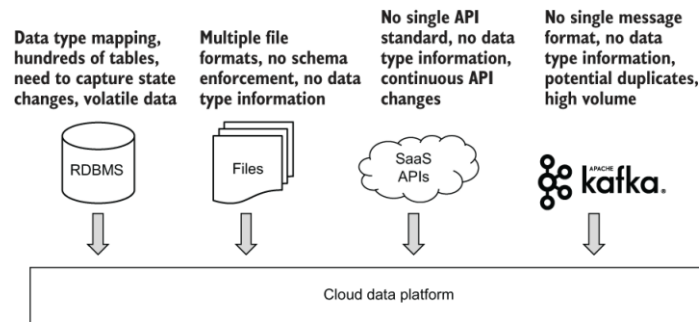


Figure 5: Darstellung von den möglichen Datenquellen die in eine Cloud Plattform hineinfließen [22]

dass erwartet werden kann, dass bestimmte Änderungen angemessen verarbeitet werden. Das folgende Diagramm⁵ veranschaulicht die Hauptmerkmale dieser Datenquellen. Streaming-Daten stellen eine besondere Kategorie dar, die in den letzten zehn Jahren eine Expansion erfahren hat. Dies sind Daten, die mit hoher Geschwindigkeit generiert werden und Unternehmen dazu zwingen, über angemessene Möglichkeiten nachzudenken, diese schnell wachsenden Daten zu akzeptieren und in Echtzeit zu verarbeiten.^[22]

Datenbank als Quelle

Unternehmen, die den größten Teil ihres Geschäfts über eine Website oder App abwickeln (z. B. Amazon), verfügen über eine große Menge an Betriebs- und Transaktionsdaten. Diese Anwendungen sind direkt mit der Produktionsdatenbank verbunden, die der zentrale Ort ist, an dem alle Transaktionsdaten gespeichert sind, die für den ordnungsgemäßen Betrieb der Anwendung und anderer interner Systeme erforderlich sind. Die Anwendungsdatenbank kann eine SQL-Datenbank oder eine NoSQL-Datenbank sein. Relationale Datenbanken sind eine der häufigsten Quellen, aus denen Daten für die Analyseplattform gesammelt werden. Die Daten in diesen Datenbanken sind in Tabellen organisiert, wobei jede Tabelle eine oder mehrere Spalten enthält und jede Spalte einen eigenen Datentyp hat. Dieses klar und präzise definierte Datenschema stellt sicher, dass kein Fall auftritt, in dem Daten in die falsche Tabelle oder Spalte eingefügt werden. Die Tabellen innerhalb der Datenbank sind oft normalisiert, was bedeutet, dass logische Entitäten in mehrere physische Tabellen aufgeteilt werden, die durch Primär- und Fremdschlüssel miteinander verbunden sind. Aufgrund dieser Art der Tabellenmodellierung ist es nicht ungewöhnlich, dass eine einzige relationale Datenbank Hunderte von Tabellen enthält. Da relationale Datenbanken als Geschäftsunterstützungsbasis verwendet werden, ändern sie sich häufig, und jedes Ereignis innerhalb der Anwendung muss in einigen Änderungen an der Datenbank gespiegelt werden. Wenn wir erwägen, Daten aus relationalen Datenbanken in unsere Plattform zu ziehen, haben fast alle Datenbanken einige gemeinsame Merkmale, die berücksichtigt werden müssen ^[22]:

- Datentypzuordnung – Spalten mit ihren Datentypen in einer relationalen Daten-

bank müssen Spalten innerhalb des Datenspeichers zugeordnet werden. Leider hat jeder Hersteller von relationalen Datenbanken seine eigenen spezifischen Datentypen. Obwohl es Typen gibt, die in jedem System sehr ähnlich oder gleich sind, z. B. Zeichenfolge oder ganze Zahl, gibt es auch bestimmte Typen wie Zeitstempel oder Given, die eine andere Bedeutung, Genauigkeit oder ein anderes Format haben können.

- Automatisierung - Da relationale Datenbanken Hunderte von Tabellen haben können, muss der Datenziehprozess automatisiert und anpassbar sein. Es ist unwahrscheinlich, dass eine Person genug Zeit hat, um Daten aus mehreren hundert Tabellen manuell zu konfigurieren. Wenn zufällig genug Zeit für eine manuelle Einstellung vorhanden ist, besteht eine gute Chance auf menschliche Fehler in diesem Prozess.
- Variabilität - Die Daten in den Datenbanken sind variabler Natur. Das Geschäft eines Unternehmens ruht nie und es passieren ständig neue Transaktionen. Wenn wir uns das Beispiel eines Online-Shops ansehen, werden wir feststellen, dass jede Minute Hunderte oder Tausende neuer Bestellungen geplant, gesendet oder storniert werden. Diese Aktionen führen zu Änderungen an einer großen Anzahl von Tabellen.

Dateien als Quelle

Dateien sind eine weitere sehr häufige Datenquelle von der Analyseplattform. In der Regel handelt es sich dabei um Text- oder Binärdateien, die über FTP (File Transfer Protocol) direkt an ein Ziel übertragen oder in Cloud-Data Lakes wie Amazon S3, Google Cloud Storage oder Azure Blob Storage geladen werden. Dateien können auch automatisch generiert werden: Die Serveranwendung generiert Protokolldateien, in denen sie alle relevanten Ereignisse für den Tag speichert. Eine weitere Möglichkeit besteht darin, CSV-, TXT- oder XLS-Dateien manuell von einer Person innerhalb oder außerhalb der Organisation zu generieren.

Die Dateien scheinen auf den ersten Blick eine sehr einfache Datenquelle zu sein, aber unter dem Gesichtspunkt der Automatisierung des Zeichnens von Daten aus Dateien sollten sie nicht auf die leichte Schulter genommen werden und ihre Eigenschaften müssen gründlich überprüft werden. Eines der wichtigeren Merkmale ist, dass sie in verschiedenen und zahlreichen Formaten erhältlich sind, entweder Text oder binär. Die beliebtesten Formate, die wir treffen können, sind CSV, JSON und XML. Binärformate wie Parquet und Avro treffen sich seltener, spielen aber eine sehr wichtige Rolle in der Welt von Big Data. Textdateien enthalten keine Spaltendatentypen und geben in der Regel keine bestimmte Struktur in der Datei selbst an. Dies bedeutet, dass die Dateien, die wir aus derselben Quelle lesen, nicht der gleichen Struktur folgen müssen und im Laufe der Zeit geändert werden können. Die Datenerfassungsschicht muss bereit sein, auf mögliche Änderungen in den Quelldateien zu reagieren, sowie robust genug, um alle Fälle zu verarbeiten.

Details, die bei der Implementierung eines Datei-Dragging-Systems zu beachten sind [22].

- Analysieren verschiedener Formate - Das System muss in der Lage sein, verschiedene Dateien wie CSV, JSON, XML, Avro und andere zu analysieren. Für Textdateien gibt es keine Garantie, dass der Hersteller die gleiche Struktur verwendet, die Parser erwartet.
- Schemaänderung - Im Gegensatz zu relationalen Datenbanken sind das Hinzufügen einer neuen Spalte oder das Ändern des Typs einer Spalte einfache Operationen für jemanden, der JSON- oder CSV-Dateien generiert, und dementsprechend sollte das System in der Lage sein, korrekt auf die Änderung des Dateischemas zu reagieren.
- Snapshot und mehrere Dateien - Im Gegensatz zu relationalen Datenbanken, die variabel sind, sind Dateien eine Momentaufnahme des Zustands der Daten in der Zeit. Die Abfolge der Ereignisse ist wie folgt: Die Daten werden aus dem Quellsystem extrahiert, dann in einer Datei gespeichert und schließlich auf die Plattform hochgeladen. Diese Snapshots des Quellsystems können in Form einer oder mehrerer Dateien vorliegen.

Stream als Quelle

Eine der neueren Datenquellen, die von Analyseplattformen benötigt werden, ist das Streamen von Daten. Streaming-Daten stellen Ereignisse dar, die in großen Mengen und mit hoher Geschwindigkeit generiert werden. Das Konzept des Daten-Streamings wird in der heutigen Welt immer häufiger und eine wachsende Zahl von Unternehmen muss Ereignisse analysieren und in Echtzeit darauf reagieren. Ein Beispiel für das Streamen von Daten kann ein mobiles Spiel sein, bei dem die Aktion und der Klick jedes Benutzers in Ereignisse umgewandelt und an Ereignisakzeptanzsysteme gesendet werden. Ein weiteres Beispiel sind IoT-Sensoren, die jede Sekunde neue Daten produzieren und diese Messungen zur weiteren Verarbeitung und Analyse über das Netzwerk senden.^[22]

Apache Kafka ist wahrscheinlich die bekannteste Plattform zum Einlesen und kurzfristigen Speichern von Streaming-Daten [4]. In den letzten Jahren haben sich Cloud-Dienste entwickelt, die für die gleichen und ähnliche Zwecke wie Amazon Kinesis, Google Cloud Pub/Sub, Azure EventHub verwendet werden. Wenn Sie mit Streaming-Daten arbeiten, mit Ausnahme einer Plattform wie Kafka, die Nachrichten akzeptiert, gibt es Hersteller (Produzenten), die neue Nachrichten erstellen und sie an das Streaming senden, und es gibt Verbraucher (Verbraucher), die Nachrichten von Kafka lesen. Diese Technologien geben Unternehmen die Möglichkeit, große Mengen an Streaming-Daten zu verarbeiten. Die Hauptmerkmale von Streaming-Daten sind^[22]:

- Nachrichten innerhalb der Streaming-Plattform haben Einschränkungen für das Format. In der Regel werden Nachrichten als eine Reihe von Bytes gespeichert und können durch Codierung mit JSON, Avro oder ähnlichen Formaten erstellt werden. Die Datenerfassungsschicht muss in der Lage sein, diese Nachrichten zu entschlüsseln.

- Streaming-Plattformen können auch doppelte Nachrichten enthalten, was bedeutet, dass die Verbraucher in der Lage sein sollten, mit Duplikaten umzugehen.
- Nachrichten, die in Streams eingegeben werden, sind unveränderlich. Sobald eine Nachricht in ein System wie Kafka eingegeben wurde, kann sie nicht mehr geändert werden, aber eine neue Version dieser Nachricht kann eingegeben werden.
- Streaming-Daten gibt es in der Regel in großen Mengen. Es ist nicht ungewöhnlich, dass Unternehmen Milliarden von Ereignissen pro Tag sammeln. Die Sammlungsschicht sollte skalierbar sein.

SAAS-API als Quelle

Die meisten Unternehmen verwenden jetzt mindestens ein SaaS-Produkt (Software as a Service), um ihr Geschäft zu unterstützen. Produkte wie Salesforce und Marketo speichern einige der wichtigsten Datensätze für ein Unternehmen. Zusammenführen von Benutzerdaten (z. B. Salesforce-Daten) und Daten für Marketingkampagnen (z. B. Salesforce-Daten) und Daten für Marketingkampagnen (z. B. Marketo) mit den Transaktionsdaten, die das Unternehmen in seinen Produktionsbasen aufbewahrt, ist einer der sehr wünschenswerten Vorteile der Analyseplattform. Die meisten SAAS-Unternehmen ermöglichen die Datenextraktion über die API THE REPRESENTATIONAL State Transfer. In der Regel können Daten manuell aus einer Anwendung im CSV-Format heruntergeladen werden, aber die Datenextraktion über die API ist in den meisten Fällen praktischer und flexibler. Es gibt alle möglichen Probleme, auf die wir bei der Verwendung der SAAS-API stoßen. Einige der wichtigsten sind [22]:

- Jeder Anbieter hat eine einzigartige Möglichkeit, seine Daten externen Nutzern zugänglich zu machen, was bedeutet, dass es keinen Standard gibt, den alle Unternehmen implementieren.
- Die meisten SAAS-APIs verfügen nicht über Informationen zu Datentypen, und das Schema kann sich ändern.
- Es gibt keinen einheitlichen Standard für inkrementelle und vollständige Datenzeichnungen aus der SAAS-API. Organisationen müssen sich an einzelne Anbieter anpassen.

Data minning und Datenforschung

Einer der Hauptgründe für die Einführung des Data Lake ist die Tatsache, dass es nicht immer möglich ist, Eingabedaten nach dem definierten Schema und der definierten Struktur zu formatieren [25]. Durch die Speicherung der Daten in ihrer ursprünglichen Form in einem Data Lake können wir alle Daten sofort aufbewahren und die Analyse für später verlassen. Abhängig von der Art der Rohdaten und der Art der Analyse können Prozesse und Workflows je nach Komplexität von [25] variieren. Da die meisten der im See gespeicherten Daten nicht sofort verbrauchs- und nutzbereit sind,

müssen zunächst bestimmte Mining- und Datenforschungsverfahren durchgeführt werden, um die potenzielle Bedeutung dessen zu erkennen^[25]. Jupyter ist eines der am häufigsten verwendeten Tools zur Erforschung von Rohdaten. Jupyter ist eine Open-Source-Software, mit der Ingenieure interaktiv Code in verschiedenen Programmiersprachen wie Python, R und SQL ausführen können^[26]. Der Programmiercode wird in Arbeitsmappen (Notebooks) geschrieben, die in einem Webbrowser ausgeführt werden, da die Jupyter-Webanwendung^[26] ist. Google Cloud verfügt über einen Cloud Datalab-Dienst, der eine selbsttragende Version des Jupyter Notebook ist^[27]. Dieser Service wird mit bereits installierter gängiger Datensoftware wie TensorFlow und NumPy geliefert^[27]. Zusätzlich zum Datalab-Dienst stehen über Dataproc-Dienste traditionelle Hadoop-Tools zur Verfügung, um Rohdaten im Google-Ökosystem zu recherchieren. Für die SQL-Sprachforschung, Benutzer können Rohdaten über den DataPrep-Dienst verarbeiten und in den BigQuery-Datenspeicher laden. Dataprep ist ein Dienst, der die Bereinigung, Aufbereitung und Verarbeitung von Daten über die grafische Oberfläche ermöglicht^[8]. Auf diese Weise können Geschäftsanwender, die nicht unbedingt über fortgeschrittene technologische Kenntnisse verfügen, einen bestimmten Rohsatz aus dem Data Lake erkunden. Nach dem Verständnis des analytischen Potenzials eines Rohdatensatzes aus dem See beginnt die Erstellung von Prozessen und Arbeitsabläufen, diesen Rohdatensatz zu transformieren und in ein anderes Lager zu verschieben, wo er für die Analyse durch eine breitere Gruppe interner oder externer Benutzer zur Verfügung steht^[25].

Datenspeicherung und Business Intelligence

Eines der wichtigsten Dinge, die eine Organisation hat, sind ihre Daten^[28]. Sie werden in der Regel für zwei Zwecke verwendet: operative Aufzeichnungen und analytische Entscheidungsfindung. Einfach ausgedrückt, sind Business-Support-Systeme der Ort, an dem Daten in das System gelangen, und die Datenspeicherung ist der Ort, an dem Daten landen und einsatzbereit werden^[28].

Das Ziel von Business Intelligence ist es, allen Mitgliedern der Organisation, die sie für die analytische Entscheidungsfindung benötigen, Zugriff auf Daten zu ermöglichen, die sich an Fakten und Daten orientieren^[28]. Geschäftsanwender sollten in die Lage versetzt werden, Daten mithilfe von Ad-hoc-Tools der Business Intelligence zu untersuchen. Diese Tools folgen einem Paradigma der direkten Manipulation, bei dem Benutzer keinen Abfragecode schreiben müssen, sondern eine Reihe von Tabellen und Spalten auswählen, die sie interessieren, und Daten durch einfaches Ziehen, Doppelklicken oder Einfügen der Option abrufen^[29]. Business Intelligence wird für Business Analytics zusammen mit anderen Technologien wie Predictive Analytics, Machine Learning oder Operational Software Research eingesetzt^[29]. Eines der am häufigsten verwendeten Ergebnisse der Business Intelligence-Anwendung ist ein täglicher Bericht mit Grafiken und Tabellen, der in der Regel per E-Mail an Mitglieder der Organisation übermittelt wird und auf einfache Weise Einblick in den aktuellen Status des Produkts bietet. Die widersprüchlichen Aspekte von Data Science können oft in derselben Anwendung sichtbar werden. Zum Beispiel verwendet der Einsatz von Data Science und Analyt-

ics im Krankenversicherungsgeschäft Marketingdatensätze von Drittanbietern, die Informationen wie Kaufgewohnheiten, Web-Suchverlauf, Alog mit Hunderten von anderen Attributen enthalten, die sich auf den Lebensstil der Menschen beziehen.^[30](Batty, Tripathi, Kroll et al. 2010[11]). Die Verwendung dieser Daten Dritter ist problematisch, da sie eine Selbstdisziplinierung auslösen kann, bei der Menschen bestimmte Aktivitäten meiden, wie der Besuch von Extremsport-Websites oder Häufige Arztbesuche, aus Angst, höhere Versicherungsprämien zu zahlen^[31]. Die Begründung für die Verwendung dieser Daten ist jedoch, dass sie als Stellvertreter für invasivere und teurere Informationsquellen wie Blutuntersuchungen fungieren und im ONG-Begriff Kosten und Prämien senken und dadurch die Zahl der krankenversicherten Personen erhöhen werden.^[30] Die Bruchlinien in der Debatte zwischen den kommerziellen Vorteilen und ethischen Überlegungen des Einsatzes von Data Science und Analytics zeigen sich in den Diskussionen um die Verwendung personenbezogener Daten für gezieltes Marketing. Aus Sicht der Geschäftswerbung besteht der Anreiz zur Verwendung personenbezogener Daten darin, dass es eine Beziehung zwischen der Personalisierung von Marketing, Dienstleistungen und Produkten einerseits und der Effektivität des Marketings andererseits gibt. Es hat sich gezeigt, dass die Verwendung persönlicher Social-Media-Netzwerkdaten - wie z.B. die Identifizierung von Verbrauchern, die mit früheren Kunden verbunden sind - die Effektivität von Direktmailing-Marketing-Kampagnen für einen Telekommunikationsdienst im Vergleich zu herkömmlichen Marketingansätzen um das Drei- bis Fünffache erhöht.^[32]

2.4 Docker

Docker ist eine „Freie Software, und gehört zu der Familie von PaaS-Produkten (Platform as a Service), die Virtualisierung auf Betriebssystemebene verwenden, das Ziel ist es Software in Paketen bereitzustellen, das Endresultat wird dann als Container bezeichnet.^[33] Docker vereinfacht das Deployment/Einsatz von Anwendungen, da Container, die alle notwendigen Pakete enthalten, einfach als Dateien übertragen und installiert werden können. Container sorgen für die Trennung und Verwaltung von Ressourcen, die auf einem Computer verwendet werden. Dazu gehören laut den Entwicklern: Code, Laufzeitmodule, Systemtools, Systembibliotheken – alles, was auf einem Rechner installiert werden kann. Platform as a Service: - Entwickler von PaaS-Plattformen, die Anwendungen schreiben, die in der Cloud ausgeführt werden können^[32] . Es handelt sich um eine Cloud-basierte Anwendungsentwicklung und -nutzung durch Anbieter und Entwickler. Es verfügt über eine hochgradig skalierbare Multi-Tier-Architektur wie z. B. Azure und salesforces.com. Der Unterschied zwischen PaaS und SaaS besteht darin, dass SaaS nur vollständige Cloud-Anwendungen hostet, während PaaS eine Entwicklungsplattform für fertige und laufende Cloud-Anwendungen bereitstellt. PaaS bietet eine Umgebung, in der Entwickler Anwendungen erstellen und bereitstellen können, ohne wissen zu müssen, wie viel Arbeitsspeicher und wie viele Prozessoren ihre Anwendungen verwenden werden. Das PaaS-Modell bietet Entwicklern Vorteile bei der Entwicklung komplementärer Software-Lebenszyklen, von der Planung, Gestaltung und Erstellung von Anwendungen bis hin zur Bereitstellung und Wartung. Das PaaS-Modell bietet eine höhere Abstrak-

tionsebene. Entwicklung von den Software-Lebenszyklus von der Planung über das Design und Konstruktion bis hin zur Erstellung von Anwendungsbereitstellungen und der Wartung. Das PaaS-Modell bietet eine höhere Abstraktionsebene. Bei diesem Modell erstellt der Verbraucher die Software unter Verwendung der Tools und/oder Bibliotheken des Anbieters. Verbraucher steuern auch die Softwarebereitstellung und Konfigurationseinstellungen. Anbieter stellen das Netzwerk, die Server, den Speicher und andere Dienste bereit, die zum Hosten von Verbraucheranwendungen erforderlich sind. PaaS-Angebote erleichtern die Bereitstellung von Anwendungen ohne die Kosten und die Komplexität des Kaufs und der Verwaltung der zugrunde liegenden Hardware und Software und der Bereitstellung von Hosting-Funktionen. Es gibt verschiedene Arten von PaaS-Anbietern, aber alle bieten Anwendungshosting- und Bereitstellungsumgebungen sowie verschiedene Integrationsdienste an. Containerisierung wird weiterhin als leichtgewichtige Virtualisierungslösung diskutiert.

Containerisierung gilt als leichtgewichtige Virtualisierungslösung. Container bieten nicht nur Vorteile gegenüber herkömmlichen virtuellen Maschinen in der Cloud, sondern eignen sich auch besonders gut für Platform as a Service (PaaS)-Clouds, um Anwendungen zu verwalten und zu orchestrieren, wobei Container als Mechanismus zur Kapselung der Anwendung verwendet werden.^[34]

Docker-Client und -Server

Docker kann als Client- und Server-basierte Anwendung interpretiert werden, wie in Abbildung 1 dargestellt. Der Docker-Server nimmt Anfragen von Docker-Clients entgegen und verarbeitet sie entsprechend. Die vollständige RESTful-API (Representational State Transfer) und Befehlszeilen-Client-Binärdateien werden von Docker bereitgestellt. Docker-Daemon/-Server und Docker-Client können auf demselben Computer ausgeführt werden, oder ein lokaler Docker-Client kann eine Verbindung zu einem Remote-Server herstellen oder Daemon läuft auf einem anderen Rechner.^[35]

Docker images

Es gibt zwei Möglichkeiten, Bilder zu erstellen. Die erste besteht darin, ein Bild mit einer schreibgeschützten Vorlage zu erstellen. Das Fundament jedes Bildes ist ein Basisbild. OS-Images sind im Grunde Basis-Images wie Ubuntu 14.04 LTS oder Fedora 20. Ein Betriebssystem-Image erstellt einen Container, der das laufende Betriebssystem sperrt. Das Basis-Image kann auch von heruntergeladen werden rasieren. Erforderliche Anwendungen können dem Basisimage hinzugefügt werden, indem Sie es ändern, aber es muss ein neues Image erstellt werden. Der Prozess des Aufbaus eines neuen Images wird als „Änderung vornehmen“ bezeichnet. Die zweite Methode besteht darin, eine Docker-File zu erstellen. Die Dockerfile enthält eine Liste mit Anweisungen, wenn der Befehl „Docker build“ vom Bash-Terminal ausgeführt wird, alle Anweisungen in der Dockerfile befolgt und ein Image erstellt. Dies ist eine automatische Bildkonstruktionsmethode.

