

Methode zur Gestaltung sicherer präskriptiver Systeme für die Therapieunterstützung

Der Wirtschaftswissenschaftlichen Fakultät

der Universität Leipzig

eingereichte

DISSERTATION

zur Erlangung des akademischen Grades

DOKTOR-INGENIEUR

DR.-ING.

vorgelegt

von M. Sc. Philipp Skowron

geboren am 09.12.1990 in Leipzig

Gutachter: Prof. Dr. Bogdan Franczyk, Prof. Dr. Stefan Kirn

Tag der Verleihung: 09.02.2022

Bibliographische Beschreibung

Skowron, Philipp

Methode zur Gestaltung sicherer präskriptiver Systeme für die Therapieunterstützung

Universität Leipzig, Dissertation

181 Seiten, 173 Quellen, 58 Abbildungen, 17 Tabellen, 7 Anlagen

Referat:

Die vorliegende Arbeit befasst sich mit der Gestaltung sicherer präskriptiver Systeme für die Therapieunterstützung. Ziel dabei ist es, den Entwicklungsprozess von der Definition der Ziele, bis hin zur Abwicklung beim Endkunden abzubilden und verschiedene organisatorische, technische, sicherheitskritische und therapeutische Aspekte explizit einzubinden. Dabei lassen sich vorab Probleme und Hindernisse im Entwicklungsprozess abwenden, die möglicherweise ein Scheitern oder eine Inakzeptanz nach sich ziehen würden. Im speziellen Fokus der Methode liegt die explizite Betrachtung und Abbildung der Sicherheit von lernenden und automatisierten entscheidungsunterstützenden Algorithmen, welche eine Therapieunterstützung aktiv fördern. Dies wird mit einer ausdrücklichen Darstellung von sicherheitsrelevanten Anforderungen und deren Integration in alle Phasen des Vorgehensmodells der Methode, in den in dieser Arbeit entwickelten Ansatz, eingebracht. Hierbei spielen nicht nur die technischen und die organisatorischen Absicherungen eine Rolle, sondern ebenso der Brückenschlag zwischen Entwicklung und Domäne, welcher durchgängig im Vorgehensmodell der Methode einen kontinuierlichen Wissenstransfer zur Gewährleistung der Sicherheit und Nützlichkeit des Therapiesystems ermöglicht. Zusätzlich, zu der Wissenskapplung zwischen Entwicklung und Domäne, unterstützt das entwickelte Messsystem zur Risikoabschätzung von präskriptiven Algorithmen die Bewertung von Sicherheitsrisiken, indem es in bestehende Managementmethoden eine prozessuale und bewertbare Risikoabschätzung integriert. Insgesamt stellt die entwickelte Methode mit ihren Komponenten Techniken, Verfahren und Abläufe zur Verfügung, um die Gestaltung von sicheren und therapeutisch zielgerichteten entscheidungsunterstützenden Systemen, unter Einbezug der Zielgruppe, zu ermöglichen.

Vorwort

„Keine Schuld ist dringender, als die, Dank zu sagen.“

– Marcus Tullius Cicero

Wie das Zitat andeutet, soll vorab ein Dank gegenüber denjenigen ausgesprochen werden, die mich bei der Entstehung dieser Arbeit unterstützt und begleitet haben. Somit möchte ich mich an dieser Stelle bei meinem Institut und meinem Doktorvater Prof. Dr.-Ing. Bogdan Franczyk bedanken, die mir die Möglichkeit und Unterstützung gegeben haben, diese Dissertation überhaupt erst anfertigen zu können. Weiterhin danke ich meinen Freunden und Kollegen für die hilfreichen Diskussionen, aufmunternden und unterstützenden Worte und die gewinnbringenden Gespräche, die zur Fertigstellung dieser Arbeit beigetragen haben. Ein besonderer Dank richtet sich an meine Frau, meine Familie und meinen Zimmerkollegen, welche mir immer zu jeder Zeit mit Rat und Tat zu Seite standen.

Inhaltsverzeichnis

Bibliographische Beschreibung	II
Vorwort	III
Inhaltsverzeichnis	IV
Abbildungsverzeichnis	VII
Tabellenverzeichnis	IX
Abkürzungsverzeichnis	X
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielstellung und Forschungsfragen	3
1.3 Methodisches Vorgehen	4
1.4 Struktur der Arbeit.....	5
2 Theoretische Grundlagen	6
2.1 Analysetypen	7
2.1.1 Deskriptive Analysen	7
2.1.2 Prädiktive Analysen.....	7
2.1.3 Präskriptive Analysen.....	8
2.2 Reinforcement Learning	11
2.2.1 Einordnung von Reinforcement Learning	11
2.2.2 Markow-Entscheidungsprozesse	13
2.2.3 Generalisierte Policy-Iteration.....	14
2.2.4 Q-Value Funktionen	15
2.2.5 Exploration vs. Exploitation	16
2.2.6 On-/Off-Policy-Learning	17
2.2.7 Model-free vs. Model-based Learning	17
2.2.8 State-/Action-Spaces	18
2.2.9 Reward Function Design	19
2.2.10 Policy Gradient	20
2.3 Überblick über Sicherheitsrisiken in präskriptiven Algorithmen.....	21
2.4 Entscheidungsunterstützungssysteme.....	24
2.5 Therapeutische Unterstützungssysteme.....	26

Inhaltsverzeichnis

2.5.1	Einordnung & Abgrenzung	27
3	Therapietreue & Adhärenz	28
4	Anforderungsanalyse	35
4.1	Anforderungsanalyse für einen Ansatz.....	35
4.1.1	Allgemeine Anforderungen an den Ansatz	36
4.1.2	Anforderungen an therapieunterstützende Anwendungen	37
4.1.3	Anforderungen an entscheidende Algorithmen.....	38
4.1.4	Anforderungskatalog	39
4.2	Evaluierung existierender Ansätze	40
4.2.1	A self-adaptive distributed decision support model	41
4.2.2	IoT-Based Information System for Healthcare Application.....	42
4.2.3	Discovery and Clinical Decision Support	44
4.2.4	A clinical decision-making mechanism.....	45
4.2.5	Data Analytics Lifecycle	47
4.2.6	ELDAMeth	48
4.2.7	Accelerometer-based step initiation control	49
4.2.8	CRISP-DM	51
4.3	Zusammenfassung der Evaluierung.....	52
5	Methodologische Grundsätze	54
5.1	Methoden Engineering	55
5.2	Software Engineering	56
6	Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme	58
6.1	Überblick Metamodell der Methode.....	59
6.2	Überblick Vorgehensmodell der Methode	61
6.2.1	Analysephase	62
6.2.2	Entwurfsphase	72
6.2.3	Entwicklungs- & Testphase.....	79
6.2.4	Deployment- & Betriebsphase.....	83
6.3	Sub-Prozess Umsetzung Algorithmen.....	86
6.3.1	Analysephase	88
6.3.2	Entwurfsphase	95
6.3.3	Entwicklungs- & Testphase.....	105
6.3.4	Deployment-Phase.....	110
6.4	Überblick Ergebnismodell der Methode.....	111

Inhaltsverzeichnis

6.4.1	Analysephase	113
6.4.2	Entwurfsphase	115
6.4.3	Entwicklungs- & Testphase.....	118
6.4.4	Deployment- & Betriebsphase.....	118
6.5	Überblick Technikmodell der Methode.....	119
6.5.1	Analysephase	121
6.5.2	Entwurfsphase	124
6.5.3	Entwicklungs- & Testphase.....	125
6.5.4	Deployment- & Betriebsphase.....	126
6.6	Überblick Rollenmodell der Methode	127
7	Validierung der Methode.....	131
7.1	Szenario – Mobilitätsunterstützung für geriatrische Patienten.....	131
7.1.1	Schritte Hauptprozess	132
7.1.2	Schritte Subprozess.....	138
7.1.3	Zusammenfassung	147
8	Zusammenfassung & Ausblick.....	147
	Literaturverzeichnis	XII
	Anhang.....	XXIV
	Curriculum Vitae.....	Fehler! Textmarke nicht definiert.
	Selbstständigkeitserklärung	XXXII

Abbildungsverzeichnis

Abbildung 1 Forschungsrahmen nach [Österle et al. 2010]	4
Abbildung 2 Einordnung der Analysetypen	10
Abbildung 3 Übersicht Aufbau Abschnitt Reinforcement Learning.....	11
Abbildung 4 Algorithmenklassen im maschinellen Lernen	12
Abbildung 5 MDP zwischen Agent und Umgebung.....	13
Abbildung 6 GPI Iteration	15
Abbildung 7 Diskreter und kontinuierlicher State / Action Space	18
Abbildung 8 Einordnung Therapieunterstützung gemäß WHO	28
Abbildung 9 Aggregation der Adhärenz-Faktoren-Gruppen.....	30
Abbildung 10 Auswirkung verschiedener Faktoren auf Adhärenz	33
Abbildung 11 Vorgehensmodell nach Zhang et al.	41
Abbildung 12 Vorgehensmodell nach Dziak et al.....	42
Abbildung 13 Vorgehensmodell Yoon et al.	44
Abbildung 14 Vorgehensmodell Forkan/Khalil	45
Abbildung 15 Vorgehensmodell EMC	47
Abbildung 16 Vorgehensmodell Fortino/Russo	48
Abbildung 17 Vorgehensmodell Foglyano et al.....	50
Abbildung 18 Vorgehensmodell Chapman	51
Abbildung 19 Methoden Engineering	55
Abbildung 20 Metamodell der Methode	59
Abbildung 21 Gesamtablauf des Vorgehensmodells.....	61
Abbildung 22 Prozessschritte der "Analysephase" im Gesamtprozess	62
Abbildung 23 Übersicht Aktivitäten für Analysephase.....	63
Abbildung 24 Gliederung Anforderungen.....	67
Abbildung 25 Definition Medizinprodukt.....	68
Abbildung 26 Prozessschritte der "Entwurfsphase" im Gesamtprozess	72
Abbildung 27 Übersicht Aktivitäten für Entwurfsphase	73
Abbildung 28 Prozessschritte der "Entwicklungs- & Testphase" im Gesamtprozess.....	80
Abbildung 29 Übersicht Aktivitäten für Entwicklungs- & Testphase	80
Abbildung 30 Prozessschritte der „Deployment- & Betriebsphase“ im Gesamtprozess	84
Abbildung 31 Übersicht Aktivitäten für Deployment & Betrieb	84
Abbildung 32 Funktionsbaum Teilprozess Algorithmenentwicklung.....	87
Abbildung 33 Prozessschritte der „Analysephase“ im Sub-Prozess	88

Abbildungsverzeichnis

Abbildung 34 Übersicht der Aktivitäten für die Analysephase.....	89
Abbildung 35 Schema Performance- und Sicherheitskennzahlen.....	91
Abbildung 36 Aggregierter Sicherheitsprozess.....	92
Abbildung 37 FMEA Berechnungen Sicherheitsmetriken.....	94
Abbildung 38 Prozessschritte der „Entwurfsphase“ im Sub-Prozess.....	95
Abbildung 39 Übersicht der Aktivitäten für die Entwurfsphase.....	96
Abbildung 40 Enforcement- & Recovery-Strategie ITIL Einordnung.....	99
Abbildung 41 Gesamtmodell Komponentenüberblick.....	103
Abbildung 42 Prozessschritte der „Implementierungs- & Testphase“ im Sub-Prozess....	105
Abbildung 43 Übersicht der Aktivitäten für die Implementierungs- & Testphase.....	106
Abbildung 44 Trainingsphase Agent Vorgehen.....	108
Abbildung 45 Übersicht Testphase Agentensicherheit.....	109
Abbildung 46 Prozessschritt der „Deployment-Phase“ im Sub-Prozess.....	110
Abbildung 47 Übersicht der Aktivitäten für die Deployment-Phase.....	110
Abbildung 48 Übersicht Rollenmodell nach Ebenen.....	127
Abbildung 49 Gesamtvorgehensmodell Szenario Mobilitätsunterstützung.....	132
Abbildung 50 Schematische Darstellung Szenario Mobilitätsunterstützung.....	136
Abbildung 51 Speicherung der Wegeeigenschaften als Graph.....	137
Abbildung 52 Fishbone für Szenario Mobilitätsunterstützung.....	139
Abbildung 53 Reward-Definition für Szenario Mobilitätsunterstützung.....	141
Abbildung 54 Agentennetzwerk für Szenario Mobilitätsunterstützung.....	143
Abbildung 55 Trainingsverlauf mit Hyperparameteroptimierung.....	145
Abbildung 56 FMEA für Szenario Mobilitätsunterstützung.....	145
Abbildung 57 t-SNE für das Wegrate-Verhalten.....	146
Abbildung 58 Soll-FMEA in Szenario Mobilitätsunterstützung.....	146

Tabellenverzeichnis

Tabelle 1 Technische Sicherheitsprobleme	23
Tabelle 2 Anforderungskatalog an die Methode	39
Tabelle 3 Evaluierung Anforderungen Zhang et al.	41
Tabelle 4 Evaluierung Anforderungen Dziak et al.	43
Tabelle 5 Evaluierung Anforderungen Yoon et al.	44
Tabelle 6 Evaluierung Anforderungen Forkan/Khalil.....	46
Tabelle 7 Evaluierung Anforderungen EMC Education Services.....	47
Tabelle 8 Evaluierung Anforderungen Fortino & Russo.....	49
Tabelle 9 Evaluierung Anforderungen Foglyano et al.	50
Tabelle 10 Evaluierung Anforderungen Chapman et al.	52
Tabelle 11 Auswertung der Evaluierung bestehender Ansätze.....	53
Tabelle 12 Übersicht Metamodell-Elemente in einzelnen Phasen	60
Tabelle 13 Ergebnismodell der Methode.....	112
Tabelle 14 Übersicht Technikmodell	119
Tabelle 15 SWOT-Analyse für Szenario Mobilitätsunterstützung ohne Strategien.....	134
Tabelle 16 Sicherheitsmetriken für Szenario Mobilitätsunterstützung	140
Tabelle 17 Testfälle für Szenario Mobilitätsunterstützung	144

Abkürzungsverzeichnis

AI	Artificial Intelligence
AUC	Area under the curve
BDSG	Bundesdatenschutzgesetz
BPMN	Business Process Model and Notation
CRISP-DM	CRoss-Industry Standard Process for Data Mining
DAX	Deutscher Aktien Index
DevOps	Development and IT Operations
DSGVO	Datenschutzgrundverordnung
EUS	Entscheidungsunterstützungssysteme
FMEA	Fehlermöglichkeits- und Einflussanalyse
GPI	Generalized Policy Iteration
IDSS	Intelligent Decision Support Systems
IoT	Internet of Things
ITIL	Information Technology Infrastructure Library
MDP	Markov Decision Process
MSE	Mean Squared Error
RL	Reinforcement Learning
ROC	Receiver-Operating-Characteristic
RPZ	Risikoprioritätszahl
RQ	Research Question
SMART	Spezifisch, messbar, assignable (zuweisbar), realistisch, terminierbar
SWOT	Strength, Weaknesses, Opportunities, Threats
UML	Unified Modeling Language

WHO

World Health Organization

1 Einleitung

Digitalisierung und Medizin sind schon immer eng miteinander verbunden. Behandlungsmöglichkeiten und -qualität steigen mit den technischen Möglichkeiten an, die durch die Digitalisierung geboten werden. Dies zeigt sich u. a. in den Marktpotentialen für digitale Lösungen im Medizinbereich. Gerade die Bereiche Wireless Health und mHealth sollen mit zusammen 31 Mrd. USD Marktanteil (im Jahr 2020) das Monitoring und die medizinische und therapeutische Begleitung von Patienten zukünftig vereinfachen (vgl. [Bitkom e. V. 2018]). Hinzu kommen, für die Unterstützung der Ausbreitung der Bereiche mHealth und Wireless Health, Anwendungen der künstlichen Intelligenz, die bis 2025 ein Marktvolumen von 36,1 Mrd. US-Dollar dediziert erzielen werden (vgl. [Handelsblatt et al. 2019]). Hierfür bedient sich die Medizin aus dem Bereich der IoT-Lösungen, wobei Sensoren und Aktoren genutzt werden, um den Patienten und das Umfeld des Patienten zu überwachen und zu beeinflussen. Gerade Lösungen, die ihre Schlussfolgerungen und Handlungen nicht direkt vom Menschen vorgegeben bekommen, sondern diese selbstständig treffen, haben immenses Potential die Art und Weise, wie das Gesundheitswesen seine Leistungen erbringt, zu verändern (vgl. [Gottesman et al. 2019, 16ff.], [Yom-Tov et al. 2017, 338ff.], [Raghu et al. 2017, 1ff.] & [Hegner et al. 2018, 5]). Hieraus entstehen jedoch für die Entwickler solcher Lösungen sowohl technische als auch organisatorische Problemstellungen, wie der Einbezug der Entwicklung von Algorithmen in bestehende Vorgehensmodelle und die Gewährleistung der Sicherheit dieser, welche Beachtung finden müssen.