Docker-Registrierungen

Docker Architecture

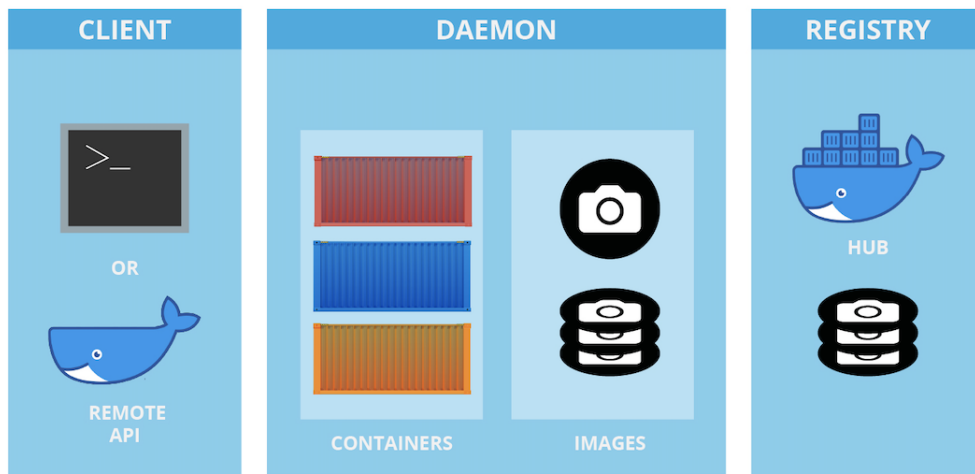


Figure 6: Der Docker-Client kommuniziert mit dem Docker-Daemon, der die schwere Arbeit beim Erstellen, Ausführen und Verteilen Ihrer Docker-Container übernimmt [56].

Docker-Images befinden sich in Docker-Registries. Es bietet entsprechende Quellcode-Repositories, in denen Bilder aus einer einzigen Quelle gepusht oder gezogen werden können. Es gibt zwei Arten von Registern, öffentliche und private. Docker-Hub ist als öffentliche Registrierung bekannt, in der jeder bereits vorhandene Images abrufen und seine eigenen pushen kann, ohne ein Image von Grund auf neu zu erstellen. Bilder können mithilfe der DockerHub-Funktion in einem bestimmten Bereich (öffentlich oder privat) verteilt werden.

Docker-Container

Docker-Image erstellt einen Docker-Container. Container enthalten alle für eine Anwendung erforderlichen Sätze(KIT), sodass die Anwendung separat ausgeführt werden kann. Angenommen, es gibt ein Ubuntu-Betriebssystem-Image mit SQL SERVER, wenn dieses Image mit dem docker run-Befehl ausgeführt wird, wird ein Container erstellt und SQL SERVER wird unter UbuntuOS ausgeführt. All diese genannten Komponenten sind Teil von Docker und die Unterteilung ist auf der Abbildung hier zu sehen 6.

Virtual Machine vs. Docker

Virtualisierung ist ein altes Konzept, das im Cloud Computing verwendet wird, nachdem IaaS als wichtige Technik für den Aufbau von Systemen, die Bereitstellung von Ressourcen und das mehrfache Leasing akzeptiert wurde. Virtualisierte Ressourcen

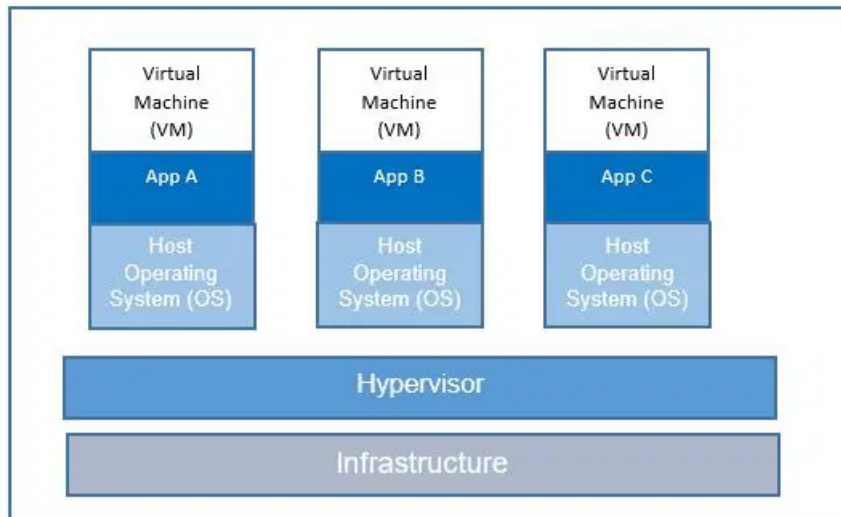


Figure 7: Der Hypervisor ermöglicht die Ausführung mehrerer VMs auf einer einzigen Maschine. Jede VM enthält eine vollständige Kopie eines Betriebssystems, der Anwendung, der erforderlichen Binärdateien und Bibliotheken – und belegt Dutzende von GB. VMs können auch langsam booten ^[57].

spielen eine Schlüsselrolle bei der Lösung von Problemen im Zusammenhang mit der Kerntechnologie Cloud Computing. Abbildung 7 zeigt die Architektur der virtuellen Maschine.

Der Hypervisor befindet sich zwischen dem Server- und dem Client-Betriebssystem. Es ist eine virtuelle Plattform und verwaltet mehr als ein Betriebssystem auf dem Server. Es arbeitet zwischen dem Betriebssystem und dem Prozessor. Virtualisierung teilt es in zwei Segmente: erstens ist Para-Virtualisierung und zweitens Vollvirtualisierung ^[36]. Die Docker-Containerarchitektur ist in Abbildung 8 dargestellt. Linux-Container werden von DockerTool verwaltet und auch als Virtualisierungsmethode auf Betriebssystemebene verwendet. Abbildung 8 zeigt, dass in einem einzelnen Steuerungsserver mehrere Linux-Container isoliert sind. Ressourcen wie Netzwerk, Speicher, CPU und E/A-Blöcke werden vom LinuxKernel zugewiesen und verarbeiten auch Pools, ohne die Virtualisierungs-Engine zu starten ^[37].

Laut Waldspurger (2002) dreht sich in der LinuxContainers-Architektur alles um das CPU-Management und die Ressourcenzuweisung. In jedem Fall von HyperV oder VMWare ist es aus Kostengründen nicht einfach, mehr als zehn VMs ^[40] zu betreiben. Mit Containern ist dieses Problem weitgehend gelöst. Container verwenden nur die Ressourcen, die für Dienste oder Anwendungen erforderlich sind. Somit können auf einer schwach konfigurierten Maschine mehr als 50 Containeranfragen gestellt werden. Angenommen, eine Organisation bietet E-Mail-Sicherheitsdienste an. Die Hauptfunktionen dieser Dienste bestehen darin, E-Mails auf Viren, Spam und Malware zu überprüfen. Darüber hinaus kann es Nachrichten an den Händler weitergeben, Lieferfehler protokollieren und melden, wenn das Produkt in der Cloud installiert ist ^[38]. Normalerweise

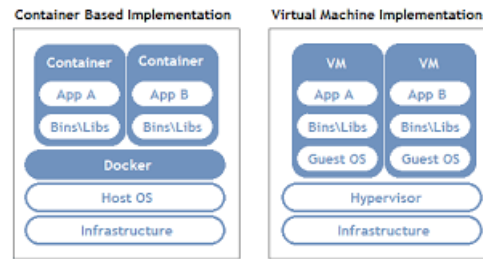


Figure 8: Container und VMs, die zusammen verwendet werden, bieten ein hohes Maß an Flexibilität bei der Bereitstellung und Verwaltung von Apps^[57].

wird in diesen Fällen je nach Betriebssystem, Bibliothek oder Kernel-Datenstruktur keine Verbindung verwendet. Daher verdient jede Container-Komponente eine Sandbox mit OpenVZ oder Docker anstelle einer virtuellen Maschine. In vielen Unternehmen werden virtuelle Maschinen zur Durchführung von Artikelprüfungen eingesetzt. Bei diesem Vorgang werden viele CPU- und Speicherressourcen verwendet. Die Container-Technologie bietet dem Benutzer die Gewissheit, dass die Überlastung der Arbeitslast die Effizienz der Ressourcennutzung nicht beeinträchtigt. Container benötigen weniger Zeit für die Installation als virtuelle Maschinen, daher ist die Anpassungsfähigkeit von Containern viel höher als bei VMS.

Darüber hinaus wurden Docker und OpenVZ hinsichtlich ihrer Sicherheitsaspekte genau überwacht. Wenn die Isolation abnimmt, wirkt sich dies direkt auf die Sicherheit aus, die ebenfalls schnell abnimmt. Linux-Root-Benutzer können problemlos auf die Container zugreifen, da die Container auch denselben Kernel und dasselbe Betriebssystem verwenden. Die Docker-Isolierung ist nicht so leistungsfähig wie eine virtuelle Maschine, obwohl Docker eine isolierte Anwendung ist, die in DockerContainer von ihrem Haupthost aus ausgeführt wird. Darüber hinaus können einige Anwendungen möglicherweise nicht in Containertechnologie ausgeführt werden und müssen auf einem anderen Betriebssystem ausgeführt werden.

Vorteile von DockerContainer

Sehen Sie sich die Anforderungen und die Entwicklung von Linux-Containern in den letzten Jahren an. Docker hat dank der Vorteile, die Docker-Container bietet, sehr schnell an Popularität gewonnen. Die Hauptvorteile von Docker sind Geschwindigkeit, Portabilität, Skalierbarkeit, schnelle Verteilung und Dichte.

Geschwindigkeit

Geschwindigkeit ist einer der herausragendsten Vorteile von Containern. Wenn die Vorteile der Verwendung von Docker hervorgehoben werden, wäre es erstaunlich, die Geschwindigkeit von Docker im Gespräch nicht zu erwähnen ^[58]. Die Verpackungszeit ist sehr schnell, weil sie wirklich klein sind. Entwicklung, Tests und Bereitstellung können

schneller erfolgen, da der Container klein ist. Container können nach dem Bau zum Test und von dort in die Produktionsumgebung ^[39] gepusht werden.

Portabilität

Die in Docker Containers integrierten Anwendungen sind extrem portabel. Diese mobilen Apps können einfach als einzelnes Element verschoben werden, und die Leistung bleibt gleich ^[39].

Skalierbarkeit

Docker kann auf mehreren physischen Servern, Datenservern und Cloud-Plattformen bereitgestellt werden. Es kann auch auf jedem Linux-Computer ausgeführt werden. Container können einfach aus der Cloud-Umgebung auf den lokalen Server und von dort blitzschnell in die Cloud verschoben werden. Die Anpassung ist einfach zu integrieren. Der Benutzer kann die Skala nach Bedarf leicht anpassen ^[41].

Schnelle Lieferung

Das Format von Docker-Containern ist standardisiert, sodass sich Entwickler nicht um mehr Aufgaben als nötig kümmern müssen. Der Administrator ist für die Bereitstellung und Wartung der Server mit den Containern verantwortlich, während sich der Programmierer um die Anwendungen im DockerContainer kümmert. Container können in jeder Umgebung funktionieren und bereitgestellt werden, da sie alle erforderlichen Abhängigkeiten in die Anwendungen eingebaut haben und alle überprüft werden ^[39]. Docker bietet eine zuverlässige, konsistente und fortschrittliche Umgebung, um vorhersehbare Ergebnisse zu erzielen, wenn Code zwischen Entwicklungs-, Test- und Produktionssystemen geteilt wird ^[58]

Dichte

Docker nutzt die verfügbaren Ressourcen effizienter, da es keinen Hypervisor verwendet. Aus diesem Grund können auf einem einzelnen Host mehr Container ausgeführt werden als auf einer virtuellen Maschine. Die Leistung von DockerContainer ist aufgrund der höheren Dichte und der Vermeidung von Ressourcenverschwendung höher^[41].

Nachteile von Docker :

- Virtualisierung wird von Docker überhaupt nicht bereitgestellt, da sie vom Linux-Kernel abhängt, der von localhost bereitgestellt wird.
- Docker läuft derzeit nicht auf älteren Rechnern. Es unterstützt nur lokale 64 -Bit-Computer.
- Eine vollständige Virtualisierungsumgebung muss von DockerContainer für Windows-

und Mac-Maschinen bereitgestellt werden. Selbst wenn boot2dockerTool diese Schwachstelle behebt, lohnt es sich dennoch zu prüfen, ob es die Benutzerakzeptanz dieser Systeme behindert oder ob die Integration und Leistung mit dem Host-Betriebssystem angemessen ist. [36]

- Die Möglichkeit von Sicherheitsproblemen sollte bewertet werden. Das Erstellen vertrauenswürdiger Binärdateien kann vereinfacht werden, indem Docker-Images für die zukünftige Unterstützung digital signiert werden.
- Eine große Sorge ist, ob die technische Gemeinschaft oder wissenschaftliche Forscher ernsthaft erwägen, Docker einzuführen.

Laut der Arbeit von Amit M , Narayan D G, Shivaraj Kengond, Mohammed Moin Mulla, in Performance Evaluation of Docker Container and Virtual Machine. [58].dockeramit In diesem Abschnitt erfolgt die Auswertung von KVM und Docker mittels Benchmarking-Tools. Die beiden HP-Server werden zur Leistungsbewertung in Bezug auf verschiedene Parameter verwendet, für diesen wird ein Server als virtuelle Maschine verwendet, die auf dem Host-Betriebssystem installiert wird und während das Gastbetriebssystem mit Zusatz zu dieser Docker-Engine auf der virtuellen Maschine installiert ist und ein weiterer Server als Bare-Metal mit Host-Betriebssystem Ubuntu 16.04 und Docker-Engine ist darauf installiert. Ubuntu 16.04 64-Bit mit Linux-Kernel 3.10.0 wurde verwendet, um alle Tests durchzuführen. So halten Sie die Konsistenz aufrecht und Einheitlichkeit, das gleiche Betriebssystem, Ubuntu 16.04, wurde als Basis-Image für alle Docker-Container verwendet.

CPU-Leistung

Die Ergebnisse hängen hauptsächlich von der Anzahl der virtuellen CPU co-Res, die dem Server zugeordnet sind. Der Vergleich der CPU-Leistung wurde über die den Tools sysbench, Phoronix und Apache benchmark gemessen/ bewertet. Hier in der Abbildung zu sehen 9 Es wird beobachtet, dass der Docker-Container viel weniger Zeit benötigt, um die Operation auszuführen, wenn im Vergleich zu VM.

Maximaler Primzahlvorgang

Auf dem Sysbench-Tool-Test durchgeführt, um herauszufinden, welche Zeit benötigt wird, um die maximale Primzahl auszuführen.

7 Zip Kompressionstest

Dieser Test misst die Zeit, die benötigt wird, um eine Datei mit 7 Zip-Komprimierungen zu komprimieren. Daher entsprechend den Ergebnissen erhalten, Docker Container-Leistung ist viel besser als VM, wenn es mit einer großen Menge an Dateien durchgeführt wird Kompression. Die Abbildung 10 zeigt die Speicherleistung in Bezug auf den RAM-Geschwindigkeits-SMP-Test.

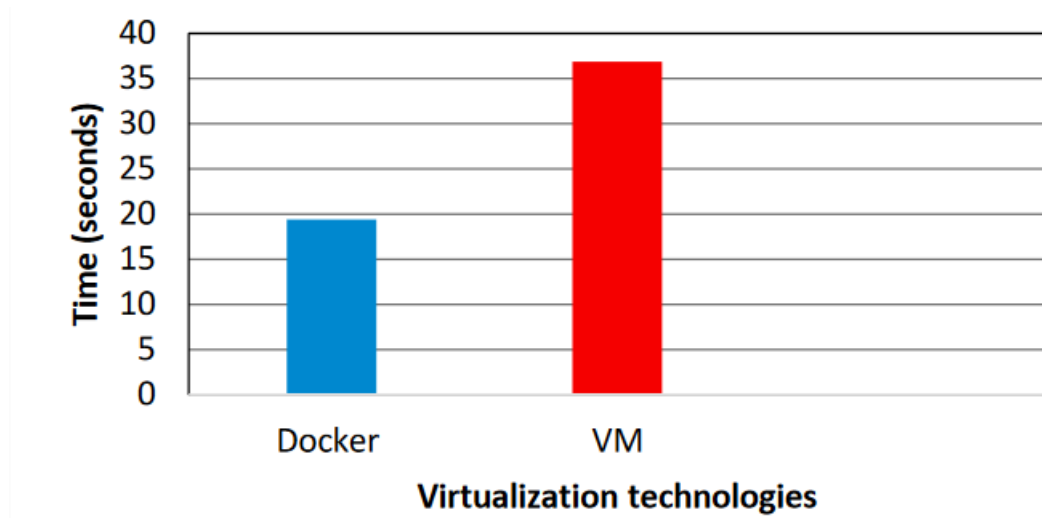


Figure 9: CPU-Vergleich von Docker und Virtual Machine gezeigt in Sekunden [58].

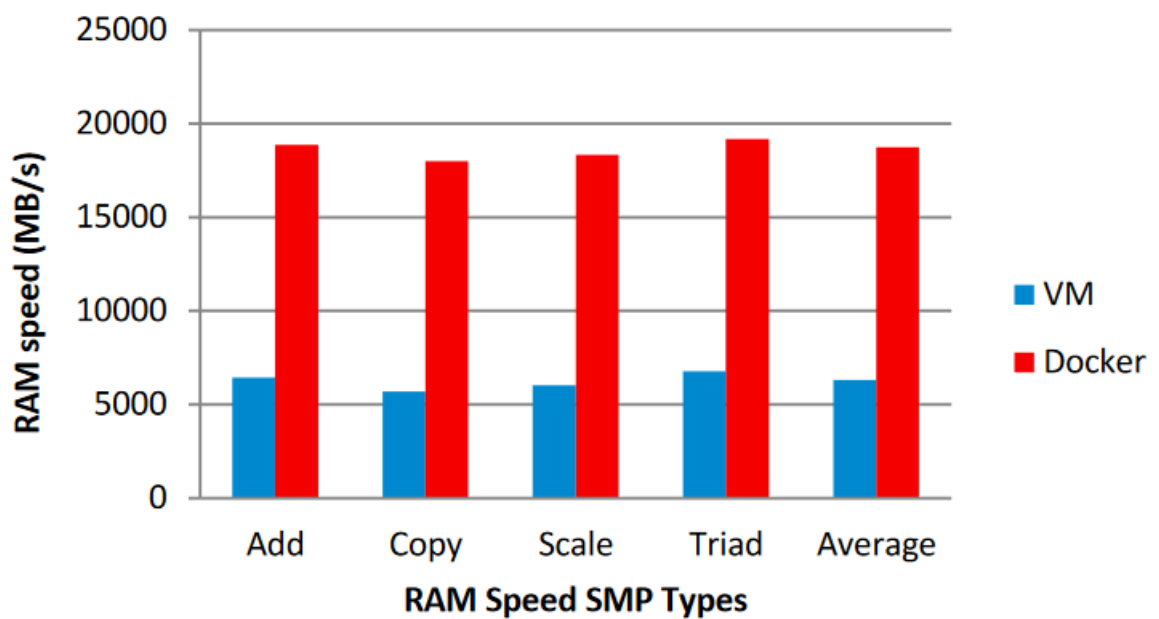


Figure 10: RAM-Geschwindigkeitsvergleiche zwischen Virtualisierungstechnologien [58].

Speicherleistung

RAM Speed/SMP ist ein Cache- und Speicher-Benchmarking-Tool, das für die Messung der RAM-Geschwindigkeit für Virtualisierungstechnologien, d. Docker und Virtual Machine. Jeder besteht aus vier Untertests, um verschiedene Aspekte des Speichers zu messen Leistung. Die Übertragung von Daten von einem Speicherort zum anderen erfolgt per Kopierbefehl.

Die Daten werden vom ersten Speicherort und dann vom zweiten Speicherort gelesen, wenn der Befehl ADD aufgerufen wird. Dann die resultierenden Daten werden an dritter Stelle platziert. Die Daten werden vom ersten Speicherort gelesen, um zu skalieren, und dann von der zweiten Stelle hinzugefügt, um sie an die dritte Stelle zu schreiben.

Die Leistung von Docker im Vergleich zur virtuellen Maschine viel besser ist. Der Datenträger schreibt und Die Lesevorgänge einer VM werden um mehr als die Hälfte der Vorgänge eines Docker-Containers reduziert. Zeigt die Festplattenleistung von virtuellen Maschinen und Docker an.

Lasttests/load testing

zeigt, dass die Durchsatzanalyse für VM im Vergleich zu Docker viel geringer ist. Die Analyse zeigt, dass Docker Container ist besser als die virtuelle Maschine bei der Verarbeitung der Anzahl der Anforderungen pro Sekunde.

Messung der Betriebsgeschwindigkeit

Maschinen. Basierend auf der Ausführungszeit benötigt der Docker-Container weniger, um das Problem zu lösen, während der virtuelle die Maschine benötigt eine viel längere Zeit. Unter Verwendung des leeren Raumes wird die Anordnung der die Anzahl der Kacheln, die der endgültigen Konfiguration entsprechen, ist das Hauptziel. Es kann vier benachbarte Operationen Kacheln in den leeren Raum - der Test misst, wie lange es dauert, das Problem zu lösen.

zeigt die Rechenleistung sowohl von Docker als auch von virtuellen Maschinen. Basierend auf der Ausführungszeit, Docker-Container es dauert weniger, um das Problem zu lösen, während die virtuelle Maschine viel länger dauert.

2.5 Data Ingestion Pipeline

Datentransformation Datentransformation ist ein Vorgang, bei dem die Struktur bestehender Daten verändert (transformiert) wird. Die wichtigsten Formen der Datentransformation sind^[23]: Clean Korrigieren, löschen oder ändern Sie falsche oder beschädigte Zeilen.

Aggregation Aggregiert Daten, um eine zusammenfassende Version eines bestimmten Satzes zu erhalten. Berechnen Sie beispielsweise die Anzahl der Transaktionen gruppiert

nach verschiedenen Regionen und Kategorien.

Berechnungen - Berechnen Sie neue Werte und Zahlen auf Basis von Geschäftsformeln und wandeln Sie Rohdaten in Geschäftskennzahlen um.

Die aus der Analyseplattform extrahierten Rohdaten liegen hauptsächlich in Form einer Transaktion vor, was bedeutet, dass der Datensatz eine Geschäftstransaktion darstellt. Um diese Daten für die Analyse zu verwenden, müssen sie in eine komfortablere Form umgewandelt werden. Der Prozess der Umwandlung roher Transaktionsdaten in einen verarbeiteten Satz, der für die Analyse geeignet ist, wird als Datenmodellierung bezeichnet [23].

Hauptgründe, warum Unternehmen Datentransformation benötigen[23]:

- Wiederverwendung - Jeder Schritt der Datentransformation kann als separate Komponente mit einer bestimmten Geschäftslogik vorgestellt werden, die auf mehrere Stellen in verschiedenen Abfragen und Berichten angewendet werden kann.
- Konsistenz des Berichts - Aufgrund der angegebenen Wiederverwendung werden Transformationen nur an einer Stelle geschrieben und können wiederverwendet werden. Dies führt zu einer größeren Konsistenz der verschiedenen Berichte. Dadurch wird verhindert, dass zwei Berichte aufgrund der Art und Weise, wie sie transformieren, unterschiedliche Zahlen für dieselbe Metrik anzeigen.
- Bessere Leistung – Datentransformations- und Aggregationsprozesse werden nur einmal für einen bestimmten Quellsatz ausgeführt, und Berichts- und Analyseprozesse über transformierte Daten können mehr als einmal ausgeführt werden. Allein die Tatsache, dass dieses Set gereinigt, aggregiert und bereit für Analysen ist, führt zu einer erheblichen Verbesserung der Leistung.
- Wirtschaftliche Rentabilität - Die Tatsache, dass der transformierte Satz reguliert ist, bedeutet, dass endgültige Abfragen und Berichte weniger Prozessorleistung verbrauchen, was niedrigere Kosten bedeutet.

Die beiden am häufigsten verwendeten Paradigmen für die Datentransformation sind ETL (Extrahieren, Transformieren und Laden) und ELT (Extrahieren, Laden und Transformieren) [23]. Die Struktur einer typischen Plattform-Pipeline ist in Abbildung 11. dargestellt.

ETL Prozesse

In den meisten Unternehmen, die über eine Analyseplattform verfügen, ist der intensivste Schritt die Datenaufbereitung, bei der Datensätze kombiniert, bereinigt und erstellt werden, die den Endbenutzern präsentiert werden können [23]. Dieser Vorgang wird als EXTRACT, Transformation und Load bezeichnet. Im ETL-Prozess werden Daten aus den Quellsystemen extrahiert, dann werden verschiedene Transformationen über diesen Satz vorgenommen, so dass schließlich bearbeitete und transformierte Sätze in den Datenspeicher geladen werden. Die drei Schritte des ETL-Prozesses sind [23]

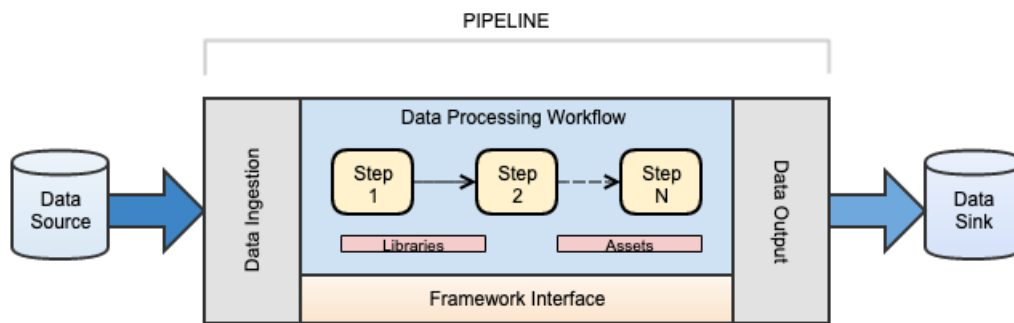


Figure 11: Pipelines werden als klassische Software-Pipeline-Anwendung implementiert, bei der eine Reihe von Datenverarbeitungselementen in einer wiederverwendbaren Pipeline-Komponente gekapselt wird. Jede Pipeline soll Eingabedaten in Streams oder Batches verarbeiten und die Ergebnisse an einen angegebenen Zielkatalog ausgeben. [59].

- Extrahieren (Extrahieren) von Daten aus einer Quelle (Schreiben Sie eine SQL-Abfrage, die in Produktionsdatenbanken ausgeführt wird, oder ziehen Sie ASON-, XML- oder CSV-Dateien).
- Transformieren (transformieren) von Daten im ETL-Serverspeicher.
- Laden (Laden) transformierter Daten in Speicher.

Umwandlung aus dem Speicher Das Transformieren von Daten außerhalb des Speichers und das Laden nur bereinigter, transformierter und bearbeiteter Daten in den Speicher ist ein Prozess, der dem oben genannten ETL-Paradigma folgt, bei dem Daten zuerst extrahiert, dann transformiert und schließlich in ihr endgültiges Ziel geladen werden. Betrachten wir drei wichtige Schritte im ETL-Prozess, die für den Prozess der Erstellung eines dimensionalen Modells im Datenspeicher spezifisch sind:

1. **Quelldatenextraktion** - Im Falle unserer Analyseplattform befinden sich die meisten Rohdaten bereits im Cloud-Speicher (Cloud Storage), der zur Hauptquelle wird, während der andere Teil der Daten aus gestreamten Daten bestehen kann, die wir direkt von Messaging-Diensten wie Cloud Pub / Sub lesen können, so dass Ereignisdaten den See überspringen können, um so früh wie möglich im Datenspeicher zu erscheinen und für die Analyse für Endbenutzer verfügbar zu sein.
2. **Datentransformation** - Dies ist ein wichtiger Schritt beim Erstellen eines dimensionalen Modells, da hier Daten aus verschiedenen Quellen bereinigt und vorbereitet werden, damit sie sich verbinden und Datensätze erstellen können, die schließlich in Dimensions- und Faktentabellen landen. Der Transformationsschritt besteht eigentlich aus einer Reihe kleinerer Verarbeitungsschritte, die mit mehreren verschiedenen Cloud-Datenverarbeitungsdiensten ausgeführt werden können.

3. Laden von Daten in den Speicher - Nur verarbeitete und gefilterte Daten werden am Ende des Prozesses in den Datenspeicher geladen, wobei Dimensions- und Faktentabellen ausgefüllt werden. Auf diese Weise enthält der Datenspeicher nur verarbeitete Daten, die durch ein dimensionales Modell dargestellt werden, und Rohdaten sind im Data Lake enthalten.

Angeichts der großen Anzahl unterschiedlicher Quellen, aus denen Daten stammen können, sowie der Menge und Geschwindigkeit dieser Daten sowie der gegenseitigen Abhängigkeit von Transformationsschritten kann dieser Prozess sehr komplex werden. Der Prozess, bei dem Daten durch verschiedene Teile des Systems, von der Quelle zum Ziel, fließen und ihre Struktur und ihr Format ändern, wird oft als Datenpipeline bezeichnet [22]. Datenfeeds sollten konfiguriert werden: Es werden genaue Neustartzeiten festgelegt (jeden Tag, jede Stunde), die Interdependenz der Schritte im Flow bestimmt, so dass Sie die genaue Reihenfolge kennen, in der die Schritte ausgeführt werden können. Es gibt zwei grundlegende Ansätze zum Erstellen und Konfigurieren einer Datenzeile: Konfiguration durch Code und Konfiguration durch eine grafische Oberfläche.

Eine weitere beliebte Möglichkeit, eine Pipeline für das Streaming von Daten durch das System zu erstellen, ist die Verwendung eines Dienstes, der eine grafische Oberfläche bietet, um verschiedene Dienste zu integrieren und einen vollständigen Interdependenzgraphen zu erstellen. Dieser Ansatz wird oft als codeless Pipeline bezeichnet, denn im Gegensatz zur Herstellung einer Pipe mit Tools wie Airflow, bei der die gesamte Konfiguration durch Code definiert wird, sind bei diesem Ansatz Programmiersprachenkenntnisse nicht erforderlich und alles wird über die Schnittstelle erledigt.

Google Cloud bietet einen Cloud-Data-Fusion-Service, der die Integration verschiedener Dienste, die Erstellung und Verwaltung von Datenpipelines auf CDAP-Basis (Complex Data Pipelines) ermöglicht [54]. Dieser Service bietet eine grafische Oberfläche, die darauf abzielt, die Effizienz bei der Erstellung einer Datenzeile zu steigern und die Komplexität der Erstellung einer Pipeline zu reduzieren. Die Hauptidee solcher Dienste besteht darin, die Aufgabe der Datenerstellung zu automatisieren und dass einige Teile des Auftrags, die normalerweise einen Datenverarbeitungsingenieur erfordern, den Dienst selbst übernehmen und beenden. Mit dieser Art von Service ist die Erstellung eines Datenintegrationssystems nicht mehr auf Ingenieure beschränkt, sondern Benutzer mit weniger technologischer Erfahrung können ihre eigenen Extraktions-, Transformations- und Datenladeprozesse erstellen, so dass sie im Speicher landen und bereit sind, in einem der Business Intelligence-Tools angezeigt zu werden [54]. Das Paradigma der Konfiguration der Pipe durch Code ist häufiger in Tech-Start-up-Unternehmen, in denen die Möglichkeit, sich zu ändern und an spezifische Anforderungen anzupassen, sehr wünschenswert ist, ebenso wie die Wirtschaftlichkeit der Lösung selbst [52]. Das Paradigma der Konfiguration der Pipe über die Schnittstelle ist in der Unternehmenswelt weiter verbreitet, wo Unternehmen, die möglicherweise nicht in erster Linie technologisch sind und nicht über ausreichend Personal verfügen, die Möglichkeit erhalten, eine Pipe einfach zu erstellen, ohne Code zu schreiben, auf Kosten eines teureren Preises und weniger Flexibilität [52]. In der Vergangenheit war die Gestaltung eines Data Warehouse sowohl aus Software- als auch aus Hardwaresicht ein sehr teurer Prozess [23]. Die

Kosten für Server, Implementierung und Softwarelizenzen für ein Datenspeicherprojekt vor 20 oder 30 Jahren belaufen sich selten auf Millionen von Dollar, und die Implementierung dauert oft Monate oder Jahre [23]. Bei teilweise sehr hohen Preisen müssen Unternehmen auf die Kosten achten und nur bereinigte, sauber transformierte und aggregierte Daten ins Warehouse laden. Damals war die Welt der Softwareentwicklung noch das dominierende Prinzip des Wasserfalls, daher war es akzeptabel, sich zu Beginn des Projekts die Zeit zu nehmen, die notwendigen Datentransformationen zu planen, die durchgeführt werden sollen, bevor die Daten endgültig im Speicher sind. [?] Aus dieser Perspektive ist der ETL-Prozess, der nur verarbeitete Daten in den Datenspeicher lädt, am sinnvollsten.

Einige Dinge haben sich in der Welt der ELT-Technologie dramatisch verändert und Prozesse, bei denen Daten nicht vollständig verarbeitet wurden, gewinnen heute an Popularität. Änderungen, die einen großen Einfluss auf die Verbreitung des ELT-Verfahrens hatten, sind [23]:

- **erfügbarkeit und Wirtschaftlichkeit der Datenspeicherung in der Cloud** Moderne Datenspeicherung ist heute in der Lage, große Datenmengen zu relativ geringen Kosten zu speichern und zu verarbeiten.
- **Die Explosion der Menge und Vielfalt der gesammelten Daten** Die Big-Data-Revolution hat zur massiven Entwicklung neuer Tools und Technologien geführt, die in der Lage sind, Datensätze mit mehreren Terabyte zu verarbeiten.
- **Agile Softwareentwicklung verbreiten** Dies bedeutet, dass Analyseteams den Prinzipien der agilen Entwicklung folgen, anpassungsfähig und bereit sein müssen, schnell zu iterieren.

Ähnlich wie bei OutsideStorageTransformations gibt es, wenn wir uns für die Verwendung von Warehouse-Transformations entscheiden, zwei Möglichkeiten, diese Transformationen durchzuführen und die Daten vollständig zu modellieren: Modellierung durch Code und Modellierung durch die Schnittstelle. Der Prozess der Erstellung eines Datenmodellierungssystems in einem Speicher ist dem Prozess der Erstellung eines ETL-Systems sehr ähnlich, um Daten aus dem Speicher zu transformieren. Einige Transformationsorchestrierungssoftware wird weiterhin verwendet, in unserem Fall Cloud Composer, und der Unterschied besteht in Diensten, die konkrete Transformationen, Aggregationen, Deduplikate und andere Datenmanipulationen durchführen. Im ETL-System war Cloud Composer der Hauptvermittler zwischen einer Reihe verschiedener Verarbeitungsdienste und für deren Integration verantwortlich. Im ELT-System erfolgt die gesamte Datenverarbeitung innerhalb des Speichers, was bedeutet, dass in unserem Fall BigQuery, ein moderner Cloud-Datenspeicher, der einzige Dienst ist, der für die Verarbeitung der Daten verantwortlich ist. Die einzelnen Schritte in diesem Prozess sind eigentlich Transformationen über Speichertabellen, nämlich SQL-Abfragen, die über Cloud Composer orchestriert und mit Hilfe eines BigQuery-Verarbeitungsservers durchgeführt werden. Separate SQL-Abfragen, die Verarbeitungsschritte darstellen, weisen bestimmte Interdependenzen auf, die durch den im Python-Skript definierten direkten azyklischen Graphen

dargestellt werden, unter dem Cloud Composer weiß, in welcher Reihenfolge die Transformationen ausgeführt werden sollen.^[23]

Der ELT-Ansatz reduziert die Komplexität der gesamten Dateizeile etwas, da die gesamte Verarbeitung über SQL-Abfragen innerhalb nur eines Dienstes (BigQuery) erfolgt. Dieser Ansatz ermöglicht die Erstellung neuer Modelle und Transformationen für mehr Mitglieder der Organisation, da SQL zu einem weit verbreiteten Standard geworden ist. Der ELT-Ansatz weist auch bestimmte Mängel auf, wie bereits erwähnt. SQL-Sprache bringt viele Vorteile, aber die Verarbeitung von Daten ohne die richtige Programmiersprache mit Grundprinzipien der Modularität, Wiederverwendung und Testfunktionen ist ein Fehler in diesem Ansatz. ^[23]

2.6 CAP Theorem

Das CAP-Theorem beschreibt eine Eigenschaft eines verteilten Datenbanksystems. Die Abkürzung „CAP“ steht für die englischen Begriffe „Consistency“, „Availability“ und „Partition Tolerance“. Diese drei Begriffe bedeuten im Deutschen „Konsistenz“, „Usability“ und „Fault Tolerance“.

Konsistenz im CAP-Theorem beschreibt die Konsistenz von Daten in diesem Zusammenhang. Das bedeutet, dass alle Clients in einem verteilten System gleichzeitig dieselben Daten sehen können.

Usability bedeutet im CAP-Theorem eine akzeptable Reaktionszeit des Systems. Die akzeptable Dauer hängt manchmal auch vom Benutzer des Systems ab. Einige Benutzergruppen warten möglicherweise länger als andere.

Fehlertoleranz im CAP-Theorem bedeutet in diesem Zusammenhang, dass das System auch bei Ausfall einzelner Knoten stabil bleibt.

Das CAP-Theorem verteilter Datenbanksysteme betrifft die Tatsache, dass diese drei Eigenschaften nicht gleichzeitig implementiert werden können. Bis zu zwei der drei Eigenschaften können erfüllt werden.

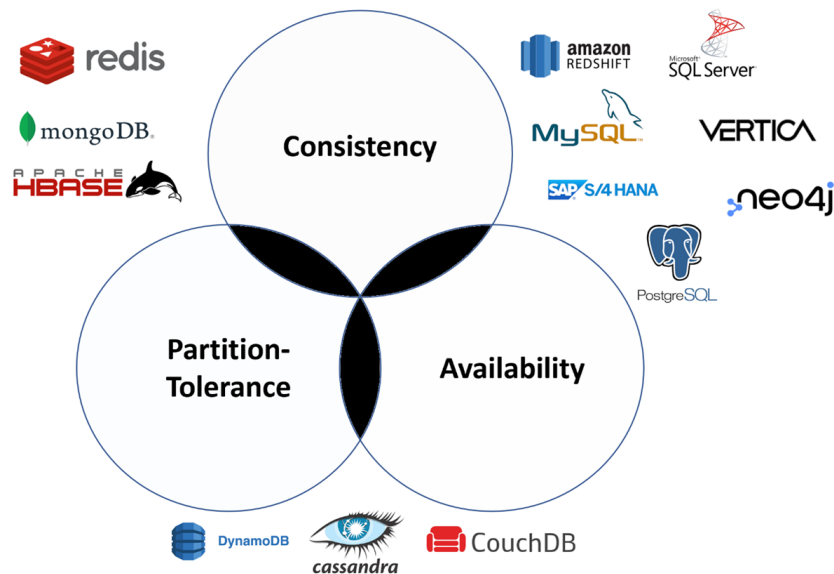
Struktur und Funktion des CAP-Theorems:

Das CAP-Theorem hat die gleiche Struktur wie ein Dreieck. Jeder Knoten repräsentiert eine der drei Eigenschaften des Systems. Da es nicht möglich ist, alle drei Eigenschaften gleichzeitig zu erreichen, muss beim Aufbau des Systems eine Entscheidung getroffen werden, die beiden Eigenschaften zu kombinieren. Dies führt zu CP-, AP- oder CA-Systemen.

Da in den meisten Bereichen eine hohe Systemverfügbarkeit erwünscht ist, kommen CP-Systeme in der Praxis selten vor. Bei diesen Systemen bedeutet die Netzwerkpartitionierung, dass der Teil des Netzwerks, der konsistent gehalten werden kann, immer noch verfügbar ist, aber nicht der Rest.

CA-Systeme sollen die Konsistenz und Verfügbarkeit eines Systems sicherstellen zu sehen in der Abbildung12.

AP-Systeme streben nach Verfügbarkeit und Fehlertoleranz. Da diese nicht immer konsistent sind, reagiert das System auf Netzwerkpartitionen inkonsistent. Das System antwortet auf Anfragen, aber die Antworten sind nicht notwendigerweise konsistent.

Figure 12: Darstellung von dem CAP Theorem^[60]

Die theoretische Beschreibung des CAP-Theorems ist eine idealisierte Definition, die in der Praxis nicht vollständig realisiert werden kann, weil die Grenzen einzelner Eigenschaften, ob sie sich treffen oder nicht, oft verschwimmen oder Systeme miteinander vermischt werden.

Das CAP Theorem angewendet auf die gegebene Datenbankschema ergibt sich eine Auswahl auf die SQL Datenbank. Durch die SQL Datenbank die in dem Projekt benutzt wurde ist das Theorem auf das CA System angepasst. Durch die Daten die sich fast nie ändern außer es kommt zu einer Änderung des Fragebogens. Für die momentane Version des Fragebogens sind die Fragen biss zum nächsten Update nicht veränderlich, da die Daten Konsistenz mit der aktuellen Version gewährleistet ist. Damit können alle Daten in der Datenbank gleichzeitig aktualisiert werden. Wie bereits in den Risiken besprochen eine mögliche Änderung des „Red Flag counters“ in dem Fragebogen könnte erst in der neuen Version des Projektes eine Änderung an der „Data Ingestion pipeline“ Code Struktur fordern. Durch die ständige Verfügbarkeit der Daten die durch die SQL Datenbank gewährleistet wird eignet sie sich gut für die Funktionalität des Fragebogens ganz gut. ^[42]

2.7 Stand der Dinge in der Digitalisierung im Gesundheitswesen

Die Digitalisierung erfasst zunehmend alle Bereiche des täglichen Lebens und ist aus dem Alltag nicht mehr wegzudenken. Die rasante Entwicklung neuer Technologien hat in vielen Branchen zu drastischen Veränderungen geführt. Unternehmen erweitern zunehmend ihre Netzwerke und suchen über die traditionelle Vertriebskanäle hinaus zunehmend nach digitaler Kundenreichweite für das Produkt- und Dienstleistungsmarketing. ^[43] Seit An-

fang der 2000er ist der Einzug der digitalen Technik in die unterschiedlichsten Lebensbereiche ein besonderes Thema. Hauptsächlich die Entwicklung des deutschen Gesundheitssystems gilt als weit hinterher gebliebene im Vergleich zu internationalen Vorbildern wie z.B.: Dänemark ^[44]. In Deutschland werden die Potenziale zur Digitalisierung des Gesundheitswesens im Vergleich zu anderen gesellschaftlichen Bereichen bisher zu wenig geschöpft. Im internationalen Vergleich liegt Deutschland bei der Digitalisierung im Gesundheitswesen derzeit sehr weit hinten. Daher sind erhebliche Änderungen zu erwarten. ^[44]Eine der sichtbarsten Auswirkungen der Digitalisierung von Gesundheitsdiensten für Bewohner und Patienten ist die Möglichkeit, Patientendaten in Form von elektronischen Patientenakten digital zu speichern und die Möglichkeit der digitalen Kommunikation. Seine bundesweite Einführung in Deutschland beschrieb die damalige Gesundheitsministerin Ulla Schmidt 2003 im Gesetzentwurf zur Modernisierung der gesetzlichen Krankenversicherung ^[44] Die Digitalisierung fördert auch weitere Innovationen im Gesundheitswesen. Internetzugang und Smartphone-Nutzung sind für die Deutschschweizer Bevölkerung heutzutage selbstverständlich. Die digitalen Infrastrukturen wie elektronische Patientenakten (EPCs) befinden sich immer noch in der Entwicklungsphase. Die Digitalisierung Dienstleistungen im deutschen Gesundheitswesen breiten sich stark aus. Die Medizinbranche stehen der Digitalisierung der ambulanten Versorgung grundsätzlich positiv gegenüber. Allerdings sind es nur rund 50 niedergelassene Schweizer Ärzte die eine elektronische Patientenakte führen. Dadurch können jährlich rund 3 Milliarden Euro eingespart werden. ^[45] Das entspricht etwa 12 Prozent der jüngsten jährlichen Gesamtkosten für Gesundheit und Pflege von etwa 290 Milliarden Euro. Elektronische Patientenakten und elektronische Rezepte sowie webbasierte Arzt-Patienten-Interaktionen bieten das größte Potenzial. ^[46] Die bisherige Trägheit der digitalen Transformation im Gesundheitswesen ist zurückzugreifen auf die strengen Auflagen des Gesetzgebers als auch durch die datenschutzrechtlichen Sicherheitsbedenken. Ein weiteres Problem ist die sogenannte Cyber-Kriminalität. Die Patientendaten enthalten sehr sensible Informationen, die nicht in unbefugte Hände geraten dürfen. Im Gesundheitswesen muss durch entsprechende Sicherheitsmaßnahmen dafür Sorge werden, dass die Daten aller Patienten geschützt werden. Die digitale Transformation im Gesundheitswesen fordert eine präzise Umsetzung der datenschutzrechtlichen Auflagen.^[47]

V.Stephani, R.Geissler und A.Busse vergleichen den internationale Stand der Krankenhaus-IT. Dazu stellen sie das folgendes Modell EMRAM (Electronic Medical Record Adoption Model) vor. Durch das EMRAM wird der aktuelle Digitalisierungsgrad in einem Krankenhaus ermittelt. In Deutschland gibt es insgesamt 2000 Krankenhäuser. Seit 2014 haben sich davon insgesamt 167 Krankenhäuser dieser Prüfung unterzogen. Etwa 40 Prozent der Krankenhäuser sind überhaupt nicht digitalisiert, lediglich zwei Häuser erreichen auf der 7-stufigen Skala die Stufe 6 und nur eines, das Universitätskrankenhaus Eppendorf, erreichte temporär die höchste Stufe. Die Autoren kommen zu dem Ergebnis, dass die Digitalisierung bei deutschen Krankenhäusern noch am Anfang ist.^[44]

Im Vergleich dazu hat sich die dänische Gesundheitspolitik zu einer grundlegenden Neuausrichtung des Krankenhaussystems entschlossen. Dafür wurde die Anzahl der bestehenden Krankenhäuser deutlich reduziert und eine Großzahl neuer hochgradig spezialisierte Krankenhäuser eingerichtet. Die bestehende Notfallversorgung wurde grundsätzlich

umgestaltet. Im Zusammenhang mit der Umgestaltung spielt die umfassenden Digitalisierung des Gesundheitswesens eine wesentliche Rolle. Das landesweite gesundheitsbezogene Datennetzwerk soll nun auch die Leistungserbringer außerhalb des Krankenhauses und die persönlichen gesundheitsbezogenen Daten der Bürger einbinden. Als Ziel wird ist ein vollumfängliches Management sämtlicher gesundheitsbezogener Daten formuliert. Neben den Ärzten ist auch das Pflegepersonal besonders von der Digitalisierung betroffen. Dieses Themengebiet wird von Fachinger und Mähr untersucht. [44] Die Autoren betonen das erheblichen Potenziale einer stärkeren Digitalisierung in der Pflege. Die Informations- und Kommunikationstechnologien als auch die Robotik und Assistenzsystemen sind noch weitgehend ungenutzt. Die angestrebte Digitalisierung erschafft ein erhebliches Rationalisierungspotenzial. Mit Hinblick auf den verstärkten Arbeitskräftemangel im Pflegebereich werden technische Lösungen immer attraktiver. ..Es sei erforderlich, das Pflegepersonal bei der Digitalisierung in den Krankenhäusern und Pflegeeinrichtungen entsprechend einzubeziehen, als auch um befürchteten Effekte der Deprofessionalisierung entgegen zu wirken.[44]

Im Jahre 2010 entwickelte die Europäische Kommission die Europa 2020-Strategie als Antwort auf die Finanzkrise aus dem Jahr 2008. Dadurch sollten die Schwächen der europäischen Wirtschaft aufgezeichnet werden. Zur Zielerreichung hat die Europäische Union sieben Leitinitiativen formuliert, eine davon die Digitale Agenda für Europa. Quelle Diese sieht unter anderem vor, die Informations- und Kommunikationstechnologie abgekürzt IKT zu fördern. Dadurch soll das gesellschaftliche Interesse am Online-Gesundheitsfürsorge und an den Telemedizinische System geweckt werden .[48] Dabei soll die Integration elektrischer Hilfsmittel für eine besser Gesundheitsversorgung in Europa beitragen [49] Ferner sorgen telemedizinische Systeme und Dienste mit Hilfe der IKT für die Übertragung von medizinischen Informationen zwischen Ärzten bzw. Ärzten und Patienten. Somit werden solche Systeme und Dienste bei der Diagnose, Behandlung und Weiterbetreuung der Patienten eingesetzt, was zu einer Verbesserung der Behandlungsqualität und Effizienz führen soll. [50] Auch auf nationaler Ebene wurden einige Maßnahmen zwecks der Digitalisierung des Gesundheitswesens umgesetzt. So wurde in Deutschland ebenfalls im Jahre 2010 die Electronic-Health Initiative (E-Health-Initiative) vom Bundesministerium für Gesundheit(BMG) ins Leben gerufen[51]

Patient im Wandel Die zunehmende Digitalisierung des Gesundheitswesens beeinflusst auch die Beziehung zwischen Arzt und Patient. Dies geht aus einer Umfrage des Digitalverbands Bitkom in Zusammenarbeit mit der Bayerischen Telemed Allianz aus dem Jahre 2017 hervor. Demnach informieren sich 88Prozent der Befragten zu Gesundheitsthemen, wovon mit 55Prozent mehr als die Hälfte das Internet nutzen., siehe folgende Abbildungen 13,14,15.