Ziel der Forschungsarbeit ist es folglich, eine Methode für die Entwicklung von therapieunterstützten Entscheidungssystemen bereitzustellen, sodass bei der Entwicklung solcher Systeme alle relevanten Aspekte betrachtet, einbezogen und umgesetzt werden sowie die Sicherheit beim Einsatz dieser Systeme für den Patienten gewährleistet werden kann.

1.1 Motivation und Problemstellung

Das Internet of Things (IoT) prägt die aktuelle technische Entwicklung im Gesundheitswesen deutlich. Neuartige Lösungen unterstützen die medizinischen Fachkräfte bei der Ausübung ihrer Tätigkeiten oder übernehmen diese teilweise komplett. Neben der allgemeinen Steigerung der Qualität unseres Gesundheitssystems, dienen diese Lösungen vor allem der Unterstützung und Entlastung des medizinischen Personals. Wie in vielen anderen Branchen, wird das Gesundheitswesen ebenfalls durch einen Fachkräftemangel begleitet

1 Einleitung

(vgl. [Rothgang et al. 2012, 37ff.]). Gerade im Bereich der Pflege ist dies deutlich zu erkennen. Zu unattraktiv, zu altmodisch, zu unflexibel wird gerade der berufliche Zweig in der Medizin genannt, welcher neben den Ärzten maßgeblich für das Wohlbefinden der Patienten und für die Umsetzung des gesetzlich zugesicherten medizinischen Grundversorgung zuständig ist (vgl. [Eggert et al. 2019, 21f.]). Hierunter leiden die bereits im Berufszweig etablierten Fachkräfte, die über psychische, physische und zeitliche Überlastung klagen (vgl. [DAK 2015]). Gerade in diesem Bereich des Gesundheitswesens profitieren die Fachkräfte von entlastenden, technischen Maßnahmen, welche einerseits den Beruf attraktiver gestalten und gleichzeitig die Qualität der Versorgung verbessern (vgl. [Initiative Neue Qualität der Arbeit (BAuA) 2018, 34]). Diese Lösungen können den Patienten unterstützen oder Entscheidungen über die Unterstützung des Patienten fällen, ohne das unmittelbar das medizinische Personal vor Ort sein muss (vgl. [Initiative Neue Qualität der Arbeit (BAuA) 2018, 18], [Elmer et al. 2019, 77ff.]). Solche technischen Entwicklungen zur Entlastung und Entscheidungsübernahme fallen unter die entscheidungsunterstützenden Lösungen.

Damit diese aus Politik und Gesellschaft geborene Vision flächendeckend umgesetzt werden kann, müssen die zu entwickelnden unterstützenden Systeme sowohl praxistauglich als auch sicher im Umgang mit dem Patientenwohl sein (vgl. [Hassan et al. 2018, 303ff.] & [Brundage et al. 2018, 18ff.]). Dies dient nicht nur der rechtlichen Absicherung, sondern auch der gesellschaftlichen Akzeptanz dieser Lösungen und deckt sich mit den Grundprinzipien der Datenethikkommission der Bundesregierung über den Einsatz von künstlicher Intelligenz (vgl. [Datenethikkommission der Bundesregierung (Hrsg.) 2019, 39ff.]). Gleichwohl müssen solche Systeme die individuellen Ansprüche des Patienten sowie deren Rahmenbedingungen mit einbeziehen, was wiederum der Sicherheit des Patientenwohls als auch der tatsächlichen Unterstützung im Alltag, bei der Versorgung der Patienten, dient. Ebenso begünstigt der Einbezug von therapeutischen Rahmenbedingungen die Steigerung der Akzeptanz solcher Systeme, erhöht die Qualität und Wirksamkeit des Systems und reduziert die Mehrfachentwicklung von Lösungen gleichen Einsatzgebietes aufgrund Unwirksamkeit bzw. Inakzeptanz durch das Fachpersonal und den Patienten.

Mithilfe dem Einsatz von entscheidungsunterstützenden Therapiesystemen, können die oben genannten Punkte adressiert, die Qualität der Pflege gesichert und sowohl die Entlastung des Pflegepersonals als auch eine individuelle Betreuung der Patienten nachhaltig unterstützt werden.

1 Einleitung

Hierzu ist eine explizite Betrachtung der Sicherheit der Entscheidungen bei therapieunterstützenden Systemen nötig, um das Wohl der Patienten nicht zu gefährden. Diese wird in bereits etablierten Prozessen und Methoden in der Softwareentwicklung nur ungenügend abgebildet. Hieraus ergibt sich der Bedarf zur Entwicklung einer Methode, für die Entwicklung von entscheidungsunterstützenden Therapiesystemen, welche sowohl therapeutische und Personen-individuelle Charakteristika als auch die explizite Betrachtung der Sicherheit von angewendeten präskriptiven Verfahren in den Prozess einbezieht.

1.2 Zielstellung und Forschungsfragen

Zur hinreichenden Untersuchung der im vorherigen Abschnitt genannten Problemstellung, wird hierzu folgende übergeordnete Forschungsfrage definiert:

Wie muss eine Methode gestaltet sein, um ein Anwendungssystem für die therapeutische Unterstützung zu entwickeln, welches die Sicherheit der Entscheidungen der Algorithmen bei der Therapieunterstützung explizit unterstützt?

Zur Bearbeitung der übergeordneten Forschungsfrage wird diese in folgende Unterforschungsfragen (englisch Research Questions, kurz RQ) unterteilt:

RQ1: Welche Ansätze für die sichere Gestaltung von Algorithmen in entscheidungsunterstützenden Therapieunterstützungssystemen existieren bereits?

RQ2: Wie müssen die Anforderungen an eine Methode gestaltet sein, damit diese sichere Entscheidungsfindungen für Algorithmen bei therapeutischen Unterstützungssystemen gewährleisten?

RQ3: Welche Optimierungen lassen sich für die Sicherheit in therapeutischen Unterstützungssystemen erzielen?

Eine explizite Abgrenzung der Therapieunterstützung von anderen therapeutischen Systemen findet im Teil der theoretischen Grundlagen unter dem Unterabschnitt 2.5.1 statt.

Zusammengefasst, ist es das Ziel dieser Arbeit, Algorithmen, die für entscheidungsunterstützende Therapiesysteme genutzt werden, sicherer zu gestalten, indem eine Methode unter der expliziten Betrachtung der Sicherheit derartiger Algorithmen entwickelt wird. Diese Methode, soll Entwicklern derartiger Systeme als Leitlinie für die Betrachtungen der Anforderungen an die Methode dienen, wodurch nachhaltig, und bereits

1 Einleitung

in der Design-Phase des Therapiesystems, die Sicherheit der Algorithmen für die Entscheidungsunterstützung gewährleistet werden kann.

Somit minimiert man nicht nur die Risiken, für entscheidungsunterstützende Algorithmen in Therapiesystemen, sondern trägt auch zum weiteren Abbau von Sicherheitsbedenken in der Branche des Gesundheitswesens und der Pflege bei, indem man entsprechende Maßnahmen aus der Methode vorhalten kann. Dies fördert damit auch, dass solche Systeme vermehrt Einzug in die Branche halten, um Angestellte zu unterstützen und die Qualität der Dienstleistungen an und für den Patienten zu verbessern.

1.3 Methodisches Vorgehen

In diesem Abschnitt sollen der für die Arbeit und die wissenschaftlichen Fragestellungen herangezogene Forschungsrahmen sowie die angewandten Methoden erläutert werden. Der verwendete Forschungsrahmen folgt den Grundsätzen der gestaltungsorientierten Wirtschaftsinformatik gemäß [Österle et al. 2010].

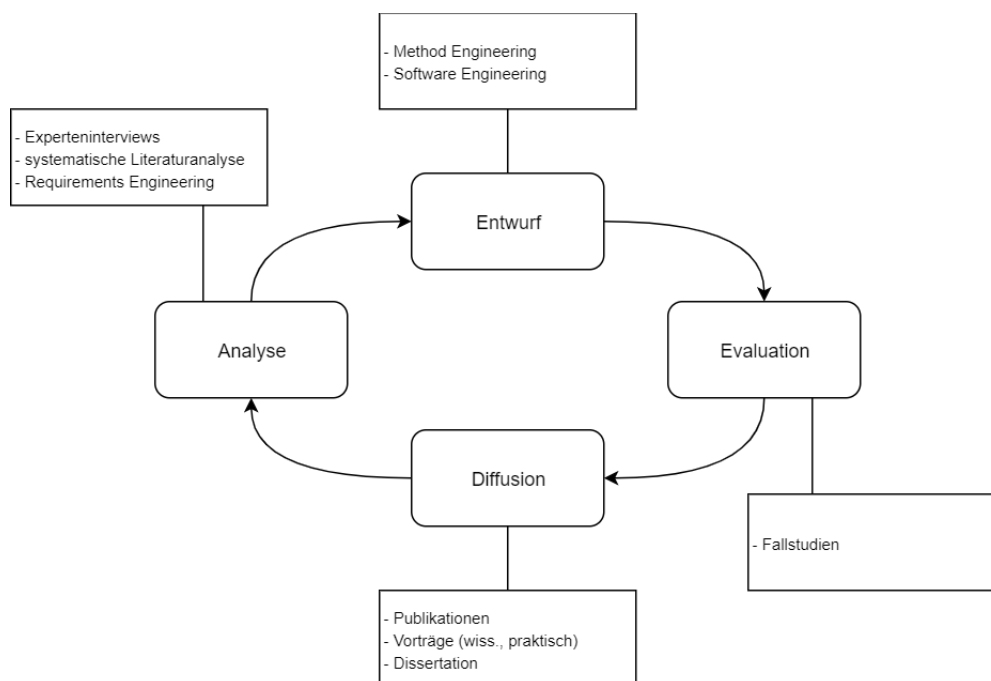


Abbildung 1 Forschungsrahmen nach [Österle et al. 2010]

Wie in Abbildung 1 zu erkennen, besteht der iterative Prozess des Forschungsrahmens aus den vier grundlegenden Schritten: der Analyse, des Entwurfs, der Evaluation und der Diffusion. In jedem Schritt dieses Rahmens werden verschiedene Forschungsmethoden genutzt, um die angestrebten Forschungsziele zu erreichen.

1 Einleitung

Demnach werden in der Analysephase die Problemstellung beschrieben, die zu lösenden Forschungsfragen definiert sowie die möglichen Ansatzpunkte zur Lösung des Problems erörtert. Hierfür wurden verschiedene Methoden eingesetzt. Zum einen wurde eine systematische Literaturanalyse nach [Vom Brocke et al. 2009] durchgeführt, um den aktuellen wissenschaftlichen Stand der Problemstellung festzuhalten. Zum anderen nach [Schnell et al. 1995, 299ff.] wurden Experteninterviews und Workshops durchgeführt, um praxisbezogene Herausforderungen und Problemstellungen mit in die Anforderungsanalyse aufzunehmen. Mit dem Requirements Engineering (vgl. Absatz 6.2.1.3 & [Rupp et al. 2009, 94ff.]) wurden die Anforderungen ermittelt und in einem Anforderungskatalog zusammengefasst (vgl. Unterabschnitt 4.1.4). In der Entwurfsphase werden die angestrebten Artefakte methodisch hergeleitet. Hierfür wurden vor allem das Method Engineering und das Software Engineering als Methoden genutzt, die in Kapitel 5 ausführlicher beschrieben werden. Die Phase der Evaluation beschreibt die Überprüfung der geschaffenen Artefakte gegen die definierten Ziele des Vorhabens. Hierzu wurden Fallstudien durchgeführt, welche die Artefakte in praktisch angelegten Studien anwenden (vgl. Kapitel 7). Im Rahmen der Diffusion werden die Ergebnisse der wissenschaftlichen Basis durch Publikationen hinzugefügt (vgl. [Österle et al. 2010, 12f.]).

Alle in der Arbeit entstandenen Artefakte sollen den Prinzipien (abstrakt, originell, begründet und nützlich) der gestaltungsorientierten Wirtschaftsinformatik entsprechen (vgl. [Österle et al. 2010, 13f.]).

Im nachfolgenden Abschnitt wird nach der methodologischen Struktur, die inhaltliche Struktur der vorliegenden Arbeit beschrieben.

1.4 Struktur der Arbeit

Dieser Abschnitt soll einen Überblick über die Struktur der vorliegenden Arbeit vermitteln.

Das erste Kapitel bietet einen Überblick sowie eine Einführung über die zu bearbeitende Problemstellung. Dabei wird auf die auslösenden Faktoren des Problems eingegangen und welche Konsequenzen sich aus diesen ergeben. Basierend auf der beschriebenen Problemstellung wird zusammen mit den zu klärenden Forschungsfragen die Zielstellung dieser Arbeit beschrieben. Das methodische Vorgehen soll dabei die wissenschaftliche Basis beschreiben, wie das geplante Ziel, die Entwicklung einer Methode zur Gestaltung von Anwendungssystemen für die therapeutische Unterstützung, erreicht werden soll.

2 Theoretische Grundlagen

Im Kapitel 2 der theoretischen Grundlagen werden dem Leser die fachlichen Grundlagen zum Verständnis der vorliegenden Arbeit vermittelt. Dabei wird allgemein auf Arten möglicher Analysen eingegangen und dies weiterführend auf die konkrete Ausprägung des bestärkenden Lernens (engl. Reinforcement Learning) ausgeweitet. Weiterhin werden neben der Funktionalität des Reinforcement Learnings auch die aktuell bestehenden Sicherheitsrisiken und deren Charakteristika beim Einsatz vom bestärkenden Lernen beschrieben. Folgend auf die Beschreibung der algorithmischen Grundlagen, werden zusätzlich fachliche Kenntnisse über die Therapieunterstützung, deren Einordnung nach der WHO, und über Therapietreue und Therapieunterstützungssysteme geschaffen.

In dem Kapitel 4 der Anforderungsanalyse werden die relevanten Anforderungen bei der Lösung des beschriebenen Problems definiert und ein Anforderungskatalog gebildet. Daraufhin werden die identifizierten bestehenden Ansätze und Verfahren, anhand dieses Anforderungskatalogs, auf ihre Vollständigkeit hin überprüft.

Das darauffolgende Kapitel 5 der methodischen Grundsätze, beschreibt noch einmal detailliert die wichtigsten methodischen Vorgehen bei der Durchführung.

Im fünften Kapitel stellt die geplante Lösung für die Problemstellung vor. In diesem werden die benötigten Elemente (Meta-, Vorgehens-, Ergebnis-, Technik- und Rollenmodell) detailliert beschrieben. Dabei basiert die Lösung auf den fachlichen und methodischen Kapiteln sowie auf der Problemstellung und dessen abgeleiteten Anforderungskatalog.

Im Kapitel 7 findet die Evaluation der Methode statt. Hierfür wird ein Routing-System für die Unterstützung der Bewegungsförderung älterer Personen auf Basis der zu erstellenden Methode entwickelt und angewendet.

Zum Abschluss wird im Kapitel 8 eine Zusammenfassung und Übersicht über den wissenschaftlichen Beitrag der vorliegenden Arbeit gegeben sowie ein Ausblick auf den noch vorhandenen Forschungs- und Entwicklungsbedarf in diesem Themenbereich geworfen.

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen für die vorliegende Arbeit geschaffen. Dabei wird auf verschiedene Analysetypen eingegangen, ein Hauptvertreter der entscheidungsunterstützenden Algorithmen näher beschrieben und eine Übersicht über die Sicherheitsrisiken beim Einsatz dieser Algorithmenfamilie dargestellt. Weiterhin werden

2 Theoretische Grundlagen

Entscheidungsunterstützungssysteme sowie die Therapieunterstützung und die daraus folgenden Therapieunterstützungssysteme eingeordnet und beschrieben. Weiterhin wird ein Abriss über die Definition und inhaltliche Beschreibung des Begriffes Adhärenz gegeben.

2.1 Analysetypen

Nummerische Analysen können in verschiedene Arten von Analysetypen bzw. Analysekatoren unterteilt werden. Bei der Kategorisierung dieser Typen stehen der temporale und der wertorientierte Bezug im Vordergrund, welche die Typen im Wesentlichen unterscheiden. Hierbei handelt es sich um deskriptive, prädiktive und präskriptive Analysen.

2.1.1 Deskriptive Analysen

Analytische Verfahren, deren Hauptaugenmerk in der Auswertung der Vergangenheit liegen, zählen zu den deskriptiven Analysen. Diese Analysen nutzen Daten aus der Vergangenheit, um durch geeignete Aggregationsmethoden wie Kategorisierung, Charakterisierung, Konsolidierung oder Klassifizierung, Informationen aus diesen Daten zu gewinnen. Dabei werden i. d. R. Maße wie Häufigkeiten, Varianzen, Mittelwerte etc. genutzt, um Statistiken, z. B. in Form von Reports zum Monitoring von Software, Prozessen oder Unternehmen, zu erstellen. Zusätzlich zu diesen einfacheren Verfahren zur Erstellung von z. B. Häufigkeiten, fallen unter die deskriptiven Analysen ebenso Musteranalysen, die unter anderem mit Cluster- oder Ausreißeranalysen angewandt werden können. Auch hierbei werden Daten aus der Vergangenheit nach Mustern durchsucht, um Aufschluss über Sachverhalte in der Vergangenheit oder der Gegenwart zu erhalten (vgl. [Seiter 2017; Lustig et al. 2010; Evans/Lindner 2012]).

Mit deskriptiven Analysen lassen sich somit Fragen wie beispielsweise „Wie waren unsere Verkaufszahlen im letzten Monat?“ oder „Welche Produkte wurden häufig zusammen verkauft?“ beantworten.

2.1.2 Prädiktive Analysen

Besteht ein Interesse daran herauszufinden, wie sich manche Zustände in der Zukunft weiterentwickeln und werden dafür mathematische Modelle genutzt, spricht man in diesem Fall von prädiktiven Analysen. Prädiktive Analysen nutzen, ebenso wie deskriptive Analysen, angefallene Daten aus der Vergangenheit, um das Eintreffen gewisser Zustände in der Zukunft vorherzusagen. Diese Vorhersagen können sowohl deterministisch als auch nicht-deterministisch sein. Der Unterschied hierbei liegt in der

2 Theoretische Grundlagen

Wahrscheinlichkeitsverteilung. Deterministische Verfahren können z. B. vordefinierte Prozesse sein, deren klar definierte Abfolge in einer kontrollierten Welt unter bestimmten Vorbedingungen nur einen gewissen Folgezustand zulassen. Nicht-deterministische Abfolgen hingegen basieren auf stochastischen Annahmen, wodurch der zukünftige Zustand nur anhand von Wahrscheinlichkeiten vorhergesagt werden kann (vgl. [Seiter 2017; Lustig et al. 2010; Evans/Lindner 2012]).

Sowohl deterministische als auch nicht-deterministische Verfahren können in zwei verschiedenen Arten der Vorhersage, Klassifikation und Regression, unterteilt werden. Während Regressionen versuchen, unter Eingabe von bestimmten Faktoren in das mathematische Modell, numerische Werte in zukünftigen Zuständen zu definieren, dient die Klassifikation der Ermittlung einer Einordnung eines Attributes in der Zukunft in eine definierte Menge an Klassenattributen (vgl. [Goodfellow et al. 2016]).

Wie auch in der deskriptiven Analyse werden in den prädiktiven Analysen (versteckte) Muster in den vorhandenen Daten ermittelt. Der Unterschied jedoch ist, dass diese Muster auf zukünftige Zustände übertragen werden, um den Ausgang zukünftiger Zustände vorherzusagen (vgl. [Goodfellow et al. 2016]).

Mit prädiktiven Analysen lassen sich Fragen beantworten wie „Welchen Wert hat der DAX übermorgen?“ oder wie „Wie wird das Wetter morgen?“.

2.1.3 Präskriptive Analysen

Prädiktive Aussagen über Zustände implizieren lediglich Wahrscheinlichkeiten, mit denen ein gewisser Zustand zutrifft bzw. zutreffen wird. Ob die mit der Information beschlossenen Entscheidungen über das nächste Handeln in der Zukunft einen optimalen Beitrag zum Erreichen von definierten Zielen liefern, wird nicht betrachtet. Werden Aktionsfolgen in der Zukunft mithilfe von Informationen aus der Vergangenheit und Gegenwart getroffen, um ein optimales Erreichen von vordefinierten Zielgrößen zu erreichen, kann man von präskriptiven Analysen sprechen (vgl. [Gröger et al. 2014; Lustig et al. 2010]).

In der Literatur wurde der Begriff präskriptiv bisher nicht einheitlich definiert und findet sich in ähnlichen Definitionen wieder, welche jedoch teilweise unterschiedliche Eigenschaften aufweisen. So wurde bereits 1987 in [Stewart/McMillan 1987] der Begriff der präskriptiven Analyse beschrieben. Dabei wird auf den Unterschied zu Wissensbasen eingegangen, welche mithilfe von Expertenwissen fest definierte Handlungsfolgen aufweisen. Präskriptive Analysen hingegen besitzen die Eigenschaft,

2 Theoretische Grundlagen

Handlungsempfehlungen zu geben, ohne dabei explizites Expertenwissen abzurufen. Grundsätzlich werden bei [Stewart/McMillan 1987] die Eigenschaften definiert, dass präskriptive Analysen Handlungsempfehlungen zur optimalen Erreichung eines zukünftigen Zieles generieren, wodurch diese sowohl auf deterministischen, als auch auf nicht-deterministischen Grundlagen beruhen können. Weitere Definitionsquellen der präskriptiven Analysen erwähnen den impliziten Einbezug von Unsicherheit in der Exploration zukünftiger Ereignisse, wenn die Annahmen über den Ausgang der zukünftigen Zustände auf statistischen Verteilungen basieren. Somit werden präskriptive Analysen den Definitionen zur Folge den stochastischen Optimierungsverfahren zugeordnet (vgl. [Lustig et al. 2010; Evans/Lindner 2012]). Dabei bezieht sich die Optimierung auf eine oder mehrere Variablen, die in der Zukunft maximiert oder, je nach Fragestellung, auch minimiert werden können (vgl. [Evans/Lindner 2012]). Alle Definitionen beschreiben hierbei die Zukunftsorientierung der Analysen als fundamentale Eigenschaft.

Führt man die Definition-Varianten zusammen, ergibt sich, unter Berücksichtigung aktueller wissenschaftlicher und technologischer Entwicklungen, die folgende Begriffsbestimmung, die für die nachstehenden Betrachtungen als Grunddefinition herangezogen werden soll:

Präskriptive Analysen berechnen auf Basis von stochastischen Verteilungen (nicht-deterministisch) und somit unter Einbezug von imperfekten Informationsflüssen (explorative Unsicherheit) Aktionen zur Erreichung zukünftiger Zustände (Zukunftsorientierung), um einen Zielzustand zu erreichen, in dem eine oder mehrere Variablen minimiert bzw. maximiert sind, damit diese in einem (nah-)optimalen Verhältnis zueinander stehen (Optimierungsverfahren).

Mit präskriptiven Analysen lassen sich somit unter Einbezug der oben genannten Definition Fragen beantworten „Wie plane ich meine Routen in Abhängigkeit von Ladung, Anzahl LKWs und Fracht, um möglichst kosteneffizient zu sein?“ oder „Wie muss eine Ampelschaltung adaptiert werden, um eine minimale Wartezeit pro Verkehrsteilnehmer zu erreichen?“.

Vergleicht man die drei Analysetypen miteinander, können diese nach verschiedenen Kriterien unterteilt betrachtet werden. Die Autoren [Mauerer 2018; Puget 2015] definieren unabhängig voneinander Grafiken, zur Gegenüberstellung der drei Analysetypen, basierend auf den zu betrachtenden Kriterien, wie es in der Abbildung 2 dargestellt wird.

2 Theoretische Grundlagen

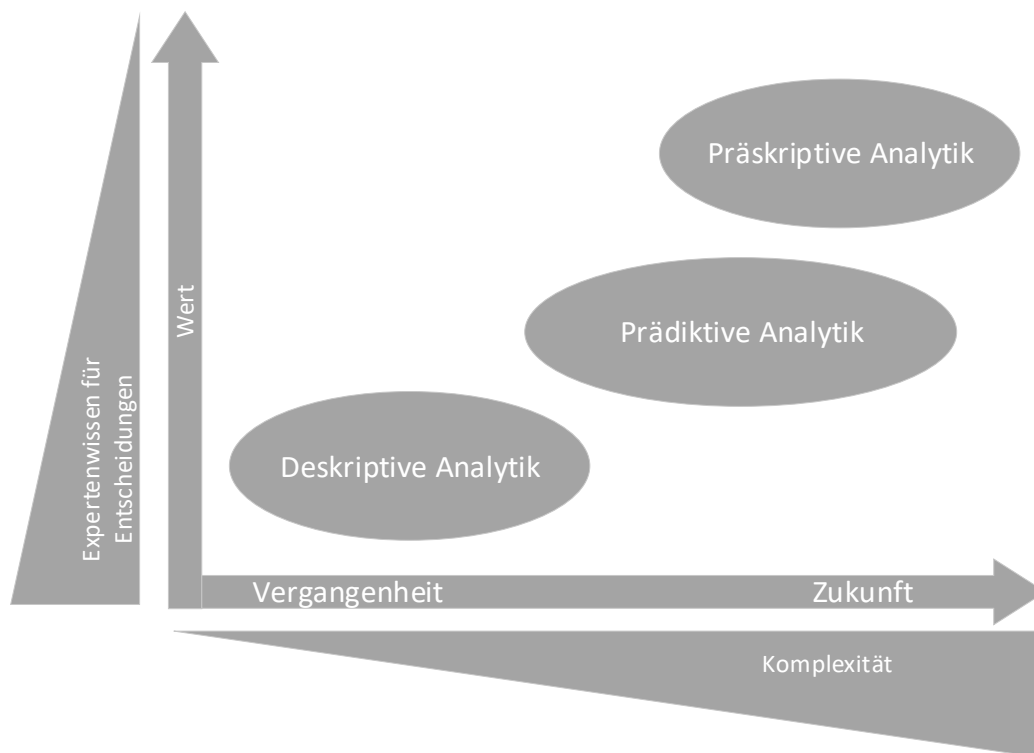


Abbildung 2 Einordnung der Analysetypen nach Wert der Information, temporalen Bezug, Komplexität der Anwendung und benötigten Expertenwissen, welches zur Entscheidungsfindung von Experten eingebracht werden muss (angelehnt an [Puget 2015; Mauerer 2018])

Als Kriterien werden sowohl der temporale Bezug und die Komplexität der Entscheidungsfindung als auch die Wertigkeit der generierten Information und damit einhergehend auch die Relevanz von menschlichem Expertenwissen für die Interpretation der Informationen mit einbezogen. Wie aus der Abbildung 2 zu erkennen ist, sortieren sich die drei Analysetypen von links unten nach rechts oben ein. Neben dem steigenden Zukunftsbezug, der bereits teilweise in den vorherigen Abschnitten beschrieben wurde, ist die steigende Komplexität ein weiteres Merkmal, welches die Analysetypen voneinander abgrenzt. Unter der Komplexität ist u. a. der Einfluss der Analysen auf z. B. geschäftliche und organisatorische Prozesse (Prozess-Neustrukturierungen, Entwicklung neuer Prozesse etc.), Ressourcenaufwendungen auf technischer Ebene (Rechenleistung, Datensammlung und -bereitstellung etc.) sowie die Verfügbarkeit von qualifizierten Fachpersonal (Data Analysts, Machine Learning Engineers etc.) zu verstehen (vgl. [Puget 2015; Mauerer 2018; Hagerty 2016]). Der Wert der Informationen korreliert mit dem Einsatz von Expertenwissen. Je informationsreicher die Analysen werden, desto mehr Wert besitzen die generierten Informationen für ein Unternehmen und desto weniger Expertenwissen wird für die Interpretation der Informationen benötigt (vgl. [Stewart/McMillan 1987, 314ff.], [Puget 2015]).

2 Theoretische Grundlagen

Präskriptive Analysen sind somit nach ihren Merkmalen betrachtet die Analysetypen, die den höchsten Mehrwert für Unternehmen, unter Einbezug hoher Komplexität, generieren.

2.2 Reinforcement Learning

In den vergangenen Jahren kam es mit den neuen Errungenschaften im Rahmen des Machine Learning, deutsch *maschinelles Lernen*, insbesondere durch den vermehrten Einsatz von Deep Learning-Technologien, zunehmend zu einem erhöhten Forschungs- und Entwicklungsdrang in die Richtung autonomer, selbstlernender und selbstentscheidender Systeme. Allen voran wurde hierbei der Einsatz von Reinforcement Learning (RL), deutsch *bestärkendes Lernen*, gefördert, um Optimierungsprobleme auf Problemstellungen anzuwenden, wie es vorher, aufgrund der fehlenden Techniken und Rechenleistungen, noch nicht möglich war.

Dieser Abschnitt soll sich der Definition, Abgrenzung und Erläuterung des RL widmen. Um eine Übersicht über die kommenden Unterabschnitte zu geben, soll folgende Abbildung 3 zur Orientierung dienen.

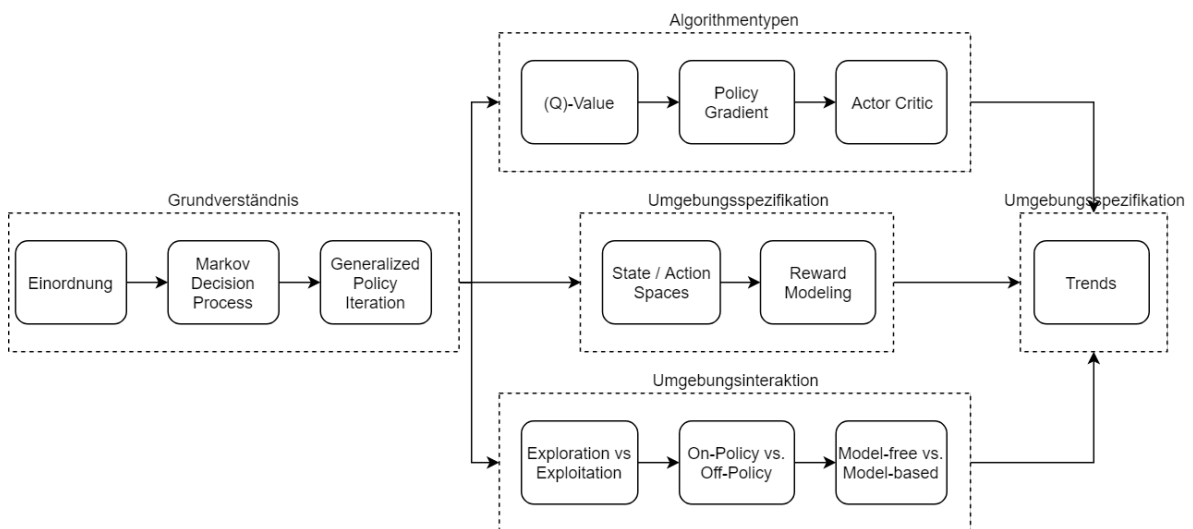


Abbildung 3 Übersicht Aufbau Abschnitt Reinforcement Learning

2.2.1 Einordnung von Reinforcement Learning

Um eine Abgrenzung zu anderen Algorithmenklassen im Bereich des maschinellen Lernens gegenüber RL zu erreichen, kann man u. a. den angewendeten Analysetyp (siehe dazu 2.1) und das Lernverhalten betrachten.

2 Theoretische Grundlagen

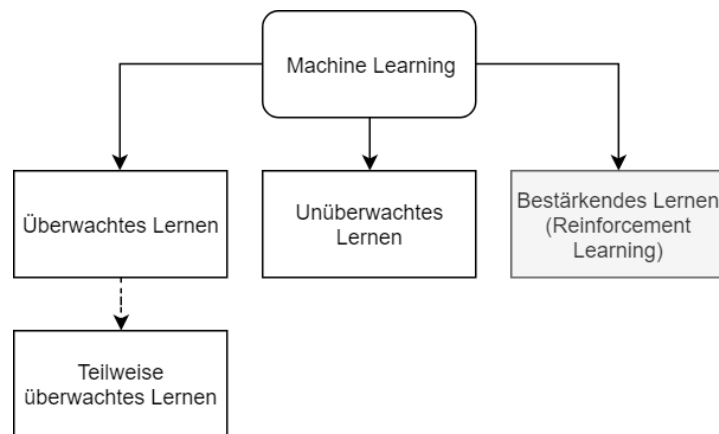


Abbildung 4 Algorithmenklassen im maschinellen Lernen

Abbildung 4 illustriert eine mögliche Einteilung der Algorithmenklassen im maschinellen Lernen, welche sich hauptsächlich herauskristallisiert haben.