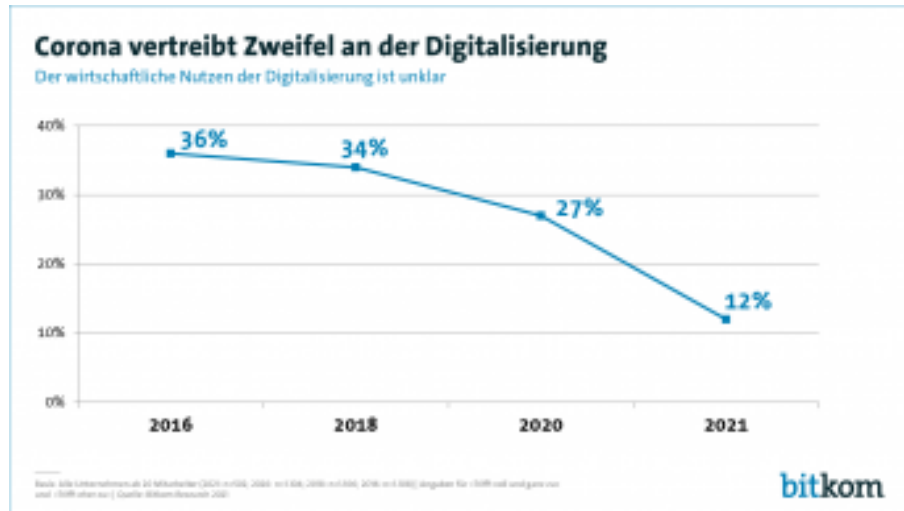


Figure 13: Darstellung des Zweifels der Menschen an der Digitalisierung in der Zeit von der Pandemie [61].

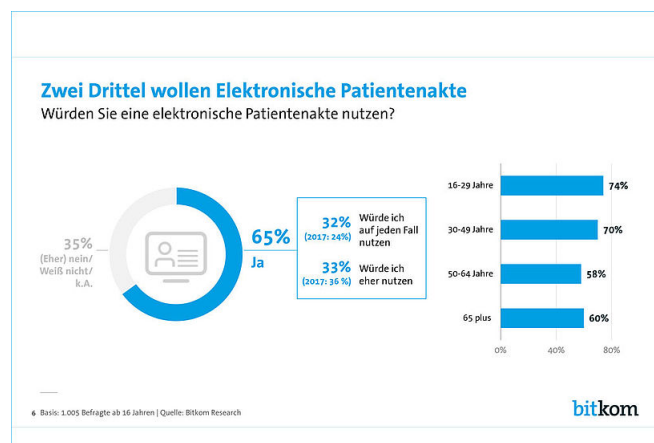
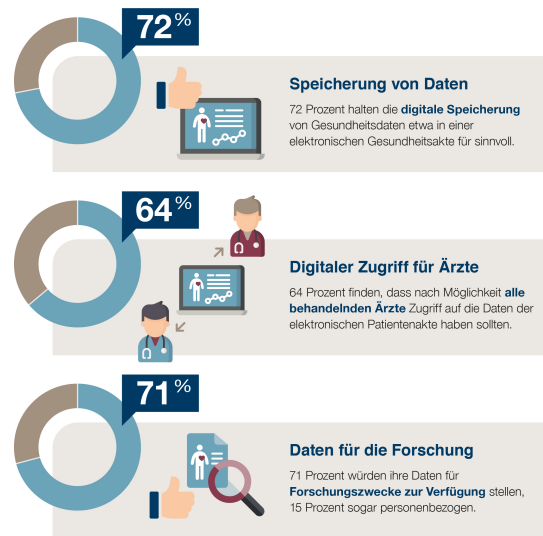


Figure 14: Darstellung der Bevölkerungsgruppen und ihrer Meinung [62].

Elektronische Patientenakte

Große Zustimmung in der Bevölkerung



Quelle: BfArG-Gesundheitsmonitor
Durchführung: The Nielsen Company, März/April 2020, n = 1.000

Bundesinstitut
für Arzneimittel
und Medizinprodukte
BfArG
Gesundheitsmonitor

Figure 15: Darstellung mehrerer Meinungen zu verschiedenen Aspekten der Elektronischen Patientenakte [63]

2.8 TEDIAS

Ziel des Projektes (Test- und Entwicklungszentrum für Digitale Anamnesesysteme)TEDIAS ist die Digitalisierung der Anamnese (medizinische Bestandsaufnahme des Patienten) als primärer und grundlegender Schritt, um darauf basierend ein digitales Krankenhaus schrittweise aufzubauen und damit eine Grundlage für das Digitale Krankenhaus der Zukunft zu erschaffen. Dazu wird ein Testzentrum für die Patientenaufnahme und die digitale Anamnese etabliert und ein mobiler „safe check-in“ entwickelt.

Um von der konventionellen, manuellen Anamnese auf automatische Systeme umzustellen, sind neue technische Konzepte erforderlich. Der Wechsel zu einem vollautomatischen System besteht aus mehreren Hauptkomponenten. Diese sind die Benutzerinteraktion mit dem Patienten und einer Messvorrichtung zur berührungslosen Erfassung physiologischer Daten. Die erfassten Daten müssen auf einer Plattform bereitgestellt werden, die für die nachfolgende Versorgung sowohl strukturiert als auch standardisiert ist. Vorhandene technische Lösungen können dabei wegen des hohen Risikos bei Fehlern nicht ohne klinische Prüfung übernommen werden. Das System muss einen validierten Prozess durchlaufen. Zusätzlich soll eine Informationsplattform geschaffen werden. Diese soll ein vollständig digitales Dossier ermöglichen, welches vom ärztlichen Personal entlang der medizinischen Prozesskette genutzt werden kann. Hierfür relevante Datenschutzaspekte sind in TEDIAS einbezogen. Die Planung von TEDIAS sieht vor, dass weitere klinische Module bis zu einem vollständigen Digitalen Krankenhaus in Nachfolgeprojekten ergänzt werden können.

„Gerade die Corona-Pandemie zeigt, wie wichtig ein leistungsfähiges und belastbares Gesundheitssystem ist. Baden-Württemberg ist hier bereits gut aufgestellt – trotzdem müssen wir uns ständig weiterentwickeln. Die Bereiche Personalisierte Medizin und Digitalisierung bietet hierbei vielfältige Möglichkeiten. Im Rahmen des Forums Gesundheitsstandort Baden-Württemberg fördern wir daher innovative Projekte, die uns nicht nur beim Kampf gegen die Pandemie unterstützen, sondern auch maßgeblich zur Stärkung der Zukunftsfähigkeit des Gesundheitsstandorts beitragen“, sagte Wirtschaftsministerin Dr. Nicole Hoffmeister-Kraut.^[1] Die Auswertung der Daten dauert nur Sekunden. Die Ergebnisse der digitalen Erstuntersuchung können dem Arzt, der den Patienten kurz darauf empfängt, direkt auf den Monitor gespielt werden. Dafür ist eine Anbindung des Systems an die gängigen Krankenhausssysteme notwendig. Die Zeit, die bisher für die routinemäßige Eingangsuntersuchung benötigt wurde, steht nun für ein längeres Arzt-Patienten-Gespräch zur Verfügung. Falls die Ergebnisse der Untersuchung auf eine relevante Infektion hindeuten, kann das Personal auch Schutzmaßnahmen ergreifen, bevor es zu Infektionen durch den Patienten oder die Patientin im Behandlungszimmer kommt.^[72]

Mit TEDIAS sollen jetzt erstmals die Routineprozesse, die bei der medizinischen Aufnahme in eine Klinik durchlaufen werden, automatisiert und damit perspektivisch ohne medizinisches Personal durchgeführt werden. Dadurch, dass der Patient dabei stärker aktiv in den eigentlichen Aufnahmeprozess involviert wird, kann er auch besser informiert werden. Im Zentrum des neuen Systems wird eine ausgetüftelte Sensorik mit einem Avatar für die Befragung des Patienten stehen. Geplant sind unter anderem integrierte

IR-Kameras zum Messen der Körpertemperatur, Sensoren zur Bestimmung der Herz- und Atemfrequenz, Mikrophone zum Aufzeichnen der Atemgeräusche oder der Stimme. [15] Bei Verknüpfung der elektronischen Patientenakte mit entscheidungsunterstützenden Systemen können patientenindividuelle Informationen der Akte beispielsweise mit evidenzbasiertem Wissen kombiniert werden, was wiederum zu einer leitliniengerechteren Behandlung der Patienten führen kann [44]. In der folgenden Abbildung 16 wird gezeigt, wie die europäischen Länder im Vergleich stehen.

Platzierung	Ranking 2016	Ranking 2018
1	Dänemark, Schweden	Dänemark
2	Estland, Finnland, Slowakei	Finnland, Schweden
3	Portugal	Estland, Spanien
4	Spanien	Schweiz
5	Österreich	Slowakei, Vereinigtes Königreich
6	Schweiz	Portugal
7	Belgien	Frankreich
8	Deutschland, Litauen, Niederlande	Niederlande, Österreich
9	Vereinigtes Königreich	Belgien, Deutschland, Litauen, Polen
10	Italien	Tschechische Republik
11	Frankreich, Slowenien	Italien, Slowenien
12	Polen	Irland
13	Tschechische Republik	
14	Irland	

Krankenhaus-Report 2019

Figure 16: Darstellung von 20 Europäischen Ländern zum Stand der Implementierung der Elektronischen Patientenkarte auf nationaler Ebene [44]

3 Material und Methoden

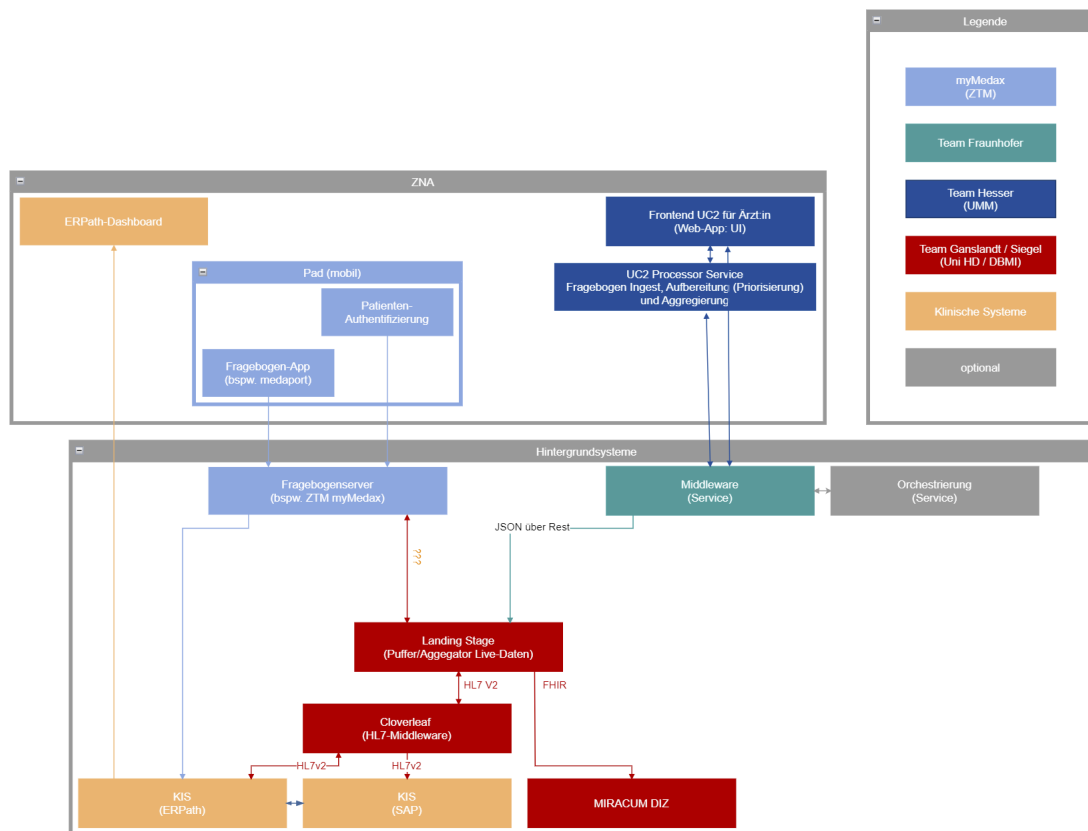


Figure 17: Das Use-Case 2 Dashboard wird über die .NET Core Middleware in die restliche TEDIAS-Landschaft integriert.

Das Architektur-Diagramm in Abbildung 17 bildeten den TEDIAS Workflow für den neurologischen Use-Case ab. Patienten der Neurologischen Notaufnahme bekommen an der Rezeption das MyMedax Tablett mit den Fragen ausgehändigt. Sie füllen den Anamnese Fragebogen selbstständig aus. Nach ausfüllen des Fragebogens werden die Daten auf einem eigens dafür eingerichteten Server geschickt.

Die Landing Stage erhält die Daten von dem Fragebogen Server per FHIR Schnittstelle und stellt die Fragebögen im JSON Format bereit. Der Orchestrator triggered den Datentransfer via der .NET Core Middleware, welche die Fragebögen als JSON Objekt an den UC2 (Use-Case 2) Backend Service schickt. Alle internen Schnittstellen basieren auf REST und sind per API Calls zugänglich.

Das UC2 Backend übernimmt die Data Processing Aufgaben und errechnet mittels einer Logik implementiert in Python zuerst die Question Priority und dann die darauf

folgende Patient Priority Score. Diese wird dann im UC2 Frontend angezeigt, welches auf einem Node.js/Vue.js Tech-Stack basiert. Das UC2 Backend basiert auf einem Flask Server.

Die Daten für das anzeigen im UC2 Frontend werden temporär im Filesystem des UC2 Backend hinterlegt. Die Datenbank (hier nicht eingezeichnet) welches in dieser Bachelorarbeit mit Data Ingest Pipeline implementiert wurde stellt eine längerfristige Datenspeicherung da. Die einzelnen Services sind soweit möglich individuell containerisiert um einen Microservices Ansatz zu folgen. Microservices bieten den Vorteil dass einzelne Services entwickelt, getestet und falls notwendig skaliert oder gelöscht werden können.

Die Anwendung besteht aus einem Back-End und einem Front-End-Service (siehe Architekturdiagramm). Beide Dienste werden als Teil einer Microservices-Architektur containerisiert und in einem Docker-Container ausgeführt. Das Docker-Basisimage ist ein Linux-Betriebssystem, vorzugsweise Ubuntu. Der Back-End-Dienst ist bei der registriert. NET Core Middleware, über die die gesamte REST-Kommunikation abgewickelt wird. Das Backend ist Python-basiert, während das Front-End auf einem Node.js/Vue.js Web Tech-Stack basiert.

SQL Code

```
1 CREATE TABLE IF NOT EXISTS Symptom
2 (
3     symptom_id          INT          NOT NULL AUTO_INCREMENT,
4     symptom              VARCHAR(50)  NOT NULL,
5     symptom_description  VARCHAR(100) NOT NULL,
6     PRIMARY KEY (symptom_id)
7 );
8
9 CREATE TABLE IF NOT EXISTS Patient
10 (
11     patient_id INT          NOT NULL,
12     first_name VARCHAR(100) NOT NULL,
13     last_name  VARCHAR(100) NOT NULL,
14     address    VARCHAR(100) NOT NULL,
15     gender     VARCHAR(100) NOT NULL,
16     birth_date DATE         NOT NULL,
17     PRIMARY KEY (patient_id)
18 );
19
20 # CREATE TABLE IF NOT EXISTS Patient_Symptom
21 # (
22 #     date          DATE NOT NULL,
23 #     patient_id INT  NOT NULL,
24 #     symptom_id INT  NOT NULL,
25 #     PRIMARY KEY (date, patient_id),
26 #     FOREIGN KEY (patient_id) REFERENCES Patient (patient_id),
27 #     FOREIGN KEY (symptom_id) REFERENCES Symptom (symptom_id)
28 # );
29
30 CREATE TABLE IF NOT EXISTS Question
31 (
32     question_id  INT          NOT NULL AUTO_INCREMENT,
33     question_text VARCHAR(255) NOT NULL,
34     PRIMARY KEY (question_id)
35 );
36
37 CREATE TABLE IF NOT EXISTS Symptom_Question
38 (
39     symptom_id INT NOT NULL,
40     question_id INT NOT NULL,
41     PRIMARY KEY (symptom_id, question_id),
42     FOREIGN KEY (symptom_id) REFERENCES Symptom (symptom_id),
```

```
43     FOREIGN KEY (question_id) REFERENCES Question (question_id)
44 );
45
46 CREATE TABLE IF NOT EXISTS Answer
47 (
48     answer_id    INT            NOT NULL AUTO_INCREMENT,
49     answer_text  VARCHAR(255) NOT NULL,
50     PRIMARY KEY (answer_id)
51 );
52
53 CREATE TABLE IF NOT EXISTS Question_Answer
54 (
55     question_id INT NOT NULL,
56     answer_id   INT NOT NULL,
57     PRIMARY KEY (question_id, answer_id),
58     FOREIGN KEY (question_id) REFERENCES Question (question_id),
59     FOREIGN KEY (answer_id) REFERENCES Answer (answer_id)
60 );
61
62 CREATE TABLE IF NOT EXISTS Questionnaire
63 (
64     questionnaire_id VARCHAR(50),
65     patient_id        INT NOT NULL,
66     PRIMARY KEY (questionnaire_id),
67     FOREIGN KEY (patient_id) REFERENCES Patient (patient_id)
68 );
69
70 CREATE TABLE IF NOT EXISTS Questionnaire_Response
71 (
72     questionnaire_id VARCHAR(50) NOT NULL,
73     question_id       INT         NOT NULL,
74     answer_id         INT         NOT NULL,
75     PRIMARY KEY (questionnaire_id, question_id, answer_id),
76     FOREIGN KEY (questionnaire_id) REFERENCES Questionnaire
77     ↪ (questionnaire_id),
78     FOREIGN KEY (question_id) REFERENCES Question (question_id),
79     FOREIGN KEY (answer_id) REFERENCES Answer (answer_id)
80 );
81
82 CREATE TABLE IF NOT EXISTS Waiting_Queue
83 (
84     patient_id INT            NOT NULL AUTO_INCREMENT,
85     start_time DATETIME NOT NULL,
86     end_TIME   DATETIME NULL,
```

```
86     PRIMARY KEY (patient_id, start_time),  
87     FOREIGN KEY (patient_id) REFERENCES Patient (patient_id)  
88 );
```


3.1 Datenbank

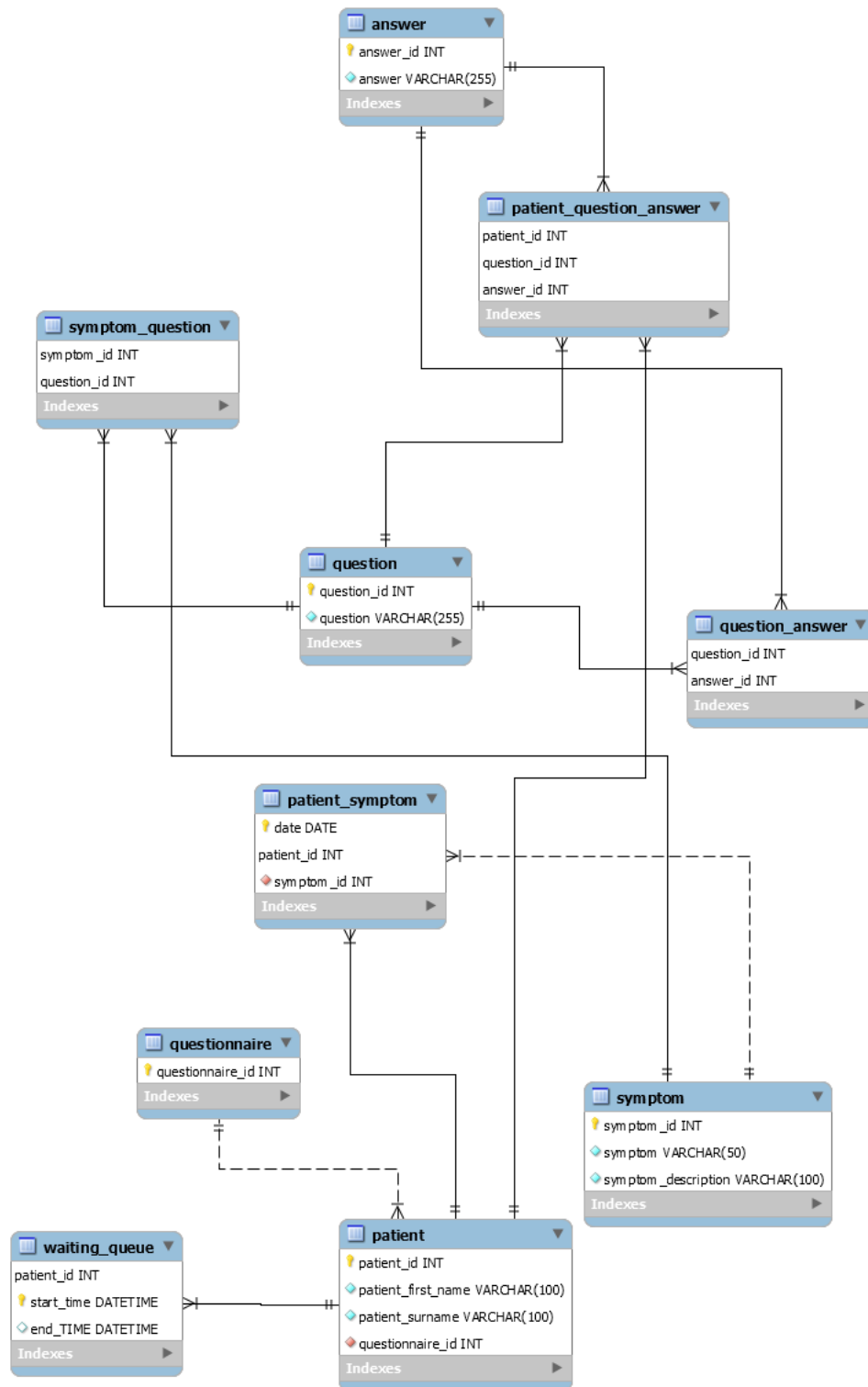


Figure 18: Datenbank Schema

Beschreibung der -funktionsweise des Datenbank Schemas:

In Abbildung 18 ist die Symptom Tabelle mit der Liste aller möglichen Symptomen die enthalten sind dazu war eine character length von 250 festgestellt mit dem gewissen das es in zukünftigen Projekten zu einer eventuellen Erweiterung kommen kann. Jedes Symptom hat neben seinem Namen und eine Description(Beschreibung) seinen ID als primärer Schlüssel der in den anderen Tabellen als ein Fremder Schlüssel benutzt werden soll damit das Problem von der Daten Wiederholung nicht auftreten kann die sogenannte Datenredondanz der Daten in der Tabelle.