Das überwachte Lernen nutzt eine Kombination aus Daten und deren entsprechenden Bezeichnungen, zu Englisch *labels*, um eine Relation zwischen den Daten und dem Label zu erlernen. Als überwacht wird dieses Lernverhalten deswegen bezeichnet, weil zu jedem Datensatz klar definierte und abgegrenzte Labels existieren. In der Sonderform des teilweise überwachten Lernens, werden zu den Daten mit konkreten Labels weitere, nicht gelabelte Daten hinzugefügt, um das angestrebte prädiktive Ergebnis zu verbessern. Wie bereits erwähnt, wird das überwachte Lernen vor allem für prädiktive Analysen (Regression, Klassifikation etc.) eingesetzt (vgl. [Sutton/Barto 2018, 2] & [Pumperla et al. 2019, 12ff.]).

Im Gegensatz zum überwachten Lernen, kommt das unüberwachte Lernen ohne Labels für die Daten aus. Hier wird nicht versucht, eine Relation zwischen den Daten und den Labels zu erstellen, sondern zwischen den Daten selbst. Dafür ändert sich auch der Einsatz der hierin verorteten Algorithmen. Diese sind vor allem deskriptiver Natur und werden bspw. für das Finden von Clustern in Daten genutzt (vgl. [Sutton/Barto 2018, 2] & [Pumperla et al. 2019, 12ff.]).

Das RL bildet eine dritte Klasse von Algorithmen, die sich sowohl im Einsatz und in der Art des Lernens als auch im Analysetyp von den bereits beschriebenen zwei Algorithmenklassen unterscheidet. RL zielt auf präskriptive Analyse ab und wird vor allem bei Optimierungsproblemen genutzt. Dabei beschreibt ein Optimierungsproblem im Kontext des RL das Maximieren oder Minimieren eine Zielfunktion unter den gegebenen Umgebungsbedingungen, die durch die Auswahl einer Reihe von (optimalen) Aktionen erreicht wird (vgl. [François-Lavet et al. 2018, 219f.], [Sutton/Barto 2018, 9ff.]).

2 Theoretische Grundlagen

Weiterhin unterscheidet sich RL in der Grundart des Lernverhaltens, wie eine Konvergenz zu einem Optimum erreicht werden kann. Dabei werden keine expliziten Labels auf bestimmte Datensätze genutzt, sondern Beobachtungen aus einem Optimierungsproblem und damit zusammenhängend ein Signal, welches die präskriptive Entscheidung des RL-Algorithmus während der Beobachtung bewertet. Durch das Signal erhält der Algorithmus Feedback über seine Entscheidung in der zugrundeliegenden Beobachtung und kann mit dafür entwickelten Verfahren in seiner Entscheidungsfindung optimiert werden (vgl. [Sutton/Barto 2018, 2], [François-Lavet et al. 2018, 225ff.] & [Pumperla et al. 2019, 12ff.]).

Im Folgenden wird das RL näher erläutert und beschrieben, um die Funktion und das Lernverhalten verstehen zu können.

2.2.2 Markow-Entscheidungsprozesse

Mathematische Grundlage und Abstraktion für die Art und Weise, wie RL-Algorithmen lernen und vorgehen, liefern die Markow-Entscheidungsprozesse, zu Englisch *Markov decision processes*. Im weiteren Verlauf soll die Abkürzung MDP, in Anlehnung an die englische Bedeutung, genutzt werden.

Zum weiteren Verständnis müssen zwei Begriffe eingeführt werden. Der Entscheidungsfinder ist der sogenannte *Agent*, die Interaktionsfläche des Agenten wird als *Umgebung* bezeichnet (vgl. [Sutton/Barto 2018, 6f.]).

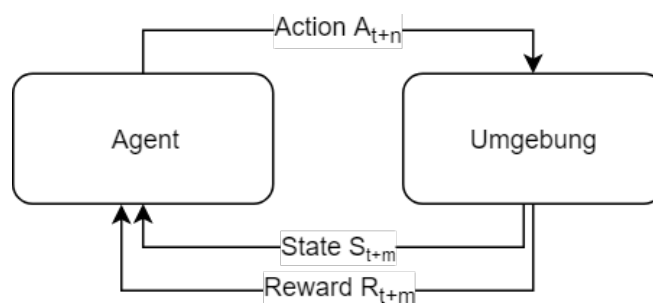


Abbildung 5 MDP zwischen Agent und Umgebung (in Anlehnung an [Sutton/Barto 2018, 47ff.])

Wie aus Abbildung 5 zu erkennen ist emittiert der Agent zu einem bestimmten Zeitpunkt eine Entscheidung, eine sogenannte *Action*, auf der Basis einer Beobachtung, dem *State*, und einem Feedback-Signal, dem *Reward*. Unter Anwendung der Action innerhalb der Umgebung wird der State zu einem neuen, zukünftigen Zeitpunkt verändert und ein neuer State gebildet. Gleichwohl wird auf Basis der Änderung und der emittierten Action ein neuer Reward als Feedback für die Qualität der Action des Agenten an den Agenten weitergegeben. In Bezug auf einen MDP hat jedes Tupel aus State und Reward eine ganz

2 Theoretische Grundlagen

bestimmte Wahrscheinlichkeit aufzutreten und ist ausschließlich vom vorausgehenden State und der Action abhängig. Somit kann unter Einbezug einer Action und eines State eine exakte Übergangswahrscheinlichkeit zwischen zwei States definiert werden. Dies ermöglicht es, eine Funktion zu definieren bzw. zu erlernen, die eine bestimmte zu erreichende Zielgröße maximiert, indem sie Tupel aus States, Actions und Rewards einbezieht. MDP sind somit zusammengefasst eine mathematische Abstraktion für eine zielorientierte Optimierung eines Problems durch Interaktionen (vgl. [François-Lavet et al. 2018, 225ff.] & [Sutton/Barto 2018, 47ff.]).

Diese mathematische Grundlage von MDP bildet den Ausgangspunkt und das mathematische Fundament für die Definition und das Verhalten von RL-Algorithmen. Im weiteren Verlauf wird nicht nochmals auf MDP eingegangen, da diese nur als mathematischer Hintergrund dienen.

2.2.3 Generalisierte Policy-Iteration

Um einen Bestwert in Optimierungsproblemen, definiert auf Grundlage der MDP, zu finden, ist das Einführen zweier Begriffe und eines Prozesses nötig, welche im Folgenden erläutert werden sollen. Dabei soll ein grundlegendes Verständnis für den allgemeinen Ablauf des Optimierungsprozesses geschaffen werden, das zum Verständnis der Ausführungen in dieser Arbeit ausreicht. Für tieferegehende Analysen und Charakterisierungen der Ansätze wird an gegebenen Stellen auf die zugrundeliegende Literatur verwiesen.

Jeder State in einem MDP kann, unter Berücksichtigung des zu erreichenden Ziels der Optimierung, einen bestimmten Wert annehmen. Dieser Wert gibt Aufschluss darüber, wie gut es für einen bestimmten Agenten ist, in diesem State zu sein und wie hoch sein kumulativer Reward, ausgehend von diesem State, bis zum Optimierungsziel sein wird. Diese sogenannten *Value-Funktionen* dienen als Evaluation und Bewertung aller States, die von einem Agenten gesehen wurden (vgl. [Sutton/Barto 2018, Va6]).

Um States in einer Umgebung überhaupt zu erreichen, bedient sich der Agent einer sogenannten *Policy*. Die Policy eines RL-Agenten beschreibt dessen Verhalten in der Umgebung. Somit ist eine Policy eine Funktion, die eine Verbindung zwischen einem aktuellen State und einer auszuwählenden Action beschreibt. Durch den Einsatz einer Policy werden die Transitionen zwischen den States durchgeführt (vgl. [Sutton/Barto 2018, 6]).

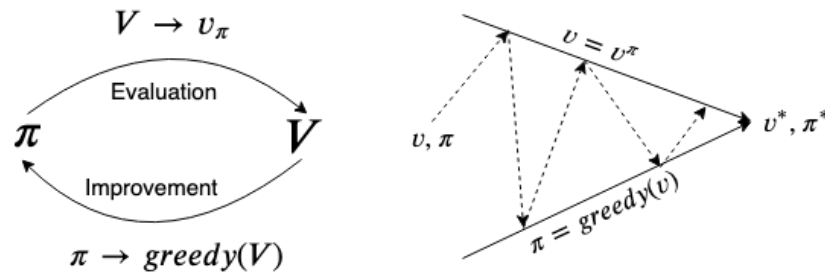


Abbildung 6 GPI Iteration (vgl. [Sutton/Barto 2018, 86f.])

Beide Funktionen, die Policy- und die Value-Funktion, bedingen sich einander. Dieses Zusammenspiel zwischen den beiden Funktionen kann als iterativer Prozess, wie in Abbildung 6 dargestellt (links), beschrieben werden und wird als *Generalized Policy Iteration* (kurz GPI) bezeichnet. Dieser Prozess verallgemeinert die Prozesse der Value- und Policy-Iteration, auf die nicht näher eingegangen werden soll, da das Verständnis der GPI für die Ausführungen genügt. Für tiefere Informationen soll auf [Sutton/Barto 2018] verwiesen sein. Eine Policy folgt in ihrer Auswahl an Actions in bestimmten States der Value-Funktion. Damit die Policy- und Value-Funktionen nicht in einem lokalen Optimum verharren, wird die Policy mit einer gewissen explorativen Wahrscheinlichkeit ausgeführt, sodass auch außerhalb der aktuellen optimalen Value-Funktion States gesehen werden. Dieser Vorgang nennt sich *Policy Improvement*. Daraufhin werden die Werte der States neu berechnet und die Value-Funktion wird somit angepasst. Dieser Vorgang wird als *Policy Evaluation* bezeichnet. Abbildung 6 zeigt (rechts), dass das stetige Durchführen beider Vorgänge zu einem, theoretisch gesehen, globalen Optimum führt (vgl. [Sutton/Barto 2018, 86f.]).

Alle RL-Algorithmen werden unter der Betrachtung der GPI beschrieben. Die Ausführungen in diesem Unterabschnitt bildeten die Grundlage für das Verständnis der Algorithmentypen und Umgebungsinteraktionen.

2.2.4 Q-Value Funktionen

Value-Funktionen sagen aus, wie optimal es ist, in einem aktuellen Zustand zu sein, also was für einen kumulativen Reward man beim Folgen der Policy bis zum Ziel, von diesem Zustand aus, erwarten kann. Fügt man der Betrachtung der Bewertung des aktuellen Zustandes die möglichen Actions in dem State hinzu, so spricht man vom *Q-Learning* bzw. von *Q-Value-Funktionen*. Q-Learning beschreibt im Gegensatz zur Value-Funktion, wie optimal es ist, in einem bestimmten State eine bestimmte Aktion zu wählen. Somit wird die Bewertung, von der aus die Policy ihre Entscheidungen fällt, um die möglichen Aktionen in

2 Theoretische Grundlagen

einem Zustand nochmals erweitert. Dies führt zu einer schnelleren Konvergenz der Optimierung, da durch das Messen des Q-Values sowohl die Action als auch der State bewertet werden, ohne dass bei der Action-Auswahl auf die indirekt gelernte Policy zurückgegriffen werden muss (vgl. [Sutton/Barto 2018, 131f.], [François-Lavet et al. 2018, 228f.] & [François-Lavet et al. 2018, 243f.]).

RL-Algorithmen wie Deep-Q-Networks (vgl. [Mnih et al. 2015]) und dessen Derivate sind erste populäre Vertreter bei der Nutzung des Q-Learnings. Nahezu jeder moderne RL-Algorithmus bedient sich direkt oder indirekt des Q-Values in seinen Berechnungen.

2.2.5 Exploration vs. Exploitation

Interagiert ein Agent mit der Umgebung und versucht, eine optimale Policy zu finden und seine Value-Funktion zu maximieren, muss er sich bei der Wahl seiner Actions zwischen einem explorativen Verhalten und dem befolgen der bereits erlernten Policy entscheiden. Dieser Kompromiss zwischen Exploration und Exploitation ist eines der Kernprobleme im RL. Beide Zustände bedingen sich einander und können nicht exklusiv zu einer brauchbaren Policy führen. Ohne Exploration wird der Agent und seine Policy niemals genügend Zustände sehen, um ein globales Optimum zu finden. Ohne Exploitation kann eine eventuell gefundene optimale Policy nicht befolgt werden, da immer wieder suboptimale Zustände erreicht werden. Somit müssen beide Zustände immer zusammenbetrachtet werden, woraus sich verschiedene Konzepte, Methoden und Algorithmen ergeben haben, diese Problematik anzugehen (vgl. [Sutton/Barto 2018, 26], [Sutton/Barto 2018, 3] & [François-Lavet et al. 2018, 285ff.]).

Ein gängiger explorativer Ansatz ist bspw. *e-greedy*, bei dem mit einer gewissen Wahrscheinlichkeit eine Action in einem Zustand gewählt wird, die nicht der aktuellen Policy folgt (vgl. [Auer et al. 2002]). Ein weiterer ist der *Boltzmann-Ansatz*, der anders als der e-greedy-Ansatz nicht irgendeine zufällige Aktion wählt, sondern mit gewichteten Wahrscheinlichkeiten Aktionen als mehr oder weniger optimal über den Verlauf des Trainings einschätzt (vgl. [Cesa-Bianchi et al. 2017]). Modernere Ansätze, die vor allem bei Umgebungen mit wenig Reward-Signalen eingesetzt werden (z. B. wenn bei einem langen Lauf des Agenten Rewards erst am Ende einer Episode ausgeschüttet werden), basieren bspw. auf der *Curiosity-driven Exploration*. bei diesen wird der Ansatz verfolgt, dass der Agent sich intrinsisch selbst eine Reward-Funktion erlernt und somit seine Rewards selbst generiert (vgl. [Burda et al. 2018] & [Pathak et al. 2017]).

2 Theoretische Grundlagen

Genauere Beschreibungen der explorativen Ansätze können in der beistehenden Literatur nachgelesen werden und werden bei Anwendung an geeigneter Stelle näher beschrieben.

2.2.6 On-/Off-Policy-Learning

Soll ein RL-Algorithmus eine Policy erlernen, muss dieser in Interaktion mit der Umgebung treten. Dies basiert auf dem Austausch von Actions des Agenten und States sowie eines Rewards von der Umgebung an den Agenten. Um möglichst viele States während der Lernphase zu sehen und somit idealerweise irgendwann ein globales Optimum zu finden, muss die Policy mit einem gewissen Anteil an explorativen Verhalten ausgestattet sein. Interagiert der Agent mit der Umgebung und sucht stetig nach neuen, ungesesehenen States, nennt man dieses Verhalten *On-Policy Learning*. Beim On-Policy-Learning lernt der Agent eine Policy, welche in ihrem Charakter immer, selbst wenn diese bereits ein Optimum gefunden hat, zu einem gewissen Anteil explorativ handelt (vgl. [François-Lavet et al. 2018, 241f.], [Sutton/Barto 2018, 103ff.] & [Sutton/Barto 2018, 115f.]). Bekannte Vertreter für On-Policy-Algorithmen sind z. B. SARSA oder PPO (vgl. [Sutton/Barto 2018, 129ff.] & [Schulman et al. 2017]).

Im Gegensatz dazu erlernt das *Off-Policy-Learning* zwei verschiedene Policies bzw. die dazugehörigen Value-Funktionen. Hierbei wird die Bewertung eines States oder State-Action-Paars vom eigentlichen Verhalten getrennt. Die bewertende Policy wird *Target-Policy*, die ausführende *Behaviour-Policy* genannt. Während die Behaviour-Policy mit der Umgebung interagiert und wie On-Policy-Algorithmen einen explorativen Charakter besitzt, lernt die Target-Policy von den gesammelten Erfahrungen der Behaviour-Policy. Das Lernen von bereits durchgeführten, gespeicherten Erfahrungen wird als *Importance Sampling* bezeichnet (vgl. [Sutton/Barto 2018, 115f.], [François-Lavet et al. 2018, 241f.] & [Sutton/Barto 2018, 103ff.]). Bekannte Vertreter für Off-Policy-Algorithmen sind Deep-Q-Networks oder der DDPG-Algorithmus (vgl. [Mnih et al. 2015] & [Lillicrap et al. 2015]).

2.2.7 Model-free vs. Model-based Learning

Während des Lernens eines Agenten interagiert dieser mit der Umgebung, um sein Ziel zu erreichen. Dabei sendet er eine Action zur Umgebung und bekommt einen neuen State sowie einen neuen Reward zurück. Diese direkte Interaktion mit der Umgebung wird als *model-free RL* bezeichnet. Manche Systeme und Umgebungen sind aus verschiedenen Gründen jedoch nicht so häufig ausführbar, wie es für das Training benötigt wird. Das kann z. B. an

2 Theoretische Grundlagen

einem komplexen, teuren oder sicherheitskritischen System liegen (vgl. [Sutton/Barto 2018, 159] & [François-Lavet et al. 2018, 265ff.]).

Unter solchen Rahmenbedingungen kann man sich eines anderen Interaktionsansatzes bedienen, den des *model-based RL*. Beim model-based RL wird in einem ersten Schritt ein Modell einer Umgebung erlernt und danach werden in einem zweiten Schritt die eigentliche Policy- und Value-Function in Interaktion mit dem Modell erlernt. Das Modell einer Umgebung lernt die Zustandsübergänge und Rewards vorherzusagen, um diese dann an die Policy beim Lernen zurückzugeben. Somit muss vom Policy-Netzwerk nicht direkt auf die Umgebung zurückgegriffen werden, sondern nur auf das Modell. Der Vorteil hier ist, dass model-based RL Agenten lernen, States und Rewards vorausszusehen und entsprechend der Aktionen zu planen. Ein Nachteil ist, dass es sowohl beim Erlernen des Modells der Umgebung als auch beim Erlernen der Value-Function, zu Abweichungen vom Idealzustand kommen kann, wodurch im model-based RL zwei Unsicherheitsquellen existieren, anstatt nur einer (vgl. [François-Lavet et al. 2018, 265ff.], [Sutton/Barto 2018, 159] & [Silver 2018b]).

2.2.8 State-/Action-Spaces

Während des Durchlaufens der Umgebung durch den Agenten, in dem dieser auf bestimmte States entsprechende Actions emittiert, ist die Art der gelieferten States und Actions, je nach Algorithmus, für den Optimierungserfolg von Bedeutung. Alle States und Actions können jeweils diskret oder kontinuierlich dargestellt werden.

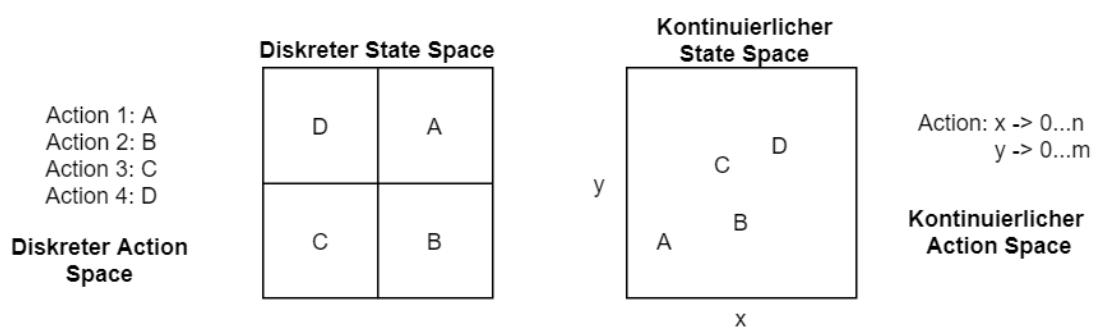


Abbildung 7 Diskreter und kontinuierlicher State / Action Space

Abbildung 7 stellt den Unterschied zwischen diskreten und kontinuierlichen Spaces schematisch dar. Diskret bedeutet sowohl im Fall der Actions als auch der States, dass diese jeweils nur finite Zustände annehmen können. Kontinuierlich hingegen bedeutet, dass die Anzahl an Actions oder States keiner finiten Menge unterliegen und theoretisch, innerhalb von bestimmten Grenzen, unendliche viele Werte annehmen können.

2 Theoretische Grundlagen

Die Eigenschaft, endliche und unendliche Werte zuzulassen, unterscheidet auch die verwendbaren Algorithmen. Da wie bereits in 2.2.3 geschrieben, eine Policy ein Mapping zwischen State-Space und Action-Space herstellt, können nicht in jeder Kombination aus kontinuierlichen / diskreten Action / State-Spaces alle Algorithmen genutzt werden. So kann, im Gegensatz zu einem PPO, ein DQN bspw. keine kontinuierlichen Action-Spaces verarbeiten (vgl. [Sutton/Barto 2018, 335f.] & [Lillicrap et al. 2015, 1]). Für die mathematischen Grundlagen zu den unterschiedlichen Berechnungen von diskreten und kontinuierlichen Verhalten wird auf [Sutton/Barto 2018, 197ff.] verwiesen.

Ein weiteres Kriterium zur Unterscheidung von State-/Action-Spaces ist die Dimensionalität. Je mehr Actions und States, sowohl diskret als auch kontinuierlich, erreichbar sind, desto schwieriger wird es für einen Algorithmus, eine optimale Policy zu finden. Weiterhin können je nach erreichten State, die Anzahl der zugelassenen Actions unterscheiden, wodurch sich die Komplexität bei der Berechnung der optimalen Policy erhöht. Ein weiterführendes Beispiel aus sowohl diskreten als auch kontinuierlichen Action- und State-Spaces hoher Dimensionalität (mögliche 170.000 Actions) und je nach State unterschiedlicher Anzahl an Actions, liefert OpenAI Five, ein Multi-Agenten Setting in einem Multiplayer Online Battle Arena Spiel (vgl. [Matiisen 2018] & [Gauci et al. 2018, 1f.]).

2.2.9 Reward Function Design

Wie bereits in den vorhergehenden Unterabschnitten aufgezeigt, dient das Reward-Signal für den Agenten als Feedback über seine Entscheidungen. Bei welchem State welches Feedback an den Agenten geht, wird durch die Reward-Funktion definiert. Die Reward-Funktion beschreibt somit das Ziel des Agenten im RL (vgl. [Sutton/Barto 2018, 6]).

Für die Erstellung einer geeigneten Reward-Funktion, ist eine geeignete Problemdefinition erforderlich. Hierbei kann man drei verschiedene Reward-Funktions-Spezifikationen unterscheiden. Die **ideale Spezifikation** ist jene, die vom Nutzer erdacht wurde und das ideale Verhalten garantieren soll. Die **Design-Spezifikation** beschreibt die eigentlich umgesetzte Reward-Funktion, wie sie in der Umgebung genutzt wird. Interagiert der Agent mit der Umgebung, ergibt sich die **Verhaltensspezifikation**. Diese ist eine Reward-Funktion, die am besten das Verhalten des Agenten beschreibt. Je komplexer bzw. abstrakter das zu erreichende Ziel ist, desto größer wird die Lücke zwischen Verhaltensspezifikation, also jenem realen Verhalten, und der idealen Spezifikation, zum ideal gewünschten Verhalten (vgl. [Ortega/Maini 2018a]).

2 Theoretische Grundlagen

Eine solche Lücke wird als Spezifikationsproblem bezeichnet. Dieses entsteht durch das Beinhalten von ungewollten Verhalten durch eine inakkurate Definition der Reward-Funktion. Gründe dafür können neben der fälschlichen Definition auch Transformations-Probleme, wie falsche Größenordnungen sowie zu wenig oder falsche Feedback-Signale, sein (vgl. [Leike et al. 2018, 5] & [Leike et al. 2018, 11]).

Problemlösungsansätze können das Verhaltenstesten in speziell dafür vorgesehenen Test-Umgebungen, interpretier- und nachvollziehbare Grafiken und Pläne über die Entscheidungen des Agenten sowie die formale Verifikationen sein (vgl. [Leike et al. 2018, 16f.]).

2.2.10 Policy Gradient

Wie in Unterabschnitt 2.2.4 beschrieben, wird unter Einsatz einer Value-Funktion bzw. einer Q-Value-Funktion gelernt, wie gut es ist, in einem aktuellen Zustand zu sein und eine spezielle Aktion zu wählen. Die Policy, also die Transformations-Funktion vom State Space zum Action Space, wird hierbei indirekt gelernt. Ein anderer Ansatz zur Erschließung einer optimalen Policy sind die sogenannten *Policy-Gradient-Methoden*. Diese parametrisieren das Lernen einer Policy aus State-Action-Paaren direkt und bieten somit die Möglichkeit, auch ohne eine Value-Funktion, eine zumindest garantierte lokal-optimale Policy zu finden. Weiterhin können Policy-Gradient-Methoden effektiver als Value-Funktionen in hochdimensionalen, kontinuierlichen State- und Action-Spaces genutzt werden. Zur Anpassung der Parameter der Policy wird ein Gradienten-Aufstiegs-Verfahren und eine Policy-Gradient-Funktion zur Bewertung der Policy genutzt. Alle Problemstellungen, welche differenzierbar sind, können mithilfe von Policy-Gradient-Methoden angegangen werden, da aufgrund der Funktionalität von Gradienten-Auf-/Abstiegsverfahren mit den Ableitungen der Policy-Funktionen gearbeitet wird (vgl. [Sutton/Barto 2018, 321ff.], [Silver 2018a] & [François-Lavet et al. 2018, 255ff.]).

Unter die Kategorie der Policy Gradient Methoden fallen mehrere Arten von Policy-Gradient Varianten mit verschiedenen Algorithmen. So nutzt der REINFORCE-Algorithmus (vgl. [Sutton et al. 1999]) bspw. eine Form der Policy-Gradient-Methoden, welche nach einem ähnlichen Prinzip wie die der Monte-Carlo-Baumsuche funktioniert. Eine andere, populäre Gruppe von Policy-Gradient-Methoden lernt sowohl eine Policy- als auch eine Value-Funktion, was nachweislich das Optimierungsverfahren stabilisiert und verkürzt. Diese Methode wird als Actor-Critic-Methode bezeichnet, bei der der Actor die Policy- und die Critic eine Value-Funktion erlernt (vgl. [François-Lavet et al. 2018, 259ff.] & [Sutton/Barto

2 Theoretische Grundlagen

2018, 331f.)). Der Soft-Actor-Critic-Algorithmus ist ein Vertreter aus dieser Gruppe (vgl. [Haarnoja et al. 2018]).

Neben den Policy-Gradient-Methoden, welche auf Gradienten-Aufstiegs-Verfahren setzen, sind ferner noch die Gradienten-freien Ansätze zu erwähnen, unter welche die genetischen Algorithmen fallen. Diese bewegen sich nicht mithilfe eines Gradienten-Aufstiegs zu einem Optimum, sondern hin zu einer optimal parametrisierten Lösung, unter Rekombination zufällig parametrisierter Agenten aus einer Population aus Agenten (vgl. [Ashlock 2006, 31f.]).

2.3 Überblick über Sicherheitsrisiken in präskriptiven Algorithmen

Aktuelle und zukünftige präskriptive Systeme bieten ein hohes Potential zur Stiftung eines gesellschaftlichen Mehrwerts in verschiedenen Domänen. Genauso wie positiv, können diese Technologien und Fähigkeiten auch für negative, gesellschaftlich schadhafte Tätigkeiten genutzt werden. Dabei wird davon ausgegangen, dass es zu drei grundlegenden Veränderungen in der Durchführung aktueller Angriffe kommen wird.

Es wird davon ausgegangen, dass aktuell existierende Sicherheitsrisiken expansiver genutzt werden. So werden bspw. Spear-Phishing Attacken, also zielgerichtete Attacken auf Nutzer in z. B. Unternehmen (vgl. [Kaspersky Inc. (Hrsg.) 2019]), einfacher und mit weniger Aufwand umzusetzen sein, um personalisierte E-Mails für das Spear-Phishing zu generieren. Somit wird auch die Einstiegshürde reduziert, professionellen Betrug durchzuführen (vgl. [Brundage et al. 2018, 18ff.]).

Neben der expansiveren Nutzung von Sicherheitsrisiken, werden diese laut Prognose auch ihren Charakter verändern. Somit werden z. B. Angriffe nicht mehr nur auf Personen oder Institutionen gelenkt, sondern auch vermehrt auf vermeintlich vertrauenswürdige KI-Systeme, von denen aus dann wiederum Gefahren, durch bspw. Falsch-Prädiktionen, hervorgehen (vgl. [Brundage et al. 2018, 18ff.]).

Noch problematischer als aktuelle Gefahren, wird das Aufkommen neuer Sicherheitsrisiken mithilfe unter Zuhilfenahme neuester Techniken gesehen. So können falsche Nachrichten gestreut und bestehende abgeändert werden. Eine gezielte Desinformation und Beeinflussung ist hier das Resultat. Auch wird es einfacher, gezielte und hochgradig koordinierte, physische Angriffe mit autonomen Fahr- und Flugzeugen durchzuführen. Ein bisher noch nicht abzuschätzender Schaden, durch die zeitliche und räumliche

2 Theoretische Grundlagen

Ungebundenheit des Angreifers bei solchen Angriffen, ist die Folge. (vgl. [Brundage et al. 2018, 18ff.]).

Diese Szenarien und Veränderungen in Angriffsmustern entsprechen einem gewollten, vorsätzlichen schadhafte Verhalten. Neben dem vorsätzlich schadhafte Verhalten ist auch das nicht-vorsätzliche, ungewollte Verhalten eines Systems zu betrachten. Das ungewollte Verhalten eines Systems ist zwar nicht vorsätzlich schadhaft, jedoch nicht weniger relevant für die Sicherheitsbetrachtungen in aktuellen und zukünftigen präskriptiven Systemen.

Beide Typen von schadhafte Verhalten lassen sich in drei Sicherheitsdomänen einsortieren, die unterschiedliche gesellschaftliche Lebensbereiche beeinflussen können.

Eine dieser Domänen ist die digitale Sicherheit eines Systems. Diese beinhaltet alle Aktivitäten und Risiken der Cybersicherheit von Systemen. Böartig genutzte Technik kann zur automatischen Fehlersuche bei z. B. Betriebssystemen genutzt werden, um Zero-Day-Exploits zu erstellen. Auch ist autonomes Hacking auf menschlichen Level denkbar. Ungewolltes Verhalten kann in dieser Domäne z. B. zur Preisgabe von vertraulichen Informationen in Chatbots führen (vgl. [Brundage et al. 2018, 23ff.]).

Eine weitere Domäne ist die der physischen Sicherheit eines Systems. Diese Domäne beinhaltet alle Schäden, die durch Cypber-Physical-Systems erreicht werden können. So können, wie eingangs bereits beschrieben, autonome Fahrzeuge für Angriffe missbraucht werden. Ungewolltes Verhalten kann zu Schäden von Schutzbefohlenen, z. B. bei Therapiesystemen oder OPs, führen (vgl. [Brundage et al. 2018, 23ff.]).

Die letzte und größte Domäne ist die der politischen Sicherheit. Meinungsbeeinflussung, Fake-News-Generierung, Enttarnung von politischen Meinungsführern und Online-Zensur durch intelligente Filter können die Folge sein. Ungewolltes Verhalten kann z. B. zu Chatbots mit menschenverachtenden bzw. rassistischen „Meinungen“ führen (vgl. [Brundage et al. 2018, 23ff.] & [Beuth 2016]).

Wie aus den obigen Ausführungen ersichtlich, ist Sicherheit im Kontext von selbstlernenden Systemen ein differenziertes und weitreichendes Thema. Es betrifft verschiedenste Ebenen und Interaktionen in der Gesellschaft und kann sowohl absichtlich als auch unabsichtlich zu schadhafte Verhalten führen. Im Rahmen dieser Arbeit und zur Einschränkung dieser komplexen Thematik, sollen für die Lösungsentwicklung lediglich die ungewollten schadhafte Interaktionen betrachtet werden, welche auch im Weiteren näher ausgeführt werden sollen.

2 Theoretische Grundlagen

Sowohl in der Literatur als auch in der Praxis haben sich über die vergangenen Jahre verschiedene technische Probleme bei der Umsetzung von präskriptiven Systemen, vor allem vorangetrieben durch die Entwicklungen im Reinforcement Learning Bereich, ergeben, die ungewolltes Verhalten fördern und das eigentlich vorgesehene Verhalten eines Systems beeinflussen (siehe dazu auch 2.2.9). Tabelle 1 benennt und beschreibt die existierenden Probleme im Detail.