Die Patienten Tabelle enthält alle Daten die den Patienten naher beschreibt wie der Vorname, Nachname, das Geschlecht , die Adresse und andere.

Die Question Tabelle enthält alle möglichen von der Ärztin gegebenen Fragen die sich nicht wiederholen.

Die Answer Tabelle enthält alle Unikaten antworten zu den gegebenen Fragen .

Question_Answer ist eine junction (Verbindungstabelle) zwischen der Question und der Answer Tabelle und hier wird eine MANY – TO – MANY Verbindung genutzt (relationship) zwischen diesen Zwei Tabellen. Da eine Frage mehrere Antwortmöglichkeiten beinhaltet und da eine Antwort sich in mehreren Fragen befinden kann.

Symptom_Question Tabelle modelliert das Verhältnis zwischen dem Symptom und der Frage und sie gibt uns Auskunft welche Frage sich in dem Fragebogen befinden muss wenn der Patient ein Hauptsymptom auswählt hier wird auch eine MANY – TO – MANY Verbindung genutzt.

Questionnaire Tabelle enthält den Questionnaire_ID als primärer Schlüssel und den Patienten ID als fremder Schlüssel. Diese Tabelle gibt uns Auskunft welcher Patient diesen Questionnaire(Fragebogen) ausgefüllt hat.

Questionnaire_response Tabelle verbindet eine Instanz von dem Fragebogen (welche bereits mit dem Patienten in Verbindung steht) mit allen Fragen und Antworten in diesem Fragebogen. Hier wird ein Compound Key verwendet mit den Elementen(Questionnaire_ID, Question_ID, Answer_ID) damit bereitgestellt wird das eine Frage mehrere Antwortmöglichkeiten(multiple choice) hat.

Waiting_Queue Tabelle repräsentiert die Warteschlange also welche Patienten warten. Hier wird auch ein Compound Key benutzt mit den Elementen (patient_ID, Start time) und damit wird sichergestellt dass ein Patient mehrmals in der Waiting_Queue Tabelle seinen kann mit natürlich anderen starting times damit wird die reuseability die Wiederverwendung der Software signifikant gesteigert.

Docker YAML

```
1 version: '3.1'
2
3 services:
4   db:
5     image: mysql
6     command: --default-authentication-plugin=mysql_native_password
7     restart: always
8     environment:
9       MYSQL_ROOT_PASSWORD: test
10      MYSQL_DATABASE: db
11     ports:
12       - 3306:3306
13
14   adminer:
15     image: adminer
16     restart: always
17     ports:
18       - 8080:8080
```

Install requirements

```
1 pip install -r requirements.txt
```

Run containers

```
1 docker-compose up
2 docker exec -it mario\_db\_1 bash
3 mysql db --password
4
5 Stop container and remove data
6
7 docker-compose down -v
```

docker-compose.yaml definiert 2 container

- Mysql Service der den default mysql image (ohne Veränderung benutzt) und der repräsentiert unsere Base in dem Projekt.
- Adminer Service der den default adminer image benutzt und der wird als admin interface benutzt für die Untersuchung der Daten.
- docker-compose up kommand ist dazu verantwortlich das alle Container gestartet werden.
- docker-compose down -v ist dazu verantwortlich das alle Container gelöscht werden und damit auch alle Daten die da gespeichert sind.

3.2 Data Ingest

`get_query_result` hat die Funktion sich an die Datenbank zu verbinden und die SQL Statements auszuführen und die gegebenen Ergebnisse an die Query zurückzuführen.

`execute_query` hat die Funktion die Data Definition Language(DDL) Statements auszuführen die sie bekommen hat. Die Funktion hat keinen return oder den Python NONE return.

`get_questions` hat die Funktion die Question in der JSON zu lesen und die auf den Python Dictionary zu übergeben(to parse).

`create_tables` hat die Funktion alle SQL Query auszuführen welche sich in der Query befinden und die DDL Statements. Sie liest die Files und ihren Content und macht einen Teilung(Split) bei dem Punkt und Komma und führt die einen nach den anderen durch die Query.

`insert_question_answer_data` hat die Funktion die Daten die sie bekommen hat aus den JSON files die zu transformieren und in die entsprechenden Tabellen einzufügen. Es handelt sich um die Tabellen Symptom, Question, Answer, Question_Answer und Symptom_Question mit den entsprechenden Daten die in den Stakeholdern gefüllt werden hier in VALUES gezeigt. Wie zu sehen ist geht in die Symptom Tabelle nur das Symptom, in die Tabelle Answer geht der Wert Answer und in die Tabelle Question_Answer wird die entsprechende Frage mit der Antwort verbunden die Tabelle Symptom_Question bekommt die Symptom Id und die Question als Values.

`get_questionnaire_responses` hat die Funktion mehrere JSON Files zu öffnen und gibt eine Reihe von Objekten zurück. Im JSON string werden die in Python Objekte umgewandelt und kehren als Liste zurück jeder von responses die es zurück gibt ist eine Python Objekt.

`insert_patient_data` hat die Funktion die Tabellen entsprechend auszuführen basierend auf den Input das es bekommt.

Die main hat die Funktion alles in korrekter Reihe auszuführen wie hier zu sehen werden die Tabellen zu erst erstellt.

Python Code

```
1 import json
2 import pymysql.cursors
3
```

```
4
5 def get_query_result(sql, data=None):
6     with pymysql.connect(host='localhost', user='root', password='test',
7         ↪ database='db') as conn:
8         with conn.cursor() as cursor:
9             cursor.execute(sql, data)
10            return cursor.fetchall()
11
12 def execute_query(sql, data=None):
13     with pymysql.connect(host='localhost', user='root', password='test',
14         ↪ database='db') as conn:
15         with conn.cursor() as cursor:
16             cursor.execute(sql, data)
17             conn.commit()
18
19 def get_questions():
20     with open('../json/questions.json', 'r') as f:
21         questions_json = json.load(f)
22         questions = questions_json['questions']
23
24     return questions
25
26
27 def create_tables():
28     with open('../sql/ddl.sql', 'r') as f:
29         sql = f.read()
30         for stmt in sql.split(';'):
31             if stmt.strip():
32                 execute_query(stmt)
33
34
35 def insert_question_answer_data(questions):
36     inserted_answers = set()
37     inserted_symptoms = set()
38     for question in questions:
39         sql = "INSERT IGNORE INTO `Question` (`question_id`,
40             ↪ `question_text`) VALUES (%s, %s)"
41         execute_query(sql, (question['id'], question['text']))
42
43         symptom = question['symptom']
44         if question['symptom'] not in inserted_symptoms:
45             inserted_symptoms.add(symptom)
```

```
45         sql = "INSERT IGNORE INTO `Symptom` (`symptom`) VALUES (%s)"
46         execute_query(sql, (symptom,))
47
48     for answer in question['answers']:
49         if answer not in inserted_answers:
50             inserted_answers.add(answer)
51             sql = "INSERT IGNORE INTO `Answer` (`answer_text`)
52                 ↳ VALUES (%s)"
53             execute_query(sql, (answer,))
54
55         answer_id = get_query_result(f"SELECT * FROM Answer WHERE
56                 ↳ answer_text = '{answer}'")[0][0]
57         sql = "INSERT IGNORE INTO `Question_Answer` (`question_id`,
58                 ↳ `answer_id`) VALUES (%s, %s)"
59         execute_query(sql, (question['id'], answer_id))
60
61     if symptom:
62         symptom_id = get_query_result(f"SELECT * FROM Symptom WHERE
63                 ↳ symptom = '{symptom}'")[0][0]
64         sql = "INSERT IGNORE INTO `Symptom_Question` (`symptom_id`,
65                 ↳ `question_id`) VALUES (%s, %s)"
66         execute_query(sql, (symptom_id, question['id']))
67
68 def get_questionnaire_responses():
69     responses = []
70
71     with open('../json/response_1.json', 'r') as f:
72         responses.append(json.load(f))
73
74     with open('../json/response_2.json', 'r') as f:
75         responses.append(json.load(f))
76
77     return responses
78
79 def insert_patient_data(questionnaire_responses):
80     for response in questionnaire_responses:
81         questionnaire_id = None
82         if response['identifier']['type']['text'] == 'Questionnaire Id':
83             questionnaire_id = response['identifier']['value']
84
85         for patient in response.get('contained', []):
86             patient_id = None
```

```
84         for identifier in patient['identifier']:
85             if identifier['type']['text'] == 'Patient Number':
86
87                 patient_id = identifier['value']
88
89                 sql = "INSERT IGNORE INTO `Patient` (`patient_id`,
90                     ↪ `first_name`, `last_name`, `address`, `gender`,
91                     ↪ `birth_date`) VALUES (%s, %s, %s, %s, %s, %s)"
92                 execute_query(sql, (patient_id, patient['name'][0]['given'],
93                     ↪ patient['name'][0]['family'],
94                     ↪ patient['address'][0]['text'],
95                     ↪ patient['gender'],
96                     ↪ patient['birthDate']))
97
98                 sql = "INSERT IGNORE INTO `Questionnaire`
99                     ↪ (`questionnaire_id`, `patient_id`) VALUES (%s, %s)"
100                 execute_query(sql, (questionnaire_id, patient_id))
101
102         for item in response.get('item', []):
103             question_text = item['text']
104             question_id = get_query_result(f"SELECT * FROM Question
105                 ↪ WHERE question_text = '{question_text}'")[0][0]
106
107             for answer in item['answer']:
108                 answer_text = answer['valueString']
109                 answer_id = get_query_result(f"SELECT * FROM Answer
110                     ↪ WHERE answer_text = '{answer_text}'")[0][0]
111                 sql = "INSERT IGNORE INTO `Questionnaire_Response`
112                     ↪ (`questionnaire_id`, `question_id`, `answer_id`)
113                     ↪ VALUES (%s, %s, %s)"
114                 execute_query(sql, (questionnaire_id, question_id,
115                     ↪ answer_id))
116
117 def main():
118     create_tables()
119     questions = get_questions()
120     insert_question_answer_data(questions)
121
122     questionnaire_responses = get_questionnaire_responses()
123     insert_patient_data(questionnaire_responses)
124
125 if __name__ == '__main__':
```

117 `main()`

data/json/folder enthält json Files die sich in das MySQL datenbank importieren Questions.json enthält alle möglichen Fragen und Antworten Response_1.json und response_2.json wurden als Beispiel ausgefüllte Fragebogen benutzt

sql/ddl.sql enthält alle Query für die Date definition Language (DDL) hier entspricht es den CREATE TABLE statements welche das ganze Schema erstellen.

src/main.py ist die hauptpython script die alles macht :

- zu erst werden alle Tabellen erstellt (durch „create_tables()“)
- Danach wird der questions.json prozessiert und aus dem werden die Question, Answer, Question_Answer, Symptom, Symptom.Question Tabellen ausgefüllt.
- Danach wird der zweite json type prozessiert aus dem die Tabellen Patient, Questionnaire und Questionnaire_Response ausgefüllt wird.
- Jedes mal wo diese Python Skripte „gerunt“(ausgeführt wird) wird die keine Duplikate für die Fragen und Antworten machen da es eine „INSERT IGNORE INTO“ genutzt die es ignoriert falls die Daten vorhanden sind mit diesen primären schlüsseln in der Tabelle.
- Für die Testung der Duplikate und die Verhinderung des „INSERTS“ (Einfügung) für die Fragen und Antworten die mit dem Patienten in Verbindung gebraucht werden, benutzt die Skript inserted_answers und inserted_questions in den Python Code wird getestet ob wir den Patienten mit den Fragen schon „INSERTED „ haben mit den gleichen Fragen und Antworten

3.3 Unit Tests

```

1  def get_query_result(sql, data=None):
2      with pymysql.connect(host='localhost', user='root', password='test',
        ↳ database='db') as conn:
3          with conn.cursor() as cursor:
4              cursor.execute(sql, data)
5              return cursor.fetchall()
6
7
8  def test_empty_questions_array():
9      """Tests behavior of question parser function when input is empty"""
10     questions = main.get_questions('../json/questions_empty.json')
11     assert len(questions) == 0
12
13
14  def test_all_tables_created():

```

```
15     """Tests if all tables from ddl.sql file are created successfully"""
16     main.create_tables()
17     table_count = get_query_result('SHOW TABLES IN db')[0][0]
18     assert table_count == 10
19
20
21 def test_question_answer_data_inserted():
22     """Tests if question and answer data are inserted into the DB"""
23     main.create_tables()
24     questions = main.get_questions('../json/questions_empty.json')
25     main.insert_question_answer_data(questions)
26
27     question_count = get_query_result('SELECT COUNT(*) FROM
    ↪ Question')[0][0]
28     assert question_count > 0
29
30     answer_count = get_query_result('SELECT COUNT(*) FROM Answer')[0][0]
31     assert answer_count > 0
```

4 Resultate

4.1 Anforderungs/Risikoanalyse

Das System ist auf ein Hauptsymptom pro Patient und Besuch beschränkt, was seine Diagnosefähigkeiten stark reduziert, was wiederum zu Nadelstress führen kann.

In der Schwangerschaft würden die Symptome eine falsche Rangplatzierung erfordern und somit zu Ineffizienz führen. In dem Entscheidungsbaum, der im Algorithmus implementiert ist, wäre diese Frau im Nachteil. Im Falle einer Schwangerschaft der Patientin kann es aufgrund unangemessener Fragenbeantwortung zu einer Fehlklassifizierung kommen.

Menschen mit psychischen Problemen könnten unter mehr Stress stehen, wenn sie sich darauf einigen müssen, gefilmt zu werden und Teil einer Studie zu sein. Das System wäre besser, diese beiden Anforderungen zu trennen.

Es gibt keinen Alarm, dass das Gerät das Gebäude verlassen hat, wenn ein Patient absichtlich oder unabsichtlich das Gerät mitnimmt. Vor allem ältere Patienten neigen dazu, zu vergessen, es zurück zum medizinischen Personal zu bringen.

Es gibt keine rote Flagge für Patienten, die "cramps / Krämpfe" als Hauptsymptom beantwortet haben, Krämpfe könnten ein wichtiger Indikator dafür sein, dass es einen ernsteren Grund dafür gibt oder in der Schwangerschaft, wo es möglich ist, mehr Symptome zu haben, die sich überschneiden. Patienten, die an einer bestimmten chronischen Krankheit wie Multipler Sklerose leiden, würden es schwierig finden, sich für ein Hauptsymptom zu entscheiden, und das Fehlen eines Feldes "Patientengeschichte" könnte diese Person falsch einstufen. Das Fehlen einer roten Flagge auf dem Gebiet der Frage "Sind Sie sich einer neurologischen Erkrankung bewusst?" und die verfügbaren Antworten.

Wenn das Hauptsymptom "Schwäche / Lähmung / Unbehagen?" in der folgenden Frage "Wo sind Ihre Beschwerden?" ist, sollte die mögliche Antwort in zwei mögliche Antworten anstelle einer "Kopf / Gesicht - einseitig / doppelseitig" aufgeteilt werden.

Der Benutzer lässt das Dashboard geöffnet und belässt Patienteninformationen auf dem Bildschirm.

Wenn das Dashboard abstürzt und nicht neu gestartet werden kann, kann der Arzt für die Dauer nicht auf Patienteninformationen zugreifen oder Elemente in die Patientenakte eingeben.

Das Dashboard ist so konzipiert, dass Ärzte jederzeit aktualisierte Informationen zu Patienten sehen können. Wenn die Intranet- oder Back-End-Server ausgefallen sind, werden einige Informationen möglicherweise nicht aktualisiert, und ein Arzt könnte Entscheidungen im Gesundheitswesen auf der Grundlage veralteter Informationen treffen. Die höchste Priorität hat die Sicherstellung der Patienten Daten und dadurch gilt es viele Vorschriften zu befolgen in diesem Fall sind strenge Dokumentationsvorschriften, um sich rechtlich abzusichern und die Haftung zu reduzieren. Deshalb hat die Prävention von Datenschutz Verletzungen die höchste Priorität. Die Risiken die mit dem Datenschutz

verbunden sind können durch viele Ursachen ausprägt werden. Um die Risiken zu vermeiden können wir die folgenden Schritte Einfügen ... 1) Fügen Sie eine zeitgesteuerte Sperrung nach einem Satz Verspätung (d.h. 5 Minuten) von keiner Aktivität, nach die der Benutzer benötigt um ihre Passwort. 2) Trainieren Sie die Benutzer, um Vorsicht vor Datenschutz und ihre Computer, wenn sie es sind Wegtreten 3) Eine Mögliche Sperre von dem Gerät einbauen wenn das Gerät das WLAN von dem Krankenhaus verlässt Damit wird die Risiko Elimination auf die höchste Stufe gebracht durch die Maßnahmen wird die Eintrittswahrscheinlichkeit dieser Risiken möglichst verkleinert. 4) Stellen Sie sicher, dass die Dashboard-Programm ist das richtige Passwort zusätzlich geschützt zum Laptop selbst. Das Dashboard ist browserbasiert, also ein Hinzufügen eines Kennworts auf Zugang zum VPN des Krankenhauses sollte enthalten sein 5) Vermeiden Sie große Displays Patienteninformationen mit Identifikatoren auf der gleicher Bildschirm. Vermeiden Verwendung von Patientenbildern tunlichst.

Eine hohe Priorität wird auch der Internet und Server Verbindung Problemen gewidmet. Das Dashboard ist so konzipiert, dass Ärzte kann aktualisiert werden Informationen zu Patienten bei alle Zeiten. Wenn das Internet oder Backend-Server sind runter, einige Informationen kann nicht aktualisiert werden, und eine Arzt könnte machen Entscheidungen im Gesundheitswesen basierend auf veraltet Information. Um die Risiken zu vermeiden können wir die folgenden Schritte Einfügen um die normale Funktionalität des Krankenhauses zu gewährleisten Eine Warnung auf die Homepage, die zeigt, dass die das Dashboard ist derzeit offline und ein Datum/eine Uhrzeit, zu der es wurde zuletzt aktualisiert. So dass Krankenhaus personal möglichst leicht wieder den Überblick bekommt und die Patient bestmöglich versorgt sind.

Das Risiko der fehlerhaften Datenspeicherung gehört von der Priorität etwas niedriger da es sehr selten passiert und keine großen prozess- beendeten folgen hat. Das Dashboard ermöglicht die Manipulation und Hinzufügen von Informationen an eine Patientenakte. Das stellt ein Risiko dar, bei dem dass Daten verloren gehen können, wenn die Dashboard geschlossen ist oder stürzte ab und war nicht richtig gespeichert. Die Schritte die eingefügt werden können um dieser Risiko zu vermindern sind Das Dashboard sollte automatisch alle 30 Minuten speichern oder wann Informationen sind irgendwo eingegeben, und ein Backup der auf der Geschütztes Krankenhaus Server Falls aber eine fehlerhafte Eingabe in dem Programm existieren sollte, gibt es eine eine beliebige Möglichkeit, Kehren Sie die Option(UNDO) um oder löschen Sie die Fehler ohne den Patient aus der Queue zu löschen. Dies sind die maßnahmen die getroffen werden sollte um eine fehlerhafte Patienten Eingabe zu beheben.