Tabelle 1 Technische Sicherheitsprobleme (vgl. [Leike et al. 2017, 2] & [Amodei et al. 2016, 3])

Sicherheitsproblem	Beschreibung
Safe Interruptibility	Ein System muss unterbrechbar sein, es darf die Unterbrechung nicht verhindern und muss nach der Unterbrechung vom letzten Status aus weiterlaufen.
Avoiding Side Effects	Ein System darf bei der Erreichung seines Zieles keine negativen Seiteneffekte, wie das Beschädigen der Umgebung bei einer Reinigungsaufgabe, provozieren.
Absent Supervisor	Es muss sichergestellt werden, dass das System sich im Einsatz in einer realen Umgebung genauso verhält, wie unter Entwicklungs- und Testverhältnissen.
Reward Hacking	Eine Zielfunktion (Reward-Funktion) oder ein System muss so designt werden, dass das System keine Möglichkeit hat, seine eigene Zielfunktion durch Fehler ungewollt auszunutzen, was zu unvorhersehbarem Verhalten führen kann.
Self-Modification	Unter realen Bedingungen ist ein Agent, das System, teil der Umgebung. Somit kann es möglich sein, dass das System versucht, sich selbst und somit sein Verhalten zu verändern, um seine Zielerreichung zu verbessern.
Distributional Shift	Ein System muss auch dann noch sichere Entscheidungen treffen, wenn sich die Eigenschaften der Umgebung ändern, also seine Distribution, und somit auch die Rahmenbedingungen.
Robustness to Adversaries	Je nach Problemstellung, vor allem bei Multi-Agenten-Systemen, kann sich die Umgebung bzw. können sich die Agenten untereinander, modifizieren und miteinander kooperieren oder sich attackieren. Die Intention der Zielfunktion muss in solchen Fällen gewahrt bleiben.
Safe Exploration	Ein System muss sich in einer Umgebung, egal ob diese bekannt oder unbekannt ist, sicher verhalten und darf nicht unbekannte Risiken zur Durchführung von explorativen Aktionen eingehen.
Scalable Oversight	Ein System soll seine Zielfunktion maximieren, auch wenn es nicht möglich ist, dies zu überprüfen.

Wie aus Tabelle 1 ersichtlich ist, gibt es eine Reihe von Problemen, die ungewolltes Verhalten fördern. Verschiedene Ansätze zur Lösung einzelner Probleme wurden bereits in der Vergangenheit entwickelt, die teilweise jedoch nicht auf jede Problemstellung

2 Theoretische Grundlagen

anwendbar sind. Zur Kategorisierung der Probleme wurde eine Einordnung von Maßnahmen getroffen, die helfen soll, solche Probleme zu erkennen und darauf zu reagieren.

Die erste Kategorie betrifft die Spezifikation eines Systems. Hier muss z. B. auf Fehlerfreiheit, mögliche Seiteneffekte, die Einhaltung von Rahmenbedingungen und die eindeutige Erstellung der Reward-Funktion geachtet werden. In der zweiten Kategorie steht die Robustheit des Systems im Vordergrund. Hier sind z. B. die sichere Exploration in der Umgebung, z. B. durch eine Simulation des Systems, die Verifikation der Einhaltung von Regeln durch das System, die Ausfallsicherheit und die Stabilität bei Fehlern, von Bedeutung. In der letzten der drei Kategorien spielt das Monitoring und die Betriebssicherheit eine Rolle. Die Interpretier- und die Nachvollziehbarkeit der Entscheidungen, die Einflussvorhersagen von Entscheidungen, die Möglichkeiten des menschlichen Eingreifens und der Unterbrechbarkeit des Systems durch den Menschen sind hier u. a. wichtig (vgl. [Ortega/Maini 2018b] & [Leike et al. 2018]).

Jede der drei Kategorien definiert eine Menge von möglichen Maßnahmen, aus denen, je nach Problemstellung, gewählt werden soll, um eine möglichst fehlerfreie und sichere Umsetzung eines Systems gewährleisten zu können.

2.4 Entscheidungsunterstützungssysteme

Systeme, die Entscheidungen auf Basis von vorliegenden Daten treffen, werden bereits seit vielen Jahren betrachtet und aktiv eingesetzt. Solche entscheidungsunterstützenden Systeme werden in der Literatur als Entscheidungsunterstützungssysteme (EUS) bezeichnet. Fasst man aus der Literatur die verschiedenen Definitionen für die EUS zusammen, so sind dies interaktive Systeme, die, vorrangig auf Management-Ebene, semi- und unstrukturierte Problemstellungen datenbasiert mit Modellen, Methoden und der Generierung neuer Daten lösen und somit die Planungs- und Kontrollaufgaben im Management unterstützen (vgl. [Alpar et al. 2016, 33ff.], [Phillips-Wren 2013, 30], [Mertens/Meier 2009, 12] & [Gluchowski et al. 2008, 80]).

Die EUS lassen sich unter die Gruppe der Management-Informationssysteme einordnen und umfassen als Anwender somit alle Management-Ebenen. Dabei unterscheiden sich die EUS von den Management-Informationssystemen dadurch, dass die EUS zur zukünftigen Planung und Kontrolle dienen, also präskriptive oder zumindest prädiktive Ansätze anwenden, währenddessen die Management-Informationssysteme das Management mit

2 Theoretische Grundlagen

sachgerechten, aktuellen und gefilterten Informationen beliefern (deskriptive Analysen) (vgl. [Gluchowski et al. 2008, 80f.] & [Alpar et al. 2016, 246]).

Die Anforderungen an EUS aus der Literatur sind vielseitig und heterogen. Hierunter zählen bspw. die Unterstützung von Entscheidungssituationen in allen Phasen des Entscheidungsprozesses, die Adaptierbarkeit und Flexibilität des Systems oder die Kontrolle des EUS durch den Entscheidungsträger. Diese und weitere Anforderungen an EUS lassen sich in vier Hauptkriterien zusammenfassen. Dazu zählt u. a. die Anforderung der Repräsentation, also die anpassbare und problemorientierte Präsentation der Daten und Ergebnisse. Weiterhin gehört die Anforderung der Operationen dazu, welche problem- und anwendungsorientiert Operationen für die Manipulation der Daten und Ergebnisse bereitstellen soll. Wie auch die Anforderung der Operationen ist eine etablierte Kontrollfunktion des EUS eine bedeutende Anforderung, als auch eine Anforderung für individuelle, also personen- und aufgabenbezogene, Daten- und Informationssichten (vgl. [Gluchowski et al. 2008, 65f.]).

Die Weiterentwicklung der prädiktiven und präskriptiven Methoden, unter Zuhilfenahme von Machine Learning bzw. unter dem Oberbegriff künstliche Intelligenz, führt ebenso zu einer Weiterentwicklung der EUS. Werden diese Methoden und Techniken in EUS eingesetzt, werden die Systeme in der Literatur häufig als Weiterentwicklungen klassischer, formal- und logikbasierter EUS angesehen und als IDSS, im Englischen stehend für *intelligent decision support systems*, also intelligente EUS, bezeichnet (vgl. [Tetali Perraju 2013, 23] & [Phillips-Wren 2013, 26]).

Sie sollen, im Gegensatz zu den klassischen Ansätzen, mehr Unterstützung hinsichtlich des Entscheidungsprozesses bieten und ggf. menschliches Entscheidungsverhalten nachahmen und ggf. selbst durchführen. Intern liegt der Unterschied zu klassischen EUS lediglich im Einsatz der Methoden der künstlichen Intelligenz. Dabei bleibt der grundsätzliche Ablauf des Systems gleich (vgl. [Alpar et al. 2016, 34f.] & [Phillips-Wren 2013, 30f.]).

Neue Anwendungsgebiete sollen sich laut Hinweisen in der Literatur beim Einsatz von intelligenten EUS erschließen. Hierunter zählen z. B. die Fähigkeiten, komplexe unbekannte Situationen richtig einzuschätzen und aus Erfahrungen lernen zu können, Änderungen in der Datenbasis zu erkennen oder auch, im Rahmen von Multi-Agenten-Systemen, eigenständig Optimierungen und Entscheidungen treffen zu können (vgl. [Phillips-Wren 2013, 32] & [Phillips-Wren 2013, 36]).

2.5 Therapeutische Unterstützungssysteme

In der Medizin ist der Begriff der Therapie ein ubiquitär vorhandener und wird zugleich als selbsterklärend dargestellt. Aus dem altgriechischen stammend bedeutet der Begriff *therapeuein* so viel wie „dienen, heilen, pflegen“ (vgl. [Bibliographisches Institut GmbH (Hg.) 2019] & [Digitale Wörterbuch der deutschen Sprache (Hg.) 2019]). In der Disziplin der Humanmedizin hat sich die Definition des Begriffs Therapie so etabliert, als dass eine Therapie alle Maßnahmen zur Linderung und Behandlung von Krankheiten, Behinderungen und Verletzungen zusammenfasst, die die körperlichen oder psychischen Einschränkungen positiv beeinflussen und die Funktionalität versuchen wiederherzustellen (vgl. [Antwerpes/Merz 2017]). Dabei kann allgemein zwischen kausalen, also die Ursachen bekämpfenden, oder symptomatischen, also die Symptome lindernden, Therapien unterschieden werden. Beide folgen der grundlegenden Definition des Therapiebegriffs, betreffen jedoch verschiedene Wirkungsweisen der Therapie, an dessen Enden entweder eine Heilung oder eine Linderung der Krankheit steht (vgl. [Antwerpes/Merz 2017]).

Wie sich bereits aus der Definition des Begriffs Therapie schließen lässt, beinhaltet diese alle Maßnahmen zur Wiederherstellung sämtlicher psychischer und physischer Funktionen. Je nach Teildisziplin der Humanmedizin werden unterschiedliche, teilweise auch kombinierte, Ansätze als Maßnahmen zur Therapie von Krankheiten eingesetzt. Die Chirurgie therapiert mithilfe der physischen Manipulation des Körpers Missstände. In der Psychotherapie werden Therapien sowohl mit verbaler und nonverbaler Kommunikation als auch mit medikamentösen Ansätzen, wie auch hauptsächlich in der inneren Medizin, durchgeführt. Bei der Physiotherapie werden physikalische Maßnahmen wie Wärme- bzw. Kältetherapien genutzt.

Eine besondere Unterscheidung findet sich in der Aufteilung von Therapien zwischen chronischen und nicht-chronischen Erkrankungen. Dabei spricht man i. d. R. von chronischen Krankheiten, wenn diese über einen langen Zeitraum bestehen oder gar nicht mehr verheilen. Diese benötigen langanhaltendere, differenziertere Therapieansätze als Erkrankungen, deren Dauer für den Patienten als kurz oder absehbar zu verstehen sind. Als ein Beispiel für nicht-chronische Erkrankungen steht bspw. die jährlich auftretende Influenza, chronische Erkrankungen sind wiederum z. B. die koronare Herzkrankheit, Herzinsuffizienz oder Diabetes (vgl. [Lange 2011, 54ff.] & [Herold 2013, 239ff.]).

Versucht man hingegen zur Therapie die Therapieunterstützung zu beschreiben, ist die Literatur zur Definition des Begriffes nicht so ergiebig. Der Begriff der

2 Theoretische Grundlagen

Therapieunterstützung wird uneinheitlich und unspezifisch für verschiedenste Verschlagwortungen genutzt. Eine einheitliche Definition ist jedoch nicht zu finden. Der Einsatz des Begriffs der Therapieunterstützung findet sich in allen Bereichen der medizinischen Versorgung, vom Qualitätsmanagement in Kliniken, über medikamentöse Zusatztherapien bis hin zum Patienten-Selbstmonitoring wieder (vgl. [Spreckelsen/Spitzer 2009, 26]). In der aktuellen Entwicklung werden häufig Therapieunterstützungssysteme im Rahmen von ambulanten Lösungen genutzt, die der Patient selbst initiieren und nutzen kann. Ambulant ist hier sowohl im medizinischen Verständnis, also bei der nicht-stationären Therapie, als auch im Verständnis einer nicht ortsgebundenen, nicht zwangsläufig medizinisch-verorteten Anwendung zu verstehen. Dazu zählen bspw. mobile Health (kurz mHealth) Lösungen, wie bspw. mobile Applikationen zur Wundversorgung oder zum Selbstmanagement von Depressionen (vgl. [Elmer et al. 2019, 77ff.] & [Aleithe et al. 2018, 3ff.]).

Im Rahmen dieser Arbeit soll für die Therapieunterstützung folgende Definition gelten und angewendet werden: *Therapieunterstützung bezeichnet technische als auch nicht-technische Systeme und Maßnahmen, die versuchen, den therapeutischen Rahmen für eine oder mehrere Therapien, sowohl für chronische als auch für nicht-chronische Krankheiten, positiv zu beeinflussen, ohne dabei selbst Teil der Therapie zu sein.*

2.5.1 Einordnung & Abgrenzung

Zur wissenschaftlichen und praktischen Einordnung von entscheidungsunterstützenden Therapiesystemen kann eine Klassifizierung der World Health Organization (WHO) für digitale Technologien im Gesundheitswesen zu Rate gezogen werden. Hierin werden digitale Interventionen zur Lösung oder Milderung bestehender Herausforderungen im Gesundheitswesen definiert. Dabei werden die Interventionen in vier verschiedene Kategorien unterteilt, welche verschiedene Rollenträger und Technologien im Gesundheitssystem ansprechen. Hierunter fallen die Patienten des Gesundheitssystems, die Krankenversicherungen, die Gesundheitsmanager im System und die Technologien zur Datensammlung, -verwendung und zum Datenaustausch. Zusätzlich definiert die WHO bestehende Herausforderungen im Gesundheitssystem sowie Systemkategorien, in denen eine weitere Unterteilung der Lösungen stattfinden kann [World Health Organization 2018].

3 Therapietreue & Adhärenz

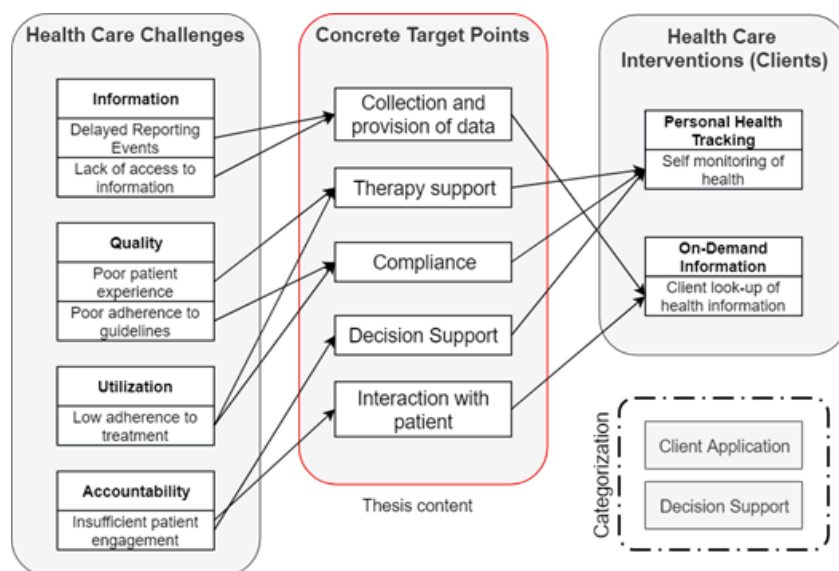


Abbildung 8 Einordnung Therapieunterstützung gemäß WHO

Wie in Abbildung 8 dargestellt, teilt sich die Einordnung in drei Hauptteile. Der Inhalt und das Ziel der Arbeit ist rot umrandet dargestellt. Lesbar ist die Einordnung wie folgt: „Durch die Sammlung und Bereitstellung von Daten über den Patienten (Target Points), können die Herausforderungen der zu späten Mitteilung von Ereignissen und der generelle Mangel an Informationen über die Patienten adressiert werden (Health Care Challenges). Dabei lässt sich die Sammlung und Bereitstellung der Daten in die Gruppe der patientenbezogenen digitalen Interventionen einsortieren, die geprägt ist durch personalisiertes Monitoring mithilfe von körpernaher Sensorik. Die Kategorisierung ist zusätzlich eine von der WHO vorgeschlagene Zuordnung der Interventionen zu Kategorien von Informations- und Telekommunikations-Systemen und bietet eine weitere Unterteilung der Systeme auf technischer Seite an (vgl. [World Health Organization 2018]).