4.1.1 User Stories

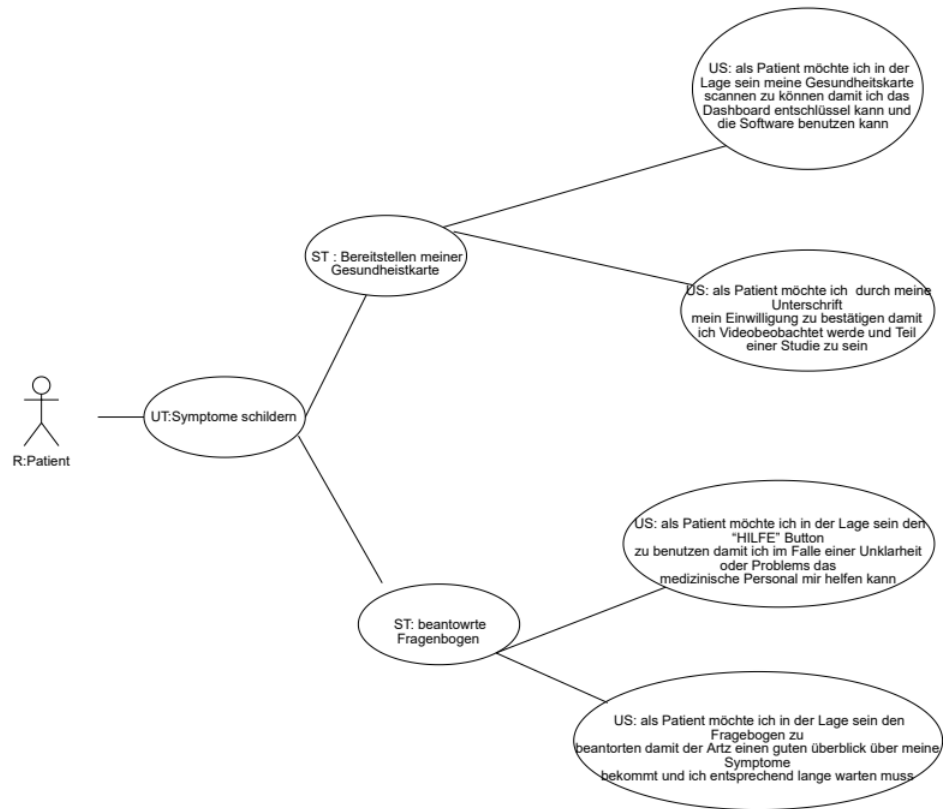


Figure 19: User Story Journey Patient

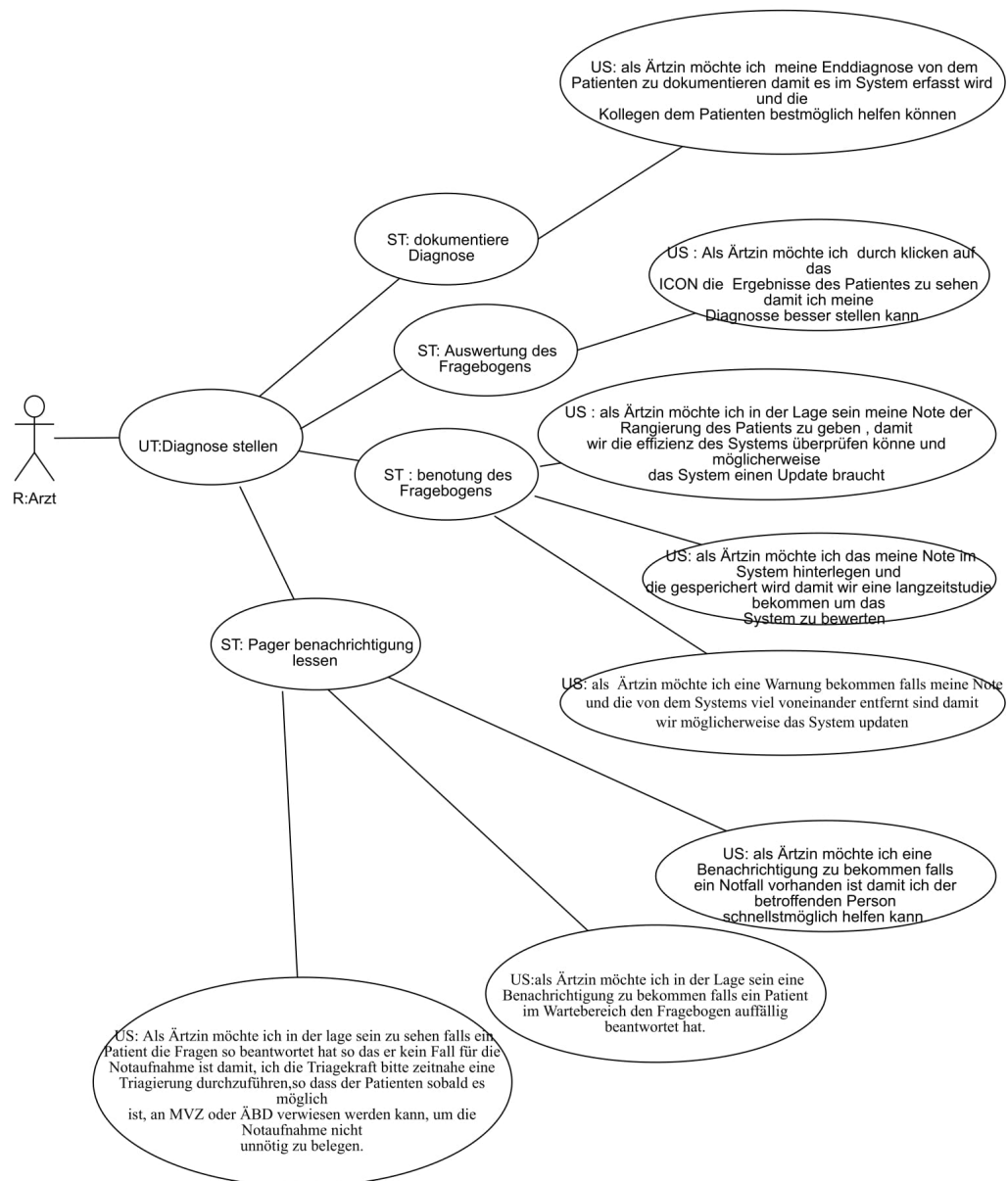


Figure 20: User Story Journey Arzt

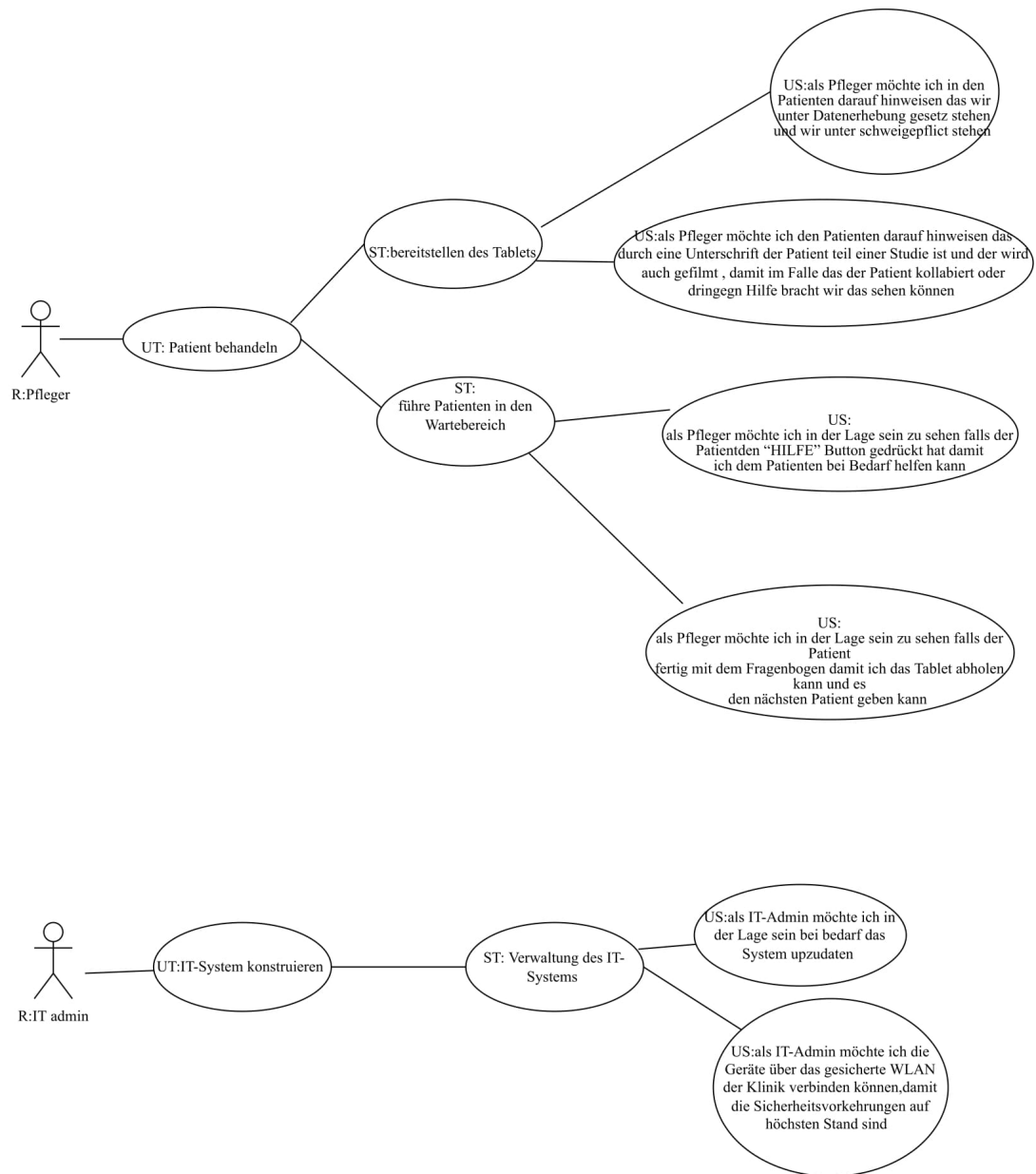


Figure 21: User Story Journey Pfleger und IT-Admin

4.1.2 Akzeptanzkriterien

Akzeptanzkriterien Story 1 : US: US: Als Patient möchte ich in der Lage sein, den Fragebogen so auszufüllen, dass die Ergebnisse in das System überführt werden. Akzeptanzkriterien :

- Alle Fragen im Fragebogen müssen beantwortet werden
- Die Eingabewerte müssen den benötigten Kriterien entsprechen

Aus dieser Story ergeben sich einige weitere wichtige Informationen : Der Patient muss alle Fragen mit gewünschten Werten ausfüllen damit die auch richtig bearbeitet werden , bei einer Überspringung einer Frage wird der Fragebogen nicht gespeichert.

Story 2 : US: Als Arzt möchte ich eine Benachrichtigung erhalten können, wenn ein Patient den Fragebogen gezielt beantwortet hat und seine Antwort eine mögliche schnelle Behandlung benötigt Akzeptanzkriterien :

- Alle Fragen im Fragebogen müssen beantwortet werden
- Erst nach dem der Patient den Fragebogen beenden hat können diese Werte der Ärztin angezeigt werden
- Die Benachrichtigungen an den Arzt wird verursacht durch Beantwortung der spezieller Fragen in einer vorgeschickten weißen die speziellen Fragen und die spezielle Beantwortung sind :
 - 6.Haben Sie zusätzliche Beschwerden seit dem Auftreten des Schwindels? → ja
 - 11.Sind die Schmerzen plötzlich oder innerhalb sehr kurzer Zeit (d. h. innerhalb einer Minute) aufgetreten? → ja
 - 15. Haben Sie Fieber? → ja
 - 17. Ging den Kopfschmerzen eine körperliche Anstrengung (inkl. Husten/Niesen/-Pressen/schweres Heben) voraus? → ja
 - 18 Haben Sie weitere Beschwerden zusätzlich zu den Kopfschmerzen? → ja
 - 19 Welche der folgenden Beschwerden haben Sie zusätzlich?
 1. 3.Schmerzen an einer Hals- oder Nackenseite
 2. 4. Sprachstörung (z. B. undeutliches „verwaschenes“ Sprechen)
 3. 5.einseitige Schwäche/Lähmung oder Missempfindung
 - 20. Wo sind Ihre Beschwerden?
 1. 1. einseitig/beidseitig
 2. 2. eine Körperhälfte (rechts/links)
 3. 3. eine Gesichtshälfte eine Körperhälfte
 4. 4. beide Beine

- 26 Haben oder hatten Sie zusätzlich eine oder mehrere der folgenden Beschwerden: → Einseitige Lähmung/Schwäche
- 29 Welche Aussage beschreibt die Beschwerden am besten? → Ich kann/konnte plötzlich auf einem Auge nichts mehr oder viel schlechter sehen. Und → Ich sehe doppelt bzw. habe doppelt gesehen (d. h. nebeneinander oder übereinander stehende Bilder).
- 30 Sind die Sehstörungen von anderen Symptomen begleitet: → Schwäche/Lähmung

Aus dieser Story ergeben sich einige weitere wichtige Informationen : Viele Fragen haben vor gespeicherte Auswirkung auf den Fragebogen sobald diese mit den kritischen Antworten ausgefüllt wird bezweckt das das der Patient einen „Red Flag“ bekommt. Es genügt nur eine Frage der gelisteten Fragen mit dem gewünschten Ergebnis auszufüllen sofort wird der Patient mit einem „Red Flag“ gekennzeichnet.

Story 3 US: Als Arzt möchte ich in der Lage sein, auf das ICON zu klicken und die Ergebnisse des Patienten zu sehen, damit ich meine Diagnose verbessern kann

Story US: Als Patient möchte ich in der Lage sein, eine Einverständniserklärung auszufüllen, um zuzustimmen, Teil einer Studie zu sein und respektvoll gefilmt zu werden. Akzeptanzkriterien :

- Erst nach dem der Patient den Fragebogen beenden werden dieser Werte an das System geschickt und dann auch in das geeignete fach für die medizinischen Studien

Aus dieser Story ergeben sich einige weitere wichtige Informationen : Es gibt Arbeitsgesetze die ebenfalls berücksichtigt werden müssen.

4.1.3 Systemanforderung

Dieser Abschnitt veranschaulicht das vollständige Design des Dashboards nach Seiten und Registerkarten in der Software. Der Inhalt jeder Seite oder Registerkarte wird durch funktionale Anforderungen beschrieben. Funktionale Anforderungen werden mit F-REQ bezeichnet. Nichtfunktionale Anforderungen sind mit NF-REQ gekennzeichnet. Das Nummerierungsschema ist fortlaufend, wenn funktionale und nicht funktionale Anforderungen für jeden Teil des Dashboards berücksichtigt werden. Beispielsweise kann die Homepage F-REQ-1.1 und NF-REQ-1.2 enthalten, aber nicht sowohl F-REQ-1.1 als auch NF-REQ-1.1. Die erste Ziffer im Nummerierungsschema ist immer 4, da alle Anforderungen hier in Abschnitt 4 aufgeführt sind. Die zweite Ziffer gibt an, auf welchen Teil der Software verwiesen wird. 4.1 ist die Homepage.





ID	Vorname	Nachname	Besuchgrund	Wartezeit		
e6f93635-cbd6-468b-9fb6-1fabde7b879d	Desire	Holland	Missempfindung (Kribbeln)/Taubheit der Haut	26		Nach Hause geschickt Weitergeleitet
e6f93635-cbd6-468b-9fb6-1fabde7b879a	Ermine	Schmidt	Schwindel	18		Nach Hause geschickt Weitergeleitet
Seit wann besteht der Schwindel? Länger als 6 Tage ohne kürzliche Verschlimmerung Leiden Sie zusätzlich unter Übelkeit/Erbrechen? Nein Ist der Schwindel dauerhaft vorhanden? Ja Tritt der Schwindel nur bei bestimmten Bewegungen auf (z. B. Umdrehen im Bett, Bücken, Kopfneigung nach hinten)? Ja Haben Sie zusätzliche Beschwerden seit dem Auftreten des Schwindels? Nein						
e6f93635-cbd6-468b-9fb6-1fabde7b879c	Paul	Müller	Krampfanfall	2		Nach Hause geschickt Weitergeleitet
e6f93635-cbd6-468b-9fb6-1fabde7b879b	Jialu	Wang	Kopfschmerzen	12		Nach Hause geschickt Weitergeleitet

Figure 22: TEDIAS Neurologische Notaufnahme Use-Case Patientendashboard. Implementierung auf JavaScript Basis bei Max Hirsch^[71].

Beschreibung und Priorität Aktualisierung der Tabelle mit hoher Priorität nach kurzer Zeit.

Stimulus-/Reaktionssequenzen Patienten werden dem Dashboard hinzugefügt, sobald sie den Fragebogen ausgefüllt haben. Nach Verlassen der Notaufnahme löscht das medizinische Personal mit einem Klick auf die Schaltfläche „Bearbeitet/Processed“ einen Patienten aus der Liste.

Funktionale Anforderungen

F REQ-1: Tabelle der wartenden Patienten

- NF REQ 1: Die Aktualisierung der Tabelle erfolgt alle 5 Sekunden oder wenn ein neuer Patient in das System importiert/hinzugefügt wurde
- NF REQ 1.2: Das System funktioniert ohne Popup-Fenster, ist also einfach zu bedienen und schränkt die Möglichkeit der Verwirrung ein
- F REQ 1.3: Die Karte zeigt die grundlegenden Patienteninformationen, anhand derer der Arzt den Patienten identifizieren kann
 - F REQ 1.3.1: Patienten-ID
 - * F REQ 1.3.2: Vorname
 - * F REQ 1.3.3: Nachname
 - * F REQ 1.3.4: Hauptsymptom
 - * F REQ 1.3.5: Wartezeit wie lange wartet der Patient
 - * F REQ 1.3.6: Alarmsystem (Patientenprioritätsstatus): Dies ist ein System auf der Homepage, das dem Benutzer alle „Alarmer“ anzeigt, die derzeit für alle Patienten auf der Station aktiv sind. Es fasst die wichtigen Aktualisierungen und Aufgaben zusammen, die der Benutzer in Bezug auf jeden Patienten anzeigen oder ausführen sollte. Hier werden alle Alarmer für alle Patienten auf der Station angezeigt.
 - F REQ 1.3.6.1: Diese sind in 3 Alarmgruppen unterteilt: rote, gelbe und grüne Alarmer

- * F REQ 1.3.7: Durch Klicken auf die Patientenzeile mit dem Patienten öffnet sich ein Feld mit den Fragen und Antworten aus dem Fragebogen, die maßgeblich für den Patientenprioritätsstatus verantwortlich waren.
 - NF REQ 1.3.7.1: Die resultierende Addition soll in einer Zeit nicht länger als 200ms angezeigt werden
 - F REQ 1.3.7.2: Das Klicken auf einen anderen Patienten in der Tabelle führt zum Schließen des Seitenmenüs des vorherigen Patienten
 - NF REQ 1.3.7.3: Durch Klicken auf eine leere Stelle wird der Alarm ausgeblendet, bis er sich geändert hat (z. B. wenn neue Patienten hinzugefügt wurden)
- * F REQ 1.4: Das Löschen eines Patienten sollte dazu führen, dass der Patient aus dem System verschwindet
 - NF REQ 1.4.1: Der Löschvorgang sollte nicht länger als 200ms dauern
 - F REQ 1.4.2: Patienten bleiben in der Patientenliste, bis sie von einem Arzt angesprochen oder entlassen werden.

Patienten-IDs werden innerhalb eines bestimmten Arbeitstages nicht wiederverwendet.

Andere nicht-funktionale Anforderungen

- **Leistungsanforderungen**

Nach dem Einschalten des Geräts und dem scannen der Gesundheitskarte von dem Mitarbeiter sollte der Fragebogen einsatzbereit sein. Nachdem ein Benutzer auf eine Schaltfläche geklickt hat, sollte innerhalb von 200 ms eine Antwort angezeigt werden, um ein Gefühl der Insuffizienz und Inkompetenz des medizinischen Personals zu vermeiden, ein Gefühl, das durch langsame Software entstehen kann. Während die von den Patienten gegebenen Informationen verarbeitet werden, sollte ein Ladebalken erscheinen, der dem Patienten signalisiert, dass das System an seinen Informationen arbeitet. Ladezeiten sollten nicht länger als 1 oder 2 Sekunden dauern. Bei Wiederholungspatienten würde das Vorladen von Daten, die bereits im System vorhanden sind, die Vorbereitungs- und Wartezeit des Patienten reduzieren. Updates, die während der Ausführung des Dashboards vorgenommen werden, sollten vollständig geladen werden, bevor sie dem Benutzer angezeigt werden. Die Quelle der Anfrage nach den Attributen stammt aus dem Interview und ist als Rückverfolgbarkeit vor der Anforderungsspezifikation definiert.

- **Sicherheitsanforderungen**

Es ist von größter Bedeutung, dass die Sicherheit des Dashboards gewährleistet ist, da es sich direkt in Patientensicherheit und Datenschutz niederschlägt. Da das Dashboard der einzige Zugangspunkt für einen Patienten ist, um Patienteninformationen zu geben, ist es wichtig, dass seine Stabilität geschützt wird, um mögliche Therapiefehler zu vermeiden, da ein Ausfall eines Teils des Dashboards oder der

dahinter liegenden Server zu Ärzten führen würde nicht in der Lage sind, Patienteninformationen nachzuschlagen und Patienten im Rangsystem falsch einzuordnen, würde dies dazu führen, dass ein Patient lange warten muss, was seine Situation verschlechtern könnte. Dazu gehören regelmäßige nächtliche Backups und Offline-Datenspeicherung, um auch im Fehlerfall verfügbar zu sein. Das Offline-Datenformat muss ein deutliches Wasserzeichen aufweisen, um sicherzustellen, dass der Arzt, der die Informationen liest, darauf aufmerksam gemacht wird, wann die Informationen vorliegen und dass sie veraltet sind. Der Datenschutz von Dashboards muss jederzeit Priorität haben. Das IT-Team muss gewissenhaft daran arbeiten, den Schutz der im Dashboard angezeigten Patientendaten sicherzustellen. Um dies zu gewährleisten, soll ein Sleep-Timer von 5 Minuten eingerichtet werden, der eine erneute Anmeldung des Patienten erfordern würde, um wieder darauf zuzugreifen, und der nur über das Krankenhausnetzwerk oder das Krankenhaus-VPN zugänglich sein sollte. Der gesamte Netzwerkverkehr sollte über das interne Netzwerk geleitet werden. Zu keinem Zeitpunkt sollte der Datenverkehr dem öffentlichen Internet ausgesetzt werden. Der Zugriff auf das Dashboard soll keinem Nutzer innerhalb des Universitätsklinikumsnetzes zur Verfügung stehen. Der Zugriff sollte über einen nicht öffentlichen Benutzernamen und ein Passwort beschränkt werden Dokumentation für rechtliche Zwecke. In bestimmten Projekten oder Branchen (z. B. in einem sicherheitskritischen Umfeld wie der Medizin, wie in diesem Beitrag) ist es notwendig, bestimmte Dokumentationsvorschriften einzuhalten.

- Softwarequalitätsattribute

Die Software muss sehr anpassungsfähig und flexibel sein. Anpassung an Umgebungsänderungen in Form von Informationen, die von dem/den Benutzer(n) bereitgestellt werden. Eines der Hauptmerkmale der Software ist ihre ständige Verfügbarkeit und Bereitschaft. Robust zu sein, wie es im Bereich der Informatik definiert ist, stellt die Fähigkeit des Systems dar, mit fehlerhaften Eingaben und irrtümlichen Eingaben fertig zu werden ein Krankenhaus. Usability definiert als eingesparte Zeit, die besser genutzt werden kann, z. B. für die Arbeit mit Patienten. Skalierbarkeit ist in diesem Dokument nicht enthalten. Hochverfügbarkeit ist in dieser Phase keine Voraussetzung.

- Geschäftsregeln (Business rules) Wird hauptsächlich für objektorientierte Softwaresysteme verwendet. Der Kunde erstellt ein Geschäftsmodell, das Artefakte aus der Geschäftswelt enthält. Dies bildet die Grundlage für die weiteren Anforderungen an das System. Der Auftragnehmer verwendet die Software Requirements Specification (SRS), um die Gesamtheit der Softwareanforderungen zu dokumentieren. Diese basieren auf dem ISO/IEC/IEEE-Standard. Dieses Produkt ist für die Verwendung in einer klinischen Umgebung vorgesehen. Die gesetzlichen Anforderungen werden dabei innerhalb der Bundesrepublik Deutschland durch das Medizinproduktegesetz sowie die EU-Datenschutz-Grundverordnung (DSGVO) geregelt. Die Einhaltung dieser Vorschriften ist unerlässlich. Weitere regulatorische An-

forderungen werden von der Ethikkommission der Universitätsklinikum Mannheim GmbH erlassen. Weitere aufsichtsrechtliche Anforderungen werden durch den Datenschutzbeauftragten der Universitätsklinikum Mannheim GmbH gestellt. Die ärztliche Schweigepflicht ist jederzeit zu wahren.