Ersichtlich ist in der Zuordnung zu den Health Care Interventions, dass ausschließlich patientenbezogene Interventionen betrachtet werden. Somit ist die zu entwickelnde Methode nach der Klassifizierung in der WHO Empfehlung darauf ausgelegt, Systeme zu entwickeln, die für die Nutzung durch den Patienten konzipiert werden.

3 Therapietreue & Adhärenz

Erfolg oder Misserfolg einer Therapie wurde im Verlauf der Zeit unterschiedlich beschrieben und gemessen. Hierbei ist der Begriff der Therapietreue als Maßgabe über den Erfolg und Misserfolg einer Therapie zu sehen. Die aktuelle Ansicht aus Praxis und Wissenschaft beschreibt die Therapietreue mit dem Begriff der Adhärenz. Dabei beschreibt Adhärenz die Genauigkeit, wie ein Patient Therapieempfehlungen des medizinischen

3 Therapietreue & Adhärenz

Personals befolgt. Hierbei spielt eine enge Kooperation zwischen dem betreuenden medizinischen Personal und dem Patienten eine wichtige Rolle. Sie beruht auf dem wechselseitigen Austausch zwischen Patienten und medizinischen Personal während der Therapiedurchführung und basiert auf der Annahme, dass für den Therapieerfolg sowohl der Patient als auch der Arzt eine Rolle spielen. Therapietreue wird vorrangig im Zusammenhang mit chronischen Erkrankungen betrachtet, da diese je nach Art, Umfang, Dauer und Schweregrad der Erkrankung und Therapie die Adhärenz beeinflussen (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017] [OECD/European Union 2018, 62ff.]).

Im Verlauf von 1960 bis heute hat sich die Ansicht und Definition von Therapietreue mehrmals geändert. Diese entwickelte sich, von einem direktiven Modell der Therapieumsetzung, also der Arzt verschreibt und der Patient gehorcht, zu einem interaktiven Modell der Therapiebetreuung, indem ein wechselseitiger Austausch zwischen Arzt und Patient stattfindet. Ein weiterer geläufiger Begriff, der „Compliance“, zur Beschreibung der Therapietreue, basiert vor allem auf der Unselbstständigkeit des Patienten, aufgrund dessen dieser ohne Fachwissen keine Entscheidungen über die Therapie treffen kann. Diese Ansicht hat sich über die Zeit, vor allem auch durch einen steigenden Grad an Informiertheit in der Breite der Gesellschaft über leichte technische Zugänge zu stark wachsenden fachspezifischen digitalen Informationsangeboten, weg vom allwissenden Status des medizinischen Personals, hin zu hinterfragenden Patienten geführt. Auch die Änderung der Ansicht von Therapieverläufen, weg von statischen Abläufen hin zu komplexen und situationsabhängigen Prozessen, begünstigte den Wandel der Begriffsdefinition der Therapietreue zur heutigen Definition der Adhärenz, da ein wechselseitiger Austausch zwischen Patient und medizinischen Personal den Verlauf einer Therapie verändern und begünstigen kann (vgl. [Schäfer 2017]).

Für die allgemeine Gesundheit der Weltbevölkerung spielt die Definition und Messung der Therapietreue eine wichtige Rolle, da diese damit die Wirksamkeit des Gesundheitssystems reflektiert und einen gesamtwirtschaftlichen finanziellen Einfluss hat. So werden z. B. 20% der Medikamentenverschreibungen nicht eingelöst, was direkt oder indirekt zu Folgekosten durch Krankenhauseinweisungen und zur Verschlimmerung von Krankheiten führen kann, was wiederum Folgekosten für das Gesundheitssystem nach sich zieht. Laut Schätzungen können ca. 10% aller Krankenhauseinweisungen in Deutschland auf eine falsche oder die nicht-Einnahme von Medikamenten zurückgeführt werden. Das entspricht dem erhobenen statistischen Wert von ca. 50% Einhaltung der Therapieanweisungen in Nicht-

3 Therapietreue & Adhärenz

Entwicklungsländern, was eine geringe Therapietreue widerspiegelt. Dies mündet in jährlichen Kosten von ca. 10 Mrd. Euro allein in Deutschland und bis zu 300 Mrd. USD (ca. 254 Mrd. Euro) jährlich weltweit (vgl. [Weltgesundheitsorganisation 2003; Grønhaug 2017; Schäfer 2017]).

Um Adhärenz messbar machen zu können, sind die Bestimmung ausschlaggebender Faktoren nötig, die die Therapietreue beschreiben. Hierfür hat die World Health Organization (WHO) fünf Kategorien von Einflussfaktoren auf die Therapietreue definiert. Dazu gehören sozioökonomische Faktoren des Patienten, Faktoren über das Gesundheitssystem und des medizinischen Teams, in dem der Patient behandelt wird, zustandsbezogene Faktoren des Patienten, therapiebezogene Faktoren und Faktoren, die auf das Wissen und die Motivation des Patienten abzielen (vgl. [Weltgesundheitsorganisation 2003; Brown/Bussell 2011]).

Aus Abbildung 9 kann erkannt werden, dass sich die fünf Faktoren zu drei Oberkategorien zusammenfassen lassen.

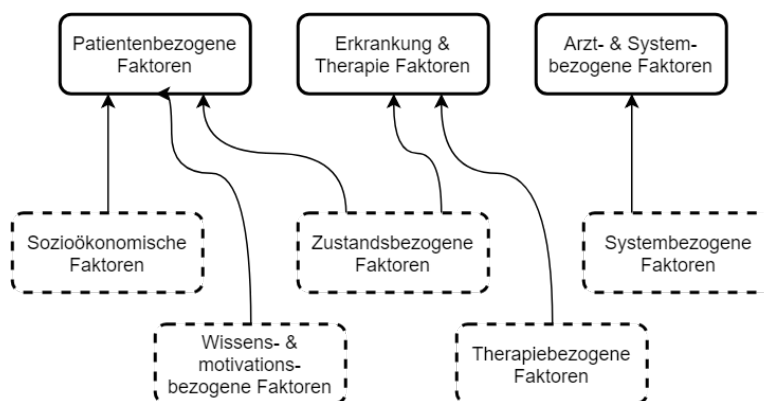


Abbildung 9 Aggregation der Adhärenz-Faktoren-Gruppen (in Anlehnung an [Schäfer 2017; Brown/Bussell 2011])

Unter die patientenbezogenen Faktoren fallen die Kategorien für die sozioökonomischen, zustandsbezogenen und wissens- und motivationsbezogenen Faktoren. Sozioökonomische Faktoren beziehen sich auf das Alter, das Geschlecht, den Bildungsstand, die finanziellen Mittel, den Rückhalt bei der Einhaltung von Therapien aus dem sozialen Umfeld sowie die Umgebungszustände wie Krieg oder Nachkriegszeiten. Wenn auch diese Faktoren in der Literatur ambivalent in ihrer Auswirkung auf die Therapietreue betrachtet werden, so finden sich doch nachvollziehbare Anwendungsfälle für die Einschätzung der Adhärenz unter Einbezug dieser Faktoren. Beispielsweise ist es statistisch relevant, wie alt ein Patient im Hinblick auf den Therapieerfolg ist. So haben Senioren andere Voraussetzungen für therapeutische Erfolge als junge Menschen, da oft Multimorbidität, hoher Konsum an

3 Therapietreue & Adhärenz

verschiedensten Medikamenten (Menschen über 60 Jahre [ca. 12% der Weltbevölkerung] benötigen weltweit 50% aller Medikamente und produzieren damit 60% aller Kosten im Gesundheitssystem) und ein höheres Maß an Unterstützung bei der Einhaltung der therapeutischen Maßnahmen nötig ist (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

Weiterhin zählen zu den patientenbezogenen Faktoren die von der WHO definierten Zustandsfaktoren. Dabei kann zwischen schon vor der Therapie vorhandenen Einschränkungen und Erkrankungen und der durch die Therapie behandelten Erkrankung, und den damit verbundenen Einschränkungen, unterschieden werden. Letztere fällt unter die Oberkategorie „Erkrankung & Therapie“, da hier noch einmal separate Betrachtungen durchgeführt werden. Die bereits vorhandenen Erkrankungen und Einschränkungen beziehen sich auf psychische und physische Einschränkungen (bspw. körperliche und geistige Behinderungen) bzw. schon vorhandene Komorbiditäten wie Depressionen, übermäßiger Alkoholkonsum oder vorhandene Erkrankungen wie AIDS oder Diabetes. Diese schon erworbenen Einschränkungen können Therapieverläufe zusätzlich erschweren oder die zu therapierende Krankheit zusätzlich verschlimmern, was sich wiederum auch auf die Motivation der Therapieverfolgung negativ auswirken kann (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

Eine weitere Faktorengruppe in den patientenbezogenen Faktoren ist die der wissens- und motivationsbezogenen Faktoren. Diese beschreiben die Grundmotivation des Patienten, also seine intrinsischen und extrinsischen Motive, die Therapie durchzuführen. Weiterhin spielt das Verständnis des Patienten über die Notwendigkeit und Auffassung der Therapie eine Rolle. Dieses Verständnis setzt eine adäquate Patientenaufklärung durch das medizinische Personal voraus. Hierzu zählen auch die Rahmenbedingungen, wie bspw. Das Medikamentenregime, die die Einnahme von bestimmten Medikamenten zu gewissen Zeiten in einer gewissen Reihenfolge benötigen. Wurden die Rahmenbedingungen vom Patienten nicht verstanden, ist dies ein Faktor für die Reduzierung der Adhärenz. Auch die Erwartungshaltung des Patienten gegenüber der Therapie ist relevant, da falsch kommunizierte oder missverständlich angenommene Therapieziele dem Patienten gegenüber zu enttäuschenden Ergebnissen führen können (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

Die Faktoren für medizinisches Personal und das Gesundheitssystem beziehen sich einerseits auf das Arzt-Patienten-Verhältnis, andererseits auf die Rahmenbedingungen, in

3 Therapietreue & Adhärenz

der die Therapie stattfindet. Ein Faktor für schlechte Adhärenz ist die mangelnde Kommunikation zwischen medizinischen Personal und Patienten. Das durch die Gespräche entstehende Vertrauen und das Verständnis für die therapeutischen Maßnahmen begünstigt die Therapietreue. Mangelhafte oder fehlende Patientenaufklärung über Therapieverlauf, Rahmenbedingungen, Maßnahmen, Medikamentenregime etc. hingegen reduziert diese. Auch das beim medizinischen Personal fehlende Verständnis für die Relevanz von Adhärenz oder die Erkennung mangelnder Therapietreue führt zu unentdeckten Abweichungen und dem Fehlen von Gegenmaßnahmen. Hierfür wird in der Literatur vor allem Zeit für die Patientenbetreuung genannt, die ein ausschlaggebender Faktor für Adhärenz ist. Systemische Rahmenbedingungen wie Zuzahlungen, Zwei-Klassen Medizin oder das Fehlen von gesundheitlicher Absicherung durch ein Sozialsystem haben zusätzlich Auswirkungen auf die Betreuung und somit auf die Adhärenz. Hierzu zählen auch Systemzustände wie fehlende Medikamente, eine schlechte Ärztedichte oder fehlendes Wissen über Adhärenz (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

Die letzte Gruppe an Faktoren bezieht sich auf die Art und den Verlauf der Therapie sowie auf die Art, die Dauer und die Schwere der Erkrankung. Die Art und der Verlauf der Therapie können sich auf verschiedene Weisen negativ auf die Adhärenz auswirken. Komplexe Abläufe, Nebenwirkungen durch Medikamente, komplizierte und häufige Medikamentenregimes können die Akzeptanz für die Therapiemaßnahmen reduzieren. Auch Einschränkungen im Tagesablauf, die durch Therapiemaßnahmen hervorgerufen werden, haben einen hohen negativen Einfluss auf die Therapietreue.

Abbildung 10 verdeutlicht noch einmal verschiedene Faktoren und deren unterschiedlichen Auswirkungen auf die Adhärenz (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

3 Therapietreue & Adhärenz

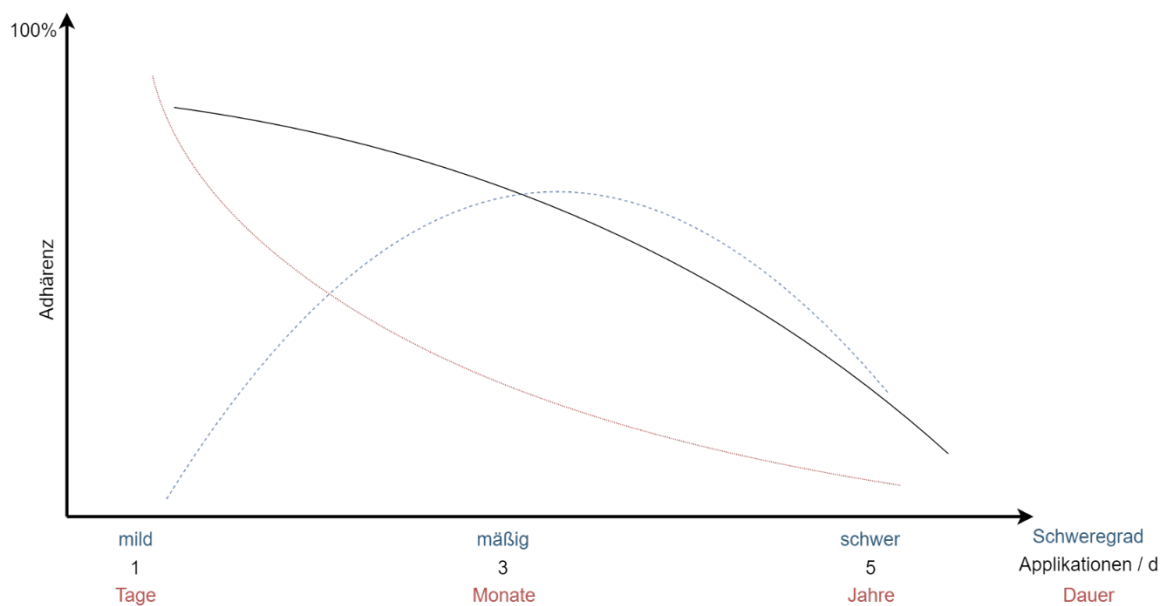


Abbildung 10 Auswirkung verschiedener Faktoren auf Adhärenz (in Anlehnung an [Schäfer 2017])

Ein relevanter Faktor für die Therapietreue ist die Dauer der Therapie bzw. der Erkrankung. Je länger eine Therapie dauert, desto mehr sinkt die Adhärenz. Dies kann durch hohe Erwartungen vom Patienten an die positiven Veränderungen durch die Therapie negativ begünstigt werden, wenn diese erst sehr langfristig und langsam auftreten. Auf eine ähnlich abfallende Weise wirkt sich die Komplexität der Therapie aus. Die Strenge und Anzahl der Applikationen an Medikamenten, Untersuchungen etc. führt ebenfalls zu einem Abfall der Therapietreue. Die Kurve des Schweregrades in Abbildung 10 zeigt, im Gegensatz zu den anderen beiden, eine differenzierte Form des Verlaufs der Therapietreue. Der Ausprägungsgrad der Adhärenz ist am höchsten, wenn die Schwere der Erkrankung sich mäßig beim Patienten äußert. Ist sie mit nur wenigen bis keine Symptome (wie es z. B. bei Bluthochdruck häufig der Fall ist) vorhanden, ist grundsätzlich mit einer niedrigen Adhärenz zu rechnen. Sind die Symptome jedoch sehr stark ausgeprägt und kommen zu diesen Symptomen der Grunderkrankung noch Komorbiditäten und weitere Komplikationen hinzu, sinkt die Adhärenzrate ebenso wie bei einem milden Verlauf der Krankheit ab (vgl. [Schäfer 2017]).

Mit den vorherigen Abschnitten ist es möglich Faktoren zu ermitteln, die die Therapietreue beim Patienten beeinflussen. Wann genau ein Patient jedoch adhärenz ist oder nicht, muss zusätzlich in drei Stufen differenziert betrachtet werden.