4.2 Technical Results

Risiken die eliminiert sind

Eine Frage kann in mehreren Symptomen vorkommen damit ist es notwendig eine „many-to-many“ Verbindung zu erstellen. Damit es nicht zur Daten redundanz kommt würde „Patient_Symptom“ eingefügt, die Tabelle verbindet Symptome mit den Patienten damit bei einer Wiederholung von Patienten von gleichen Symptomen nicht Daten wie Kopfschmerzen und die Deskription davon in der Symptoms Tabelle wenn eine Anzahl von Patienten die gleichen Hauptsymptome hat sollte dieses nicht wiederholt werden sonst hätte Symptoms Tabelle eine wiederholung z.B Kopfschmerzen werden auf 2 verschiedenen Orten da sein Patient_ID muss wiederholbar sein sonst kann ein Patient nur mit einem Hauptsymptom verbunden werden bei einem nächsten Besuch in der Klinik kann der Patient nicht mit einem anderen Hauptsymptom verbunden werden.

5 Diskussion

5.1 Requirements- und Risikoanalyse

Persönliche Interpretation Ein mögliches Risiko was auch oft vorkommen kann ist, dass sich der Zustand des Patienten ändert. Patienten merken es geht ihnen schlechter/besser, als sie es angegeben haben, deshalb könnte das System Probleme haben dies zu speichern und zugleich die bestmögliche Versorgung und Rangierung zu gewährleisten. Ein Problem wäre, auch wenn ein Patient mehrmals kommt, dies wurde aber mit dem Aufbau der SQL Datenbank gelöst.

Die nachfolgenden Versionen der Software würden sehr von dem Feedback der Ärzte abhängen, was wiederum ein Problem darstellen kann, da es wahrscheinlich auf eine kleine Anzahl von Ärzten beschränkt ist und nicht sehr viele Meinungen sich bilden können. Ein weiteres Problem besteht darin, dass der Fragebogen nicht wirklich in die Rangierung hineinbezogen wird, was dazu führen kann, dass viele wichtige Informationen des Patienten nicht verarbeitet werden. Die aber bei der Dokumentation zu Verfügung stehen und der Patient am Ende der Behandlung einen besseren Entlassungsbrief bekommt.

Eine möglich bessere Funktionalität des Systems wäre falls die Einwilligung um teil einer Studien zu sein und gefilmt zu werden auf 2 separate Einwilligungen aufzuteilen und nicht beide in eine Einwilligung zu integrieren.

5.1.1 User Stories

User Story Journey - Patient

In Abbildung 19 betrachten wir die User Story eines Nutzer mit der Rolle Patient. Der User Task umfasst das Symptome schildern, welches durch den Tablet-basierte Fragebogen geschieht. Eine Vorbedingung ist natürlich eine gültige Gesundheitskarte bereitstellen, wenn die Gesundheitskarte auf die Gültigkeit überprüft würde kann der Patient den Fragebogen beantworten durch die Beantwortung der Fragen erhofft sich der Patient eine richtige Einordnung in das System und die Schilderung seiner Symptome werden richtig bearbeitet. Bei der Beantwortung des Fragebogen erhofft sich der Patient mögliche Hilfe zu bekommen falls dies notwendig wäre. Durch die Abschließung des Fragebogen erhofft der Patient das seine Probleme richtig gespeichert sind und der die bestmögliche Pflege bekommt. Der Patient bekommt die Möglichkeit durch seine Einwilligung gefilmt zu werden und Teil einer Studie zu sein. Danach begibt der Patient sich in den Wartebereich.

User Story Journey - Arzt

In Abbildung 20 ... betrachten wir die User Story eines Nutzer mit der Rolle Arzt. Der User Task umfasst das behandeln der Patienten und eine Stellung der Diagnose fest. Zu erst muss der Arzt den Fragebogen auswerten und aus den gewonnen Informationen den Patienten bestmöglich helfen. Nach der Auswertung des Fragebogens und der Anwendung der gewonnenen Informationen an den Patienten beginnt der Arzt mit der nächsten Story Task die beinhaltet die Dokumentation der Diagnose damit der Zustand des Patienten bestmöglich dokumentiert wird, damit die Kollegen bei einem wiederkommen des Patienten mehr Informationen zum arbeiten haben. Eine Story Task die in den Folgeversonen implementiert wird ist die Verbindung mit dem Pager den jeder Arzt hat dies würde zu eine Warnmeldung beinhalten falls ein Patient den Fragebogen sehr auffällig beantwortet hat und der möglich sehr dringende ärztliche Unterstützung braucht.

User Story Journey - Pfleger

In Abbildung 21 betrachten wir die User Story eines Nutzer mit der Rolle Pfleger. Der User Task umfasst das behandeln der Patienten, welches durch das bereitstellen des Tablettts und durch die Führung des Patentes in den Wartebereich ausgefüllt wird. Dazu muss der Pfleger die Gültigkeit der Gesundheitskarte überprüfen und bei Fragen von der Patienten Seite bezüglich des Fragebogens in der Lage sein die zu beantworten. Der Pfleger hat auch die Aufgabe den Patienten auf die Möglichkeit hinzuweisen Teil einer Studie zu sein und dies durch die Unterschrift zu Bestätigen.

User Story Journey - IT-Admin

In Abbildung 21 betrachten wir die User Story eines Nutzer mit der Rolle IT-Admin. Der User Task umfasst das konstruieren des Systems dazu muss der IT-Admin das System verwalten und bei bedarf upzudaten.

Ein potenzielles Problem ist die Farbenblindheit. Im Falle der Patienten sollte dies kein Problem sein, da es keine Fragen im Fragebogen gibt, die auf Farben angewiesen sind. Anders ist dies bei der Betrachtung des Ampel-Dashboards durch medizinisches Personal. Hier müsste ein Farbenblinde freundliches Zusatzfarbschema eingefügt werden.

5.1.2 Akzeptanzkriterien

Akzeptanzkriterien entsteht ein klareres Bild für alle Beteiligten in diesem Projekt, gerade für die Informatiker, falls eine mögliche Veränderung in der Zukunft gemacht wird,

sind die Akzeptanzkriterien hilfreich für das bessere Verständnis des Codes und somit erleichtert es ihnen die Einarbeitung in die Codelogik. Und die Akzeptanzkriterien können später die Dokumentation erleichtern und einen besseren Überblick in die Entwicklung des Projektes liefern, dadurch können Aussagen später leichter getroffen werden und begründet werden.

5.1.3 Systemanforderung

Durch die Systemanforderung, die formuliert sind, ergibt sich ein besseres Gesamtbild über das Projekt und falls es zu einer zukünftigen Änderung kommt wird die geschriebene Systemanforderung die Arbeit den Developern erleichtern und ein besseres Endprodukt bezwecken. Da es mehrere Informationen an die Developer übergibt.

5.2 Datenbank

Entscheidung der Datenbank Typs

MongoDB verfügt über ein flexibles dynamisches Schema, das mit der Entwicklung von Daten, Anwendungen oder Unternehmen leicht geändert werden kann.

SQL Server verfügt über ein festes Schema, das vor dem Einfügen von Daten vordefiniert wird. Ein festes Schema bedeutet, dass das Schema nicht geändert werden kann, wenn sich Daten, Anwendungen oder Geschäftsanforderungen weiterentwickeln.

Da es aber hier um feste Größen geht und die Fragen nicht geändert werden, es geht also immer um gleichen INPUT, ist die Wahl der SQL Datenbank besser, da es für ein dynamisches Schema keinen Bedarf ergibt. Durch die gut definierten Requirements ist die Wahl eine SQL Datenbank besser.

Jedes mal, wenn eine Query ausgeführt wird, die mit der Datenbank verbunden ist, bekommen wir eine neue Verbindungen. . Es ist eine gute Bedingung eine POOL von Connection zu haben und jedes Mal wo die Query ausgeführt wird eine frei Connection zu nehmen damit es nicht zu Verzögerung kommt, wenn eine neue Connection gefordert werden. In diesem Fall haben wir eine POOL von Connection. Die Verbindung mit der Datenbank wird immer wiederverwendet, was zu einer besseren reusability führt.

Schema Datatypes and Character Length

Beim Design des Datenbankschemas muss die Entscheidung auf den Data Type sowie die Charakter Length jedes Feldes gewählt werden. Ein speicherintensives Datenformat ermöglicht die Abspeicherung von Floating Point Values mit einem hohen Grad der Genauigkeit sowie mit einer großen Value-Range. Dies kostet aber erheblich mehr Speicherressourcen wie etwa bool oder INT(8) data values. Ebenfalls sollte die Value-Range ausreichend groß genug gewählt werden, ohne unnötigen Speicher zu verbrauchen. Dies ist in einer produktiv eingesetzten relationalen Datenbank sehr wichtig. Die hier verwendeten Werte wurden gewählt, um beiden Constraints gerecht zu werden. Verbindungen

mit der Datenbank wird immer wiederverwendet, was zu einer besseren reusability führt.

Proof-Of-Concept vs Production

Aktuell (in unserem Proof-Of-Concept) haben wir die Daten im gleichen Docker container wie die Datenbank selbst. Im TEDIAS Projekt in Production wird ein Microservices Ansatz gefahren. Das heißt, die Daten liegen nicht lokal im gleichen container wie die Datenbank. Es bräucht noch eine Methode die Daten in die Datenbank zu ingesten wie z.B. ein shared Docker Volume.

Durch das CAP Theorem kommt man zu Schluss das die relational data wie SQL ist effizienter ist da es ein (schema on write) ist und durch die Tatsache da JSON like Datenbanken wie MongoDB ist (schema on read) aber ineffizienter.

Da wir eine fest definierte Datenbank haben wird die Schema immer gleich bleiben.

Die Python Script hat den job einer kleinen ETL hier werden die Files von JSON extrahieren und eingefügt in die Tabelle die schon vorgeladen ist. Es besteht kein Nutzen für eine schema on read da in dieser Methode die Schema beim lesen des Files generiert wird aber in diesem Projekt werden keine Dynamischen Schemas gebraucht.

In diesem Projekt ist die Partition tolerance nicht sehr wichtig deshalb war das ein wichtiges Argument für die Nutzung von SQL.

Falls sich die Fragebögen ändern sollten, wäre eventuell MongoDB besser Dies ist in näherer Zukunft aber nicht vorgesehen.

Falls das Datenvolumen sehr groß wird, dann ist eine relationale Datenbank (wie MySQL) effizienter.

5.2.1 Datenschutz

Da es in diesem Projekt um die Anwendung in einem Krankenhaus geht, würde eine SQL die horizontal skalierbar ist die bessere Wahl sein, da die Wahrscheinlichkeit deutlich geringer ist für einen Datenmissbrauch, weil es nur über ein System geht, in diesen Fällen ist die zentralisierte Version sichere. Mit der Wichtigkeit der Daten (es handelt sich um persönliche Patientendaten) wäre eine horizontal skalierbar besser geeignet, da ein zentralisierter Zugang sichere wäre und keine unnötigen Verbindungen braucht um zu funktionieren. Die Datenverschlüsselung wäre auch erheblich leichter, da es sich nur um ein zentrales System handelt.

5.3 Allgemein

Blinde Patienten müssten Krankenschwestern das Formular für sie ausfüllen lassen, was sie daran hindern könnte, frei zu sprechen, da andere Patienten es belauschen könnten. Eine mögliche Lösung dafür wäre, für solche Fälle einen separaten Raum zu haben, in dem Patient und Krankenschwester das Formular frei ausfüllen können.

Ältere Menschen könnten auch Hilfe benötigen, da sie oft nicht die besten Kenntnisse im Umgang mit Technologie haben. Eine Krankenschwester könnte ihnen helfen, besonders

wenn sie ein schwaches Gehör oder Sehvermögen haben. Auch in diesem Fall könnten die ausgewiesenen Einrichtungen genutzt werden.

Menschen, die aufgrund einer Behinderung nicht wirklich alle ihre Beschwerden richtig wiedergeben können, können dies ein Problem für die Software darstellen.

Ein gutes Beispiel in der "Appendix Ethik" Unsere Kollegen aus Taiwan haben eine ähnliche Studie mit vielversprechenden Ergebnissen durchgeführt und diese Arbeit bezieht sich auf sie, um zu versuchen, ihre Ergebnisse bestmöglich auf dem europäischen Markt zu verwalten. Durch die speziellen Methoden, die in dem Appendix ausführlich erläutert sind, entsteht eine mögliche Alternative, die befolgt werden kann.

6 Konklusion

Das Hauptziel, Ziel des Projekts TEDIAS ist die Digitalisierung des ärztlichen Untersuchungsprozesses damit die Registrierung von Patient schneller und effizienter gestaltet wird, von diesem Projekt war es dem Medizinischen Personal in der Klinik zu helfen bei der Unterbringung der Patienten damit so wenig Menschen Ressourcen in dieser Arbeit eingesetzt wird. Dafür wurde ein Triage-System zur automatischen Kategorisierung der Ernsthaftigkeit von Patientensymptomen für ein Krankenhaus entwickelt.

Zuerst wurde eine ausführliche Anforderungsanalyse erstellt mit einer Use Patient Journey geschrieben nach dem IREB Standard (der auch in der Appendix zu finden ist), eine Unterscheidung zwischen den User Stories und Use Cases wurde ausführlich erläutert und noch dazu wurden Akzeptanzkriterien erstellt und beschrieben. Nach der Anforderungsanalyse wurde eine Risikoanalyse erstellt die potenzielle und vorhandene Risiken entdeckt hat. Danach wurde eine SQL Datenbank erstellt mit der Folgerung der Risikoanalyse zugleich wurden die Systemanforderungen und zusammen mit den nicht-funktionalen Anforderungen geschrieben. Der Hauptfokus der Softwarequalitätsattribute der Software wurde sowohl auf die kontinuierliche Verfügbarkeit als auch auf den kontinuierlichen Bereitschaftszustand gelegt.

In dieser Arbeit wurde auch die Option eingefügt das ein Patient mehrmals kommen kann was in der Original Version nicht der Fall war, in der war es vorgesehen das ein Patient nur einmal kommen kann und dem Patienten kann ein Hauptsymptom zugeordnet werden. Dazu wurde eine Python Skript geschrieben mit der Durchführung von diesem Skript wird eine SQL Datenbank erstellt. Durch den Fragebogen kommt ein JSON File als Output der durch die Python Skript erkannt wird und in die entsprechende Datenbank gespeichert wird. Die komplette Arbeit wurde auch "dockerisiert" und damit steht auf allen Plattformen für die Nutzung bereit. In dieser Arbeit wird der Docker default Container benutzt, es wurde auch ein yaml Docker compose file geschrieben, in diesem Fall wurde ein Container gebraucht wo die Daten in gleichen Container leben wie die Datenbank selbst. Wegen der Sensibilität der Daten werden alle Antworten von einem Patient mit genau den Patient in Verbindung gebracht. Falls dieses Projekt in das TEDIAS Projekt eingebunden werden, sollte müsste eine leichte Modifikation gemacht werden, was eine Architektur Entscheidung wäre, ob das Projekt eine Microservices Ansatz Entscheidung trifft wo jeder Service ein eigenes Docker image ist. Mit diesen Ansatz werden die Daten auf einem anderen Container erstellt als die Datenbank, in diesem Falle würden mehrere Docker Container entstehen und diese müssten über ein Docker Network kommunizieren man müsste diesen Image dazunehmen anstatt die wie hier lokal zu nehmen dazu wäre es erforderlich über den gemounteten Volume zuzugreifen. Ansonsten bleibt alles gleich, falls dies eingebunden werden soll ist es dem Team also bewusst was es zu tun hat aber für diesen Sinn und Zweck reicht es mit einem Container zu arbeiten. Dazu wurden noch unit tests geschrieben und ausgeführt um sicherzustellen das auch der Code funktioniert und dies auch zu den guten Codierungspraktiken dazugehört.

7 Ausblick

Vorschläge für zukünftige Funktionen:

- Red-Flag-Klassifizierungs- und Ranking-System – Red-Flag-Stapelung und/oder -Schweregrad (angezeigt in roter Farbintensität) verbessert das Ranking der Warteliste. Durch einen Update in den Folgeprojekten würde es sinnvoll sein Red Flags zu unterteilen. Eine passende Unterteilung die sich anbieten würde wäre in die Stufenform in dieser würden dann Redflags der Grades 1,2,3 existieren den Grad Nummer 1 würde eintreffen wenn der Patient den Fragebogen so geantwortet hat das eine sofortige Behandlung notwendig wäre. Die Trigger Antworten für den Grad 2 und 3 sollte die Ärztin bestimmen.
- Ranking-Aufstieg durch Wartezeit – es wird eine Zeitschwelle festgelegt, so dass ein Patient, der eine erhebliche Zeit mit Warten verbracht hat, sein Ranking erhöht. Beispielsweise wird ein Patient mit einer grünen Flagge in die orange Kategorie hochgestuft.
- Zusammenfassungsansicht – Vollständige Patientenstatistiken werden übersichtlich an einem Ort angezeigt. Kategorie, Alter, Symptome, Behandlung, falls verabreicht und geplant. Alle mit dem System verglichenen ungefähren Warte- und Behandlungszeiten. Die durchschnittliche tägliche Wartezeit sollte sowohl an Werktagen als auch an Wochenenden und Feiertagen erfasst und entsprechend verglichen werden. Mit der Aufnahme dieser statistischen Funktionen in einer zukünftigen Version der Software würde mir der Prozess der Datenanalyse viel einfacher und wichtige Entscheidungen würden mit Zahlen und verpackt, so dass Ärzte möglicherweise die Arbeitszeiten neu organisieren können, um zu sehen, an welchen Tagen es gibt zum Beispiel an Wochenendtagen mehr Patienten, so dass an diesen Tagen mehr Ärzte auf dieser Station sein sollten.
- Eine Statistik basierend auf der Wartezeit von Patienten, gruppiert nach ihrem Hauptsymptom, könnte hilfreich sein, um den Fragebogen gegebenenfalls neu zu organisieren. Um zu sehen, ob die Wartezeit einer Hauptsymptomgruppe im Durchschnitt unverhältnismäßig länger ist als die Norm.
- Statistiken – Langfristige Statistiken, die rückwirkend angezeigt werden, um den Behandlungserfolg und die Effizienz zu beurteilen, dies könnte bei der langfristigen Datenanalyse helfen, wobei möglicherweise einige Daten geclustert werden.
- Eine bessere Sortierfunktion: Erweiterte Sortierfunktion, z.B gruppieren Sie nach Priorität, Farbe, Wartezeit, Symptomen usw. Die Krankengeschichte des Patienten sollte irgendwie in das Rankingverfahren einbezogen werden.
- Fragen der Patienteneinwilligung müssen berücksichtigt werden (dies wird in anderen Teilen der TEDIAS-Projekte behandelt), da die Menschen nicht sehr daran interessiert sind, einige ihrer Erfahrungen zu teilen.

- Dringende Info-Popups – Blasen mit wichtigen, dringenden Informationen (z. B. Penicillinallergie), die nach Belieben geschlossen werden können.
- Digitale Patientenakte für einen besseren Informationsaustausch mit anderen medizinischen Mitarbeitern während des Krankenhausbesuchs.
- Ein internes Kommunikationssystem für Ärzte und medizinisches Personal, das genaue Informationen gewährleisten würde, wird sofort weitergegeben, mit der Hinzufügung von TO DO-Listen (z. B. Blutdruckmessung) mit Feedback-Slots, die das Personal ausfüllen muss.
- Sobald ein Patient Gefährdungsvariablen wie Allergien in das System eingegeben hat, werden diese bei jedem Aufruf der Patientendaten ebenfalls angezeigt. Da in diesem Projekt die Möglichkeit besteht das Patienten wiederkommen können.
- Falls der behandelnde Arzt den Patienten zu einem anderen Arzt der in der Klinik arbeitet zu Untersuchung schickt wie z.B Kardiologe sollte dies im System vermerkt werden und den Patienten nicht aus der Liste löschen sondern in einen Speziellen zustand befördern.

8 Bibliography

- [1] Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA TEDIAS Pressemitteilung
<https://www.ipa.fraunhofer.de/de/presse/presseinformationen/automatisierungstechnik-entlastet-klinikaerzte.html>
- [2] TEDIAS Forum Gesundheitsstandort
<https://www.forum-gesundheitsstandort-bw.de/projekte/ministerium-fuer-wirtschaft-arbeit-und-wohnungsbau-2/tedias-test-und-entwicklungszentrum-fuer-digitale-anamnesesysteme-entwicklung-eines->
- [3] M. F. Kasim, D. Watson-Parris, L. Deaconu, S. Oliver, P. Hatfield, D. H. Froula, G. Gregori, M. Jarvis, S. Khatiwala, J. Korenaga, J. Topp-Muggleston, E. Viezzer and S. M. Vinko, "Up to two billion times acceleration of scientific simulations with deep neural architecture
- [4] Johannes Bergsmann Requirements Engineering für das agile Softwareentwicklung 2 Auflage 2018
- [5] [Macaulay 1993]. L. Macaulay: Requirements Capture as a Cooperative Activity. In: Proceedings of the 1st IEEE International Symposium on Requirements Engineering, 1993
- [6] Pohl, Klaus, and Chris Rupp. Basiswissen Requirements Engineering : Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag, 2015.
- [7] [Pohl 1996] K. Pohl: Process-Centered Requirements Engineering. Research Study Press, Advanced Software Development, Taunton, Somerset, 1996.
- [8] [Glinz und Wieringa 2007] M. Glinz, R. Wieringa: Stakeholders in Requirements Engineering. IEEE Software 24, 2, 2007, S. 18– 20.
- [9] [Potts et al. 1994] C. Potts, K. Takahashi, A. Antón: Inquiry-Based Requirements Analysis. IEEE Software 11, 2, 1994
- [10] [Kano et al. 1984] N. Kano, S. Tsuji, N. Seraku, F. Takahashi: Attractive Quality and Must-be Quality. Quality – The Journal of the Japanese Society for Quality Control, Vol. 14, Nr. 2, 1984, S. 39– 44.
- [11] Systematisches Requirements Engineering Christof Ebert Released January 2019 Publisher(s): dpunkt ISBN: 9783864905629
- [12] Systematisches Requirements Engineering Christof Ebert Released January 2019 Publisher(s): dpunkt ISBN: 9783864905629
- [13] Systematisches Requirements Engineering Christof Ebert Released January 2019 Publisher(s): dpunkt ISBN: 9783864905629

- [14] [Beck 2003] Beck,K.: Test Driven Development by Example. Addison-Wesley 2003
- [15] L. Williams, E.M. Maximilien, M. Vouk, Test-driven development as a defect-reduction practice,
- [16] M.M. Muller, O. Hagner, Experiment about test-first programming, presented at Empirical Assessment In Software
- [17] [Jacobson et al. 1992] I. Jacobson, M. Christerson, P. Jonsson, G. Oevergaard: Object Oriented Software Engineering – A Use Case Driven Approach. Addison-Wesley, Reading MA, 1992.
- [18] [OMG 2007] OMG: Unified Modeling Language: Superstructure, Version 2.1.1. OMG document formal/2007-02-05.
- [19] [Stachowiak 1973] H. Stachowiak: Allgemeine Modelltheorie. Springer-Verlag, Wien,
- [20] Tiwana,A; Keil, M.:The One Minute Risk Assessment Tool.Communication of the ACM, Vol. 47, No. 11,2004,S 137-146.
- [21] Schmidt,K.: IT-Risikomanagment.In:Tiemeyer,E(Hrsg.): Handbuch IT-Managment. Carl Hanser Verlag, München 2009,S 537-574.
- [22] D. Zburivsky, L. Partner, Designing Cloud Data Platforms, ISBN 9781617296444, Manning Publications, 2020.
- [23] H. Nguyen, H. Pham, C. Chinm The Analytics Setup Guidebook, Holistics Software, 2020.
- [24] (Patrick Zschech, Christoph Sager, Philipp Siebers, Maik Pertermann,2020, S.3).
- [25] Big Data, Fast Data and Data Lake Concepts Natalia Miloslavskaya and Alexander Tolstoy National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
- [26] State of Art of Data Mining and Learning Analytics Tools in Higher Education Mohammed Salihoun, EMSI the High School of engineering group
- [27] Design of a reference architecture for an IoT sensor network. Dr. Jairo Alejandro Gómez Escobar 29.4.2019 International Center for Tropical Agriculture (CIAT). Colombia
- [28] R. Kimball, M. Ross, The Data Warehouse Toolkit, ISBN-10 : 1118530802, Wiley, 2013.
- [29] Business Intelligence Technology, Applications, and Trends Muhammad Obeidat

- [30] Batty, Mike, Arun Tripathi, Alice Kroll, Peter Wu Cheng-sheng, David Moore, Chris Stehno, Lucas Lau, Jim Guszczka, and Mitch Katcher. 2010. "Predictive Modeling for Life Insurance: Ways Life Insurance Can Participate in the Business Analytics Revolution" Society of Actuaries
- [31] Mayer-Schönberger, Victor, and Kenneth Cukier. 2014. Big Data: A revolution That Will Transform How We Live, Work, and Think. Boston: Eamon Dolan/Mariner Books
- [32] Hill, Shawndra, Foster Provos and Chris Volinsky. 2006. Network-Based Marketing: Identifying Likely Adopters via Consumer Networks. Statistical Science 21(2) 256-276
- [33] O’Gara, Maureen (July 26, 2013). "Ben Golub, Who Sold Gluster to Red Hat, Now Running dotCloud"
- [34] . Pahl, "Containerization and the PaaS Cloud," in IEEE Cloud Computing, vol. 2, no. 3, pp. 24-31, May-June 2015, doi: 10.1109/MCC.2015.51.
- [35] Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization.
- [36] Felter, W., Ferreira, A., Rajamony, R., Rubio, J. (2014). An updated performance comparison of virtual machines and linux containers. technology, 28, 32.
- [37] Seo, K.-T., Hwang, H.-S., Moon, I.-Y., Kwon, O.-Y., Kim, B.-J. (2014). Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud.
- [38] [10] Van der Aalst, W., Weijters, T., Maruster, L. (2004). Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on, 16(9), 1128-1142.
- [39] Vase, T. (2015). Advantages of Docker.
- [40] Waldspurger, C. A. (2002). Memory resource management in VMware ESX server. ACM SIGOPS Operating Systems Review, 36(SI), 181-194
- [41] Joy, A. M. (2015). Performance comparison between Linux containers and virtual machines. Paper presented at the Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in.
- [42] Edwin Schicker Datenbanken und SQL Eine Praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL 5 Auflage
- [43] Vgl. Rasche et la. (2018) Entrepreneurship im Gesundheitswesen III: Digitalisierung - Innovationen - Gesundheitsversorgung, Wiesbaden, Deutschland: Springer Gabler, S. V

- [44] Jürgen Klauber Max Geraedts Jörg Friedrich Jürgen Wasem (Hrsg.) Krankenhaus-Report 2019 Das digitale Krankenhaus
- [45] Ausgangslage und Ziel im Überblick – Ein Leitbild von und für Ärztinnen und Ärzte Digitalisierung in der ambulanten Gesundheitsversorgung Tarja Zingg, Reinhold Sojer, Fabian Röthlisberger
- [46] Digitalisierung im Gesundheitswesen Marion Müller <https://www.degruyter.com/document/doi/10.9785/ag-2018-632225/html>
- [47] Digitalisierung im Gesundheitswesen Ursula Sury <https://link.springer.com/content/pdf/10.1007/s00287-020-01317-9.pdf>
- [48] (Vgl. Publicitaions office of the European union 2010) Digitale Agenda für Europa <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=LEGISSUM:si0016>
- [49] (Vgl. Publicitaions office of the European union 2008) Telemedizinische Systeme und Dienste <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=LEGISSUM:sp0003>
- [50] (Vgl. Publicitaions office of the European union 2005) Online-Gesundheitsfürsorge: Bessere Gesundheit und Gesundheitsversorgung durch Nutzung von Informations- und Kommunikationstechnologien <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=LEGISSUM:124226f>
- [51] Vgl. Bundesgesundheitsministerium (2020) – Digitalisierung im Gesundheitswesen <https://www.bundesgesundheitsministerium.de/e-health-initiative.html#c2849>
- [52] G2 No Code Platform <https://www.g2.com/categories/no-code-development-platforms#learn-more>
- [53] Gartner Low Code Platform Analysis <https://www.gartner.com/reviews/market/enterprise-low-code-application-platform>
- [54] Building a Data Lake on Google Cloud Platform with CDAP <https://medium.com/cdapio/building-a-data-lake-on-google-cloud-platform-with-cdap-a1>
- [55] Cohn,M.: User Stories für die agile Software-Entwicklung mit Scrum,XP u.a mitp-Verlag, Frechen,2010
- [56] Docker Architektur <https://docs.docker.com/get-started/overview/>
- [57] Architektur der Virtuellen Maschine <https://www.docker.com/resources/what-container/>
- [58] Amit M Potdar et al. / Procedia Computer Science 171 (2020) 1419–1428 Ami

- [59] Die Struktur einer typischen Plattform-Pipeline https://developer.here.com/documentation/pipeline/dev_guide/topics/working-with-pipelines.html
- [60] cap theorem <https://data-science-blog.com/blog/tag/sql/>
- [61] Wirkung von der Corona Pandemie auf die Meinung der Menschen bezüglich der Digitalisierung <https://www.bitkom.org/Presse/Presseinformation/Corona-Unternehmen-spueren-wirtschaftlichen-Nutzen-der-Digitalisierung>
- [62] Elektronische Patientenakte <https://e-health-com.de/details-news/zwei-drittel-der-deutschen-wollen-e-rezept-und-elektronische-patientenakte-nutzen/>
- [63] Elektronische Patientenakte2 <https://www.bah-bonn.de/presse/bah-gesundheitsmonitor/presse-detailseite/fast-drei-viertel-befuerworten-elektronische-patientenakte/>
- [64] D. Kenyon and J. H. P. Eloff, "Big data science for predicting insurance claims fraud," 2017 Information Security for South Africa (ISSA), 2017, pp. 40-47, doi: 10.1109/ISSA.2017.8251773.
- [65] Hsing AW, Ioannidis JP. Nationwide population science: lessons from the Taiwan National Health Insurance Research Database. JAMA Intern Med. 2015;175:1527–1529.
- [66] Gordis L. Epidemiology. 5th ed. Philadelphia: Elsevier; 2014. p. 163.
- [67] Eichler HG, Abadie E, Breckenridge A, Flamion B, Gustafsson LL, Leufkens H, et al. Bridging the efficacy-effectiveness gap: a regulator's perspective on addressing variability of drug response. Nat Rev Drug Discov. 2011;10:495–506.
- [68] Chen YC, Yeh HY, Wu JC, Haschler I, Chen TJ, Wetter T. Taiwan's National Health Insurance Research Database: administrative health care database as study object in bibliometrics. Scientometrics. 2011;86:365–380
- [69] Health and Welfare Data Science Center Health and welfare data science center data application. 2018 .
- [70] Ministry of Health and Welfare 2017 Announcement for institutional review boards evaluation of qualifications. 2017
- [71] Bachelorarbeit zu TEDIAS Project, "Development of a Hospital Triage System to automatically categorize Patient Symptoms", 2022, Heidelberg University, Physics Faculty
- [72] Land fördert innovative Gesundheitsprojekte <https://www.baden-wuerttemberg.de/de/service/presse/pressemitteilung/pid/land-foerdert-innovative-gesundheitsprojekte/>

All online sources were available at time of publication.

9 Appendices

9.1 Appendix A: User Dokumentation

Rolle (R, User Role)

- Namenskonvention R: <<Name >>
- Eine Rolle ist abstrakte Zusammenfassung von Bedürfnissen, Interessen, Erwartungen, Verhalten und Verantwortlichkeiten.
- Eine Rolle wird beschrieben durch
 - ihren Namen, (z.B. FreizeitsportlerIn)
 - ihre Aufgaben (UT)
 - Erfolgskriterien (d.h. wann hat die Rolle ihre Arbeit gut gemacht)
 - Kommunikationspartner bei der Aufgabendurchführung
 - Innovationsgrad (d.h. gab es die Rolle vorher schon)
 - Grobes Nutzerprofil
 - bzgl. Aufgaben
 - bzgl. Softwaresystem
- Zu jeder Rolle gibt es mehrere Personae, die zusammen das Spektrum des Nutzerprofil ausfüllen
- Verlinkung in Jira: zu Persona mit "is detailed by" und zumindestens einer User Task mit "is related to"

Persona (P)

- Namenskonvention P: <<Name >>
- Eine Persona ist möglichst realitätsnahe Beschreibung einer fiktiven Person, die stellvertretend für eine Gruppe von NutzerInnen steht
- Eine Persona ist beschrieben durch
 - einen Namen
 - biographische Angaben zu Alter, Geschlecht, Lebensumständen
 - Kenntnisse und Einstellung zur Technik allgemein
 - Bedürfnisse in Bezug auf die Aufgaben der zugehörigen Rolle und die Aktivitäten der Rolle, in denen die Software genutzt werden sollen. Sie sind unabhängig von der IT-Unterstützung
 - Ärgernisse in Bezug auf die Aufgaben oder Feature der Software (begründet durch die Bedürfnisse)

- Ideale Features der Software, über die sich die Persona freuen würde (begründet durch die Bedürfnisse)
- Die Personae einer Rolle sind möglichst unterschiedlich, um die Spezifika der Software (user stories und Systemfunktionen) gut zu begründen
- Verlinkung in Jira: zu Rolle mit "is part of"

Die möglichst detaillierte Beschreibung der „user patient journey“ würde nach den „IREB„ Prinzip geschrieben

Aufgaben (UT, User Task)

- Namenskonvention UT: <<Name >>
- Der Name einer UT muss der Grammatik: <Objekt ><Verb >genügen (z.B. Strecke planen oder als Hauptwort Streckenplanung)
- Eine UT ist eine Verantwortlichkeit einer Rolle, die ein zusammenhängendes Ergebnis produziert, z.B. Ergebnis = FreizeitsportlerIn hat Weg für ihre Fahrt. Sie ist unabhängig von der IT-Unterstützung (Aufgabenebene)
- Eine UT ist beschrieben durch
 - Ziel:
 - Eingriffsmöglichkeiten:
 - Ursachen:
 - Priorität:
 - Durchführungsprofil:
 - Vorbedingung:
 - Info-In:
 - Info-Out:
 - Ressourcen:
- Verlinkung in Jira: zu genau einer Rolle mit "is related to" und zu mehreren Sub-Tasks mit "is detailed by"

Teilaufgaben (E, Sub-Task = Epic)

- Namenskonvention ST: <<Name >>
- Der Name muss der Grammatik:<Verb ><Objekt >genügen (z.B. Erstelle eine Auswahl von Optionen).

- Eine E ist einer UT zugeordnet. Sie entspricht einem abgeschlossenen Teilziel der UT. Sie liefert ein Teilergebnis und ist nicht gut bei der Durchführung zu unterbrechen.
- Die E ist unabhängig von der IT-Unterstützung (Domänenebene).
- Die Description muss die E näher erklären. Insbesondere durch Beschreibung des Ergebnis der Teilaufgabe und Varianten und Probleme bei der Erstellung dieses Ergebnis und dazu passende Lösungen (User story und Systemfunktionen). Dabei sind insbesondere ein oder mehrere Schritte (ST) anzugeben und ggf. eine Unterstützung für jeden Schritt.
- Varianten beschreiben Umstände in den Schritten, die zu einer Abweichung vom normalen Ablauf und/oder Ergebnis des Schritts führen. Sie sind unabhängig von der IT-Unterstützung
- Probleme beschreiben Umstände in den Schritten, die dazu führen, dass das gewünschte Ergebnis nicht oder nur schlecht erreicht werden kann. Sie sind unabhängig von der IT-Unterstützung
- Zu jedem Problem muss eine Lösung (User Story oder Systemfunktion) formuliert werden, die das Problem erkennbar löst.
- Die Description muss aus der Sicht der Rolle und in Gegenwartsform formulieren werden, (z.B. falsch: eine Auswahl wird erstellt; falsch: man kann eine Auswahl erstellen).
- Die Description muss vollständig sein, d.h. beinhaltet alle wichtigen Varianten und Probleme und deren Lösungen.
- Alle Schritte, die für die E wichtig sind, werden beschrieben, auch wenn sie gar nicht von der Software unterstützt werden sollen.
- Verlinkung in Jira: zu genau einer User Task mit "is part of" und zu mindestens 2 User Stories und Systemfunktionen mit "Issue in Epic"-Link

Namenskonvention US: <<Name >>

- Der Name muss der Grammatik Als<Rolle>möchte ich <Aktivität = etwas tun >, damit ich <Begründung = Vorteile erziele >genügen
- US beschreiben EINE Funktionalität der Software. Sie ist also IT-abhängig.
- US sind problem-orientiert, nicht lösungsorientiert (keine GUI-Details)
- Begründung einer US ist motiviert durch
 - Ein Bedürfnis oder eine Frustration einer Persona

- Eine Verwendung des Ergebnis in einer anderen US, Epic oder User Task
- Die Beschreibung muss die US näher erklären, z.B. durch Angabe von benötigten und erzeugten Daten, Vor- bzw. Nachbedingung oder durch Angabe von Akzeptanzkriterien .
- US erfüllen die Kriterien für gute Anforderungen
 - Eindeutige Begrifflichkeiten
 - Schätzbar (kann geplant und priorisiert werden)
 - Als Menge ohne explizite Abhängigkeiten
 - Als Menge vollständig
 - Als Menge konflikt-frei
 - Als Menge ohne Redundanzen
- Verlinkung in Jira: zu mindestens einer Sub-Task mit "Issue in Epic"-Link, evtl. zu noch anderen Sub-Tasks mit "relates to"-Beziehung (bei Nennung mehrerer Rollen) und zu verwandten Systemfunktion mit "is related to"

In diesem Projekt würde Jira nicht benutzt deshalb ist der letzte Punkt nicht erfüllt

9.2 Appendix B: Ethik in dem Gesundheitswesen

Die Versicherungskosten steigen weltweit, mit Hilfe neuer Gesetze und Betrug. Beide Faktoren sind problematisch, da das neue Gesetz von den Versicherern nicht einfach überwacht wird und Betrug proaktiv schwer zu erkennen ist. Daher ist es schwierig, den steigenden Trend der Kosten zu kontrollieren. Daher untersuchen wir den Einsatz von Big Data und Big Data Science zur Vorhersage von Versicherungsbetrug. Big Data, Data Science und Predictive Analytics wurden in der kurzfristigen Versicherungsbranche mit Anwendungsfällen in Entwicklungsländern angewendet. Diese Studie wurde innerhalb der Grenzen der Datenschutzgesetzgebung durchgeführt, die Standards für die Aufbewahrung und Weitergabe von Daten vorschreibt – wir schlagen eine Methode zur Wahrung der Privatsphäre vor, um Betrug bei Versicherungsansprüchen vorherzusagen.^[64]

Die zunehmende Verfügbarkeit, Größe und Detailliertheit elektronischer Patientenakten (EHRs) bieten beispiellose Möglichkeiten für die Forschung. Zu den Vorteilen von elektronischen Patientenakten zählen erhöhte statistische Aussagekraft, Geschwindigkeit, große Breite, relativ niedrige Kosten, repräsentative Bevölkerungsabdeckung, Vollständigkeit der Nachsorge und die Fähigkeit, Interventionen in der routinemäßigen klinischen Versorgung zu bewerten.^[65] Durch die Verknüpfung von EHRs mit Krankheitsregistern und anderen Ressourcen kann ihr Nutzen weiter ausgebaut werden. In der Zwischenzeit kontrollieren randomisierte kontrollierte Studien (RCTs) bekannte und unbekannte Störfaktoren; Daher gelten sie als Goldstandard zur Messung der Wirksamkeit von Interventionen.^[66] Durch die Verknüpfung von EHRs mit Krankheitsregistern und anderen Ressourcen kann ihr Nutzen weiter ausgebaut werden. In der Zwischenzeit kontrollieren randomisierte kontrollierte Studien (RCTs) bekannte und unbekannte Störfaktoren; Daher gelten sie als Goldstandard zur Messung der Wirksamkeit von Interventionen.^[67] Als wichtige Quelle für Daten aus der Praxis sind EPAs zu einem praktischen Werkzeug in der medizinischen Forschung geworden. Mithilfe von EHR können Forscher die Wirksamkeit der Behandlung messen, Trends bei der Inzidenz und Prävalenz von Krankheiten identifizieren und mögliche Ursachen weiter untersuchen. Unter den nationalen EHR-Datenbanken der Welt ist Taiwans National Health Insurance Research Database (NHIRD) einzigartig. Diese große Datenbank, die die Daten von 23 Millionen Menschen in Taiwan enthält, wurde zuvor von Chen et al.^[68] NHIRD wurde jedoch im Jahr 2016 aktualisiert. Diese Datenbank hat sich von der administrativen Verwaltung geändert, wurde mit anderen Datensätzen integriert, um die Verknüpfung fortzusetzen, und hat den Datensatz mit seinen vollständigen demografischen Daten veröffentlicht. NHIRD bietet jetzt mehr Flexibilität für die wissenschaftliche Forschung. In diesem Artikel stellen wir die neueste Version von NHIRD vor, stellen die wichtigsten Funktionen vor, auf die Sie achten sollten, und beschreiben ihre Stärken und Schwächen. Nach Erhalt der Genehmigung müssen Forscher zu den Außenstellen des Data Science Center reisen, um ihre Datenanalyse durchzuführen. Die Analyse von NHIRD-Daten ist komplex, und es gibt keine strukturierte Schulung zur Verwendung von NHIRD. Als Ergebnis stellt das Data Science Center einen fiktiven Datensatz mit 100.000 Objekten bereit, um Forschern beim Schreiben von Syntax für statistische Analysen zu helfen. Wenn Forscher das Data Science Center betreten, dürfen sie die Computer und

die bereitgestellte Software, einschließlich SAS, Stata, R und SPSS, verwenden, um ihre Datenanalyse durchzuführen ^[69]. Die Genehmigung der Ethikkommission ist erforderlich, wenn die Verwendung von NHIRD-Daten beantragt wird. Es gibt 27 lizenzierbare institutionelle Prüfungsausschüsse, und alle werden vom Ministerium für Gesundheit und Wohlfahrt ^[70] beaufsichtigt und reguliert. Um die Privatsphäre des Einzelnen zu schützen, tragen alle Data Science Center-Datensätze Spitznamen. Diese persönliche ID, Geburtsdatum und Name werden verschlüsselt und dieses anonyme Verfahren wurde von einer unabhängigen Drittorganisation ^[69] genehmigt. Um die Privatsphäre der Teilnehmer besser zu schützen, sind die NHIRD-Datensätze außerhalb des Data Science Center nicht zugänglich, was bedeutet, dass Forscher auf diese Datensätze nicht zur Analyse im Data Science Center zugreifen können. Beim Zugang zum Data Science Center dürfen Forschende keine Aufzeichnungsgeräte, einschließlich Papier und Stifte, mitbringen. Darüber hinaus muss ihre statistische Analysesyntax vom Data Science Center validiert werden, bevor der bereitgestellte Computer und die Software verwendet werden. Auch die Analyseergebnisse werden vor dem Export vom Data Science Center validiert. Ergebnisse mit weniger als 3 Objekten sollten nicht ausgegeben werden, um eine Neudefinition zu vermeiden.

9.3 Appendix C: User Patient Journey Überblick

Issue Type	Summary
User Role	R: Patient
User Task	UT: Describe symptoms
Epic	ST: Providing my health card
Story	US: as a patient I want to be able to scan my health card so that I can use the dashboard
Story	US: As a patient I would like to be able to fill out a consent form in order to agree to being part of a study and to being filmed, respectfully.
System Function	SF: Give information and consent
Epic	ST: Answer questionnaire
Story	US: As a patient I would like to be able to use the "HELP" button so that I can help me in case of ambiguity or problem the medical staff
Story	US: As a patient I would like to be able to answer the questionnaire so that the doctor gets a good overview of my symptoms and I don't have to wait a long time
System Function	SF: Adequately fill out the questionnaire
Story	US: As a patient I would like to be able to finish the questionnaire so that the results are transferred into the system.
System Function	SF: Closing/ending of the questionnaire
User Role	R: Doctor
User Task	UT: Diagnosing
Story	US: As a doctor, I would like to be able to document my final diagnosis from the patient so that it is recorded in the system and the colleagues can help the patient in the best possible way
Epic	ST: Read pager notification
Story	US: As a doctor I would like to receive a notification if there is an emergency so that I can help the person concerned as soon as possible.
Story	US: As a doctor I would like to be able to receive a notification if a patient has specifically answered the questionnaire and his answer needs a possible quick treatment
Story	US: As a doctor, I would like to be able to see if a patient has answered the questions in such a way that he is not a case for the emergency room so that I ask the triage force to perform a triage in a timely manner, so that the patient can be referred to "MVZ or ÄBD" as soon as possible, so as not to occupy the emergency room unnecessarily.
Epic	ST: Evaluation of the questionnaire
Story	US: As a doctor I want to be able to click on the ICON and see the patient's results so that I can improve my diagnosis
System Function	SF: Open ICON
Epic	ST: Grading of the questionnaire
System Function	SF: Review the questionnaire
Story	US: As a doctor I want to be able to give my grade to the ranking of the patient so that we can check the efficient of the system and possibly the system needs an update
System Function	SF: Identify Problems
Story	US: As a doctor I would like to store my grade in the system and the storage so that we get a long-term study to evaluate the system
Story	US: As a doctor I would like to get a warning if my note and those of the system are much apart so that we may update the system
User role	R: Nurse
User Task	UT: Treat patient
Epic	ST: Deploying the tablet
Story	US: As a nurse I would like to point out in the patients that we are under data collection law and we are under silence
Story	US: As a nurse I would like to point out to the patients that we need a commitment so that we can film the patient and the patient becomes part of a study
Epic	ST: Lead patients to the waiting area
Story	US: As a nurse I want to be able to see if the patient has pressed the "HELP" button so that I can help the patient if necessary
System Function	SF: Get the notification
Story	US: As a nurse I want to be able to see if the patient finishes the questionnaire so that I can pick up the tablet and there can be the next patient
Role	R: IT Admin
User story	UT: IT-system construction
Epic	ST: Maintaining the IT-system
Story	US: As an IT admin I would like to be able to update the system if necessary
Story	US: As an IT admin I would like to be able to connect the devices on the secured WiFi of the clinic

Table 1: Patient User Journey

10 Ehrenwörtliche Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.



Heidelberg, 01.07.2022

Ort, Datum der Abgabe

Unterschrift