Befolgt ein Patient zu über 80% die Therapieabsprachen, so gilt er als idealer Patient und adhärenz. Liegt der Grad der Befolgung zwischen 20-80%, gilt der Patient als partiell-

3 Therapietreue & Adhärenz

adhärent. Dieser Zustand liegt vor allem dann vor, wenn eine deutliche Abweichung vom Therapieplan, also eigenmächtige Änderungen des Patienten, vollzogen wird. Diese Gruppe von Personen kann durch geeignete Maßnahmen am ehesten zu idealen Patienten gewandelt werden. Bei einer Therapiebefolgung von unter 20% spricht man von non-adhärenten Patienten. Diese können hindernd (primäre non-Adhärenz durch nicht-einlösen eines Rezepts) oder modifizierend (sekundäre non-Adhärenz, wie partielle nur wesentlich schlechter) sein (vgl. [Schäfer 2017]).

Diese Gruppen der non-Adhärenz kann man nochmals in drei Untergruppen unterteilen, welche unterschiedliche Maßnahmen zur Verbesserung erfordern. Von non-Adhärenz 1. Ordnung spricht man, wenn aus nachvollziehbaren Gründen eine Therapie komplett abgelehnt wird, z. B. aus religiösen oder Risikogründen. Eine Verbesserung der Adhärenz dieser Gruppe ist sehr schwer bis gar nicht möglich. Non-Adhärenz 2. Ordnung lehnt einzelne Therapieschritte ab, bspw. wegen möglicher Nebenwirkungen. Diese Gruppe kann durch gezielte Gespräche und Aufklärung evtl. überzeugt werden. Die 3. Ordnung der non-Adhärenz sind Patienten, die eigenmächtig Modifikationen der Therapie durchführen. Diese Gruppe hat das höchste Potential, durch Therapiemotivation und Gespräche, gefördert zu werden (vgl. [Schäfer 2017]).

Damit der Grad der Adhärenz bestimmt werden kann, müssen geeignete Messmethoden angewandt werden. Hierbei unterscheidet man zwischen direkten und indirekten Methoden der Messung (siehe Anhang 7). Es gibt keine allgemeingültige, ideale Messmethode, wodurch i. d. R. auf einen Methoden-Mix, je nach Situation, gesetzt wird. Direkte Methoden sind z. B. das Zählen von Medikamenteneinnahmen oder das Messen von Biomarkern im Blut. Diese Methoden sind jedoch nicht immer praktikabel und umsetzbar. Bspw. wären diese Methoden bei Langzeittherapien, vor allem im nicht-stationären Bereich, mit hohem Personal- und Kostenaufwand verbunden. Weiterhin geben direkte Methoden keine Rückschlüsse über die Gründe und Ursprünge der Therapieuntreue. Indirekte Methoden hingegen sind zwar ungenauer als direkte Methoden und stets subjektiv, geben jedoch ein differenzierteres Bild über Hindernisse und Erfahrungen im Therapieverlauf wieder, was Raum für Verbesserungen in der Therapie eröffnet. Zu indirekten Methoden zählen bspw. Patienteninterviews oder Auswertungen von Patiententagebüchern (vgl. [Schäfer 2017]).

Die WHO schlägt zur Erhöhung der Therapietreue vor, negative Effekte auf die Adhärenz zu reduzieren. Dazu zählen z. B. ein vertrautes und stark gefördertes Arzt-Patienten-Verhältnis, den Einbezug der individuellen Lebensumstände und Wünsche der Patienten in

4 Anforderungsanalyse

den Therapieablauf oder die Reduzierung von den im Alltag durch die Therapie erzeugten Einschränkungen sowie ein würdevoller, nicht-erniedrigender Umgang mit dem Patienten. Werden diese Hinweise befolgt, kann dies zu höherer Lebensqualität und Lebenserwartung der Patienten führen, der Reduzierung von Kosten im Gesundheitssystem, der Reduzierung von Komplikationen und Einschränkungen sowie einer Erhöhung der Sicherheit von Patienten und die allgemeine Verbesserung der Gesundheit der Weltbevölkerung (vgl. [Weltgesundheitsorganisation 2003; Schäfer 2017]).

4 Anforderungsanalyse

Im letzten Kapitel wurden die theoretischen Grundlagen für das Verständnis von entscheidungsunterstützenden Systemen im Therapiebereich sowie die dazu nötigen Techniken und vorhandenen Therapietreue-Konzepte vorgestellt. In diesem Kapitel soll ein Zielkriterienkatalog entwickelt werden, welcher die Anforderungen an ein entscheidungsunterstützendes Therapieunterstützungssystem beinhaltet. Dazu werden die Anforderungen für den Katalog identifiziert, beschrieben und bestehende Ansätze aus der Literatur und Industrie entsprechend dem Katalog bewertet.

4.1 Anforderungsanalyse für einen Ansatz

Als Basis für eine Anforderungsanalyse von bestehenden Ansätzen und zur späteren Entwicklung eines eigenen Ansatzes, müssen für diese Analyse Anforderungen an die zu entwickelnde Methode und das Vorgehensmodell gestellt werden. Mithilfe einer systematischen Literaturanalyse und Expertenbefragungen mit Praxispartnern in mehreren einschlägigen Forschungsprojekten, ergaben sich drei übergeordnete Gliederungspunkte für die Anforderungen an die Methode.

Dazu zählen die allgemeinen Anforderungen, welche generelle Eigenschaften der Methode und des Vorgehensmodells beschreiben. Eine dieser Anforderungen ist die eines iterativen Vorgehens, welches iterativ aufbauend Teilergebnisse erzeugt, die validierbare Zwischenergebnisse darstellen. Eine weitere Eigenschaft ist die der Flexibilität, also eine generische Ausrichtung der Methode. Zudem fallen hierunter noch die Nachvollziehbarkeit und die Messbarkeit sowie die Eigenschaft des ergebnisorientierten Vorgehens.

Der zweite Gliederungspunkt, welcher sich aus der Recherche ergeben hat, ist die Notwendigkeit der Beschreibung von Anforderungen für die therapeutischen Anforderungen innerhalb der Methode. Dazu zählen die Betrachtung aller medizinisch und therapeutischen

4 Anforderungsanalyse

Rahmenbedingungen (Therapeutische Abdeckung) sowie die Abbildung der Therapietreue (Patienten Adhärenz) und die Definition der datenschutzrechtlichen Rahmenbedingungen.

Im dritten identifizierten Abschnitt liegt der Fokus auf Anforderungen an die entscheidenden Algorithmen und deren Eigenschaften. Dabei sind u. a. die Machbarkeit und die Nutzung von aktuellen technologischen Entwicklungen (technologieorientiert) von Bedeutung. Anforderungen wie Interdisziplinarität und Nutzenorientierung sollen die Entwicklung dieser Algorithmen auf die wesentlichen zu unterstützenden Aspekte des Vorhabens einschränken, um Komplexität zu reduzieren und Interpretierbarkeit zu vereinfachen. Weiterhin sollen Anforderungen wie simulatives Entwickeln und Prüfkriterien einen sicheren Umgang mit den Algorithmen gewährleisten.

In den nachfolgenden Unterabschnitten werden die einzelnen Anforderungen der Gliederungspunkte detailliert beschrieben und im Anschluss zusammengefasst in einem Anforderungskatalog dargestellt.

4.1.1 Allgemeine Anforderungen an den Ansatz

Allgemeine Anforderungen an die Methode beschreiben deren Unabhängigkeit und dienen der grundlegenden Flexibilität dieser. In den allgemeinen Anforderungen wurden die Eigenschaften der Iteration, Flexibilität, Nachvollziehbarkeit, Messbarkeit und Ergebnisorientierung identifiziert.

Iterativ muss ein Vorgehen sein, um zu ermöglichen, dass das Endergebnis in Teilergebnisse zerlegt werden kann, sodass Änderungen und Fehler, etwa Sicherheitsrisiken oder Missfunktionalität, frühzeitig einfließen bzw. behoben werden können. Ein weiterer Grund ist das Einfließen lassen von Nutzerfeedback in den Entwicklungsprozess, um Änderungen im Design durch Nutzerorientierung gewährleisten zu können (vgl. [Bourque/Fairley 2014, 44]). Diese Anforderung impliziert ebenfalls die sogenannte Rücksprungmöglichkeit, also das zurückgehen von einem Prozessschritt oder Abschnitt zu einem unmittelbar vorherigen.

Flexibilität fordert von einer Methode eine generische Ausrichtung dieser. Somit sollte diese unabhängig von konkreten technologischen Entwicklungen und Use Cases sein, sodass sie ein entsprechendes Abstraktionslevel mit sich bringt. Damit entsteht eine möglichst vielseitige und flexible Anwendungsmöglichkeit der Methode.

Nachvollziehbarkeit erwartet von einer Methode, dass sich Anforderungen über den Prozess bis hin zu einem oder mehreren spezifischen Artefakten zurückverfolgen lassen. Dies hilft

4 Anforderungsanalyse

bei der Prüfung der Umsetzung aller Anforderungen während der Entwicklung des Systems, die zu Beginn und während des Prozesses definiert wurden (vgl. [Riedl 1991, 74ff.]).

Die Messbarkeit als Anforderung fordert die Operationalisierung der Zielerreichung im Prozess. Dabei müssen ausgerichtet an den Vorhabenszielen Metriken definiert werden, die diese quantifizieren. Dieser Anforderungspunkt dient somit ebenfalls der Nachvollziehbarkeit, jedoch nicht unter dem Aspekt der Anforderungen, sondern unter dem Aspekt der Umsetzung der Ziele in dem konkreten Vorhaben.

Als die letzte Anforderung wurde die Ergebnisorientierung identifiziert. Diese soll in einer Methode gewährleisten, immer klar definierte Artefakte oder Beiträge zu bestehenden Artefakten bei Abschluss eines Prozessschrittes zu produzieren. Dabei soll einerseits sichergestellt werden, dass jeder Prozessschritt im Vorgehensmodell der Methode einen Beitrag zur Zielerreichung leistet, andererseits sollen somit auch unnötige Prozessschritte identifiziert und ggf. entfernt werden.

4.1.2 Anforderungen an therapieunterstützende Anwendungen

Unter den spezifischen Anforderungen für die Methode werden im Folgenden diejenigen dargestellt, die die therapeutisch relevanten definieren. Hierunter fallen die Anforderungen an Datenschutzaspekte, die therapeutische Ausrichtung und Abdeckung und die Betrachtung der Patienten Adhärenz.

Datenschutz ist eine relevante Anforderung in der Definition einer solchen Methode, da bei therapeutischen Begleitungen und Unterstützungen medizinisch-relevante Daten verarbeitet werden können, welche grundsätzlich zu den besonders schützenswerten Daten gehören (vgl. [Bundesministerium der Justiz und für Verbraucherschutz 2018] & [European Union 2016]). Die Umsetzung von therapieunterstützenden Systemen kann ohne Daten der Nutzer kaum praktikabel durchgeführt werden.

Die Anforderung der therapeutischen Ausrichtung meint den Einbezug von Probanden und deren individuellen Lebensbedingungen und -abläufen sowie deren therapeutische Maßnahmen und Hintergründe. Es wird hierbei betrachtet, auf welche Lebensumstände sich die Therapiemaßnahmen auswirken und wo für die Unterstützung der Therapie im Vorgehen angesetzt werden kann (vgl. [Riedl 1991, 74ff.]).

In Abgrenzung zur therapeutischen Ausrichtung werden bei der Anforderung der Patienten Adhärenz nicht die therapeutischen Faktoren analysiert und Maßnahmen definiert, sondern

4 Anforderungsanalyse

die konkreten Faktoren für die Bewertung der Therapietreue aufgenommen und ausgewertet (vgl. [Weltgesundheitsorganisation 2003, 4]). Diese Anforderung soll die Messbarkeit der Therapietreue fordern und ermöglichen.

4.1.3 Anforderungen an entscheidende Algorithmen

Der zweite Part der spezifischen Anforderungen an die Methode beschäftigt sich mit denen an die entscheidenden Algorithmen. Hierzu zählen die Machbarkeit, Nutzenorientierung, Interdisziplinarität, Technologieorientierung, die simulative Ausführung und die explizite Betrachtung der Sicherheit.

Die Machbarkeit zielt auf die technologische Umsetzung auf dem aktuellen Stand der Technik ab. Es muss möglich sein, die theoretisch machbaren technologischen Möglichkeiten in ein praktisches und für den Endnutzer nutzbares System zu überführen. Hierbei spielen auch die Anforderungen an die Sicherheit und wirtschaftliche Betrachtungen eine entscheidende Rolle.

Die Nutzenorientierung soll definieren, dass eine Lösung spezifisch für die Anwendungsfälle und die Zielerfüllung des Vorhabens definiert und nicht generalistisch erstellt werden sollte, um die Komplexität zu reduzieren und um eine nachvollziehbarere Sicherheit zu gewährleisten. Dabei soll nicht die Generalisierung des Algorithmus auf mehrere Anwendungsfälle eingeschränkt werden, was i. d. R. eher mehr Sicherheit bewirkt, da der Algorithmus auch auf vorher nicht gesehene Zustände korrekt reagieren kann, sondern die Ausrichtung auf die Zielerreichung des Vorhabens gefördert werden (vgl. [Datenethikkommission der Bundesregierung (Hrsg.) 2019, 163ff.]).

Interdisziplinarität soll gewährleisten, dass entscheidende Algorithmen immer in Zusammenarbeit mit den für das Vorhaben relevanten Domänenexperten bei der Definition der Rahmenbedingungen erstellt werden sollten. Domänenwissen, vor allem im Umgang mit Menschen, kann komplexe Anforderungen an den Algorithmus nach sich ziehen, wodurch Algorithmen Experten diese Anforderungen eventuell missinterpretieren könnten, was wiederum die Sicherheit des Gesamtsystems reduzieren könnte (vgl. [Brundage et al. 2018, 52]).

Die Notwendigkeit des simulativen Erstellens und Testens des Algorithmus erwächst aus den Kosten-, Aufwands- und Sicherheitsbetrachtungen bei der Entwicklung eines Algorithmus. Simulationen bilden eine approximierte Abbildung der zukünftigen Realwelt ab, wodurch i. d. R. Kosten und Aufwand gespart und für die Endnutzer relevante

4 Anforderungsanalyse

Sicherheitsprobleme schneller und ohne Gefahr für die Nutzer identifiziert und beseitigt werden können (vgl. [Amodei et al. 2016, 15]). Somit sollten dafür entsprechende Aktivitäten und Ressourcen im Prozess vorgesehen werden, da bei der Entwicklung bzw. Konfiguration solcher Simulationen sowie beim Design der Testfälle in diesen Simulationen ein großes Aufwandspotential existieren kann (vgl. [Sutton/Barto 2018, 476]).

Die Anforderung der Anpassbarkeit soll lediglich festlegen, dass es möglich sein muss, aufgrund der schnellen technologischen Weiterentwicklung auf diesem Feld eine fortwährende des State of the Art im Projekt durchführen und ggf., unter Abwägung von Aufwand und Nutzen, Änderungen vorzunehmen zu können, um eine bestmögliche Leistung für den Endnutzer zu gewährleisten.

Die letzte Anforderung in diesem Unterabschnitt ist die der expliziten Betrachtung der Sicherheit der Algorithmen (vgl. [Amodei et al. 2016, 14ff.]). Um dies gewährleisten zu können, müssen geeignete Prüf- und Bewertungskriterien definiert werden, um die Sicherheit bei der Umsetzung messbar betrachten zu können (vgl. [Brundage et al. 2018, 16ff.]). Ebenso müssen bei fehlerhaften oder ungenügenden Verhalten der Algorithmen Maßnahmen definiert werden, um den Schutz des Endnutzers sicherzustellen (vgl. [Datenethikkommission der Bundesregierung (Hrsg.) 2019, 165ff.]).

4.1.4 Anforderungskatalog

Zusammenfassend aus den letzten Unterabschnitten werden die definierten Anforderungen in der Tabelle 2 strukturiert dargestellt. Dieser hier definierte und zusammengefasste Anforderungskatalog dient der Evaluation der bestehenden wissenschaftlichen und praktischen Ansätze, die in den folgenden Abschnitten beschrieben und bewertet werden. Zur besseren Übersicht bei der Bewertung werden die einzelnen Anforderungen mit einer Kennung, entsprechend ihrer übergeordneten Gruppen, versehen.

Tabelle 2 Anforderungskatalog an die Methode

Allgemeine Anforderungen		
A1	Iterativ	Ein System muss in Teilergebnisse zerlegt werden können und Rücksprünge zu früheren Phasen des Prozesses erlauben, um frühzeitig auf Nutzer-Feedback und auf veränderte Anforderungen eingehen zu können.
A2	Flexibel	Die Methode muss generisch sein und darf nicht an bestimmte Anwendungsfälle oder Technologien geknüpft sein.
A3	Nachvollziehbar	Die Anforderungen müssen bis zur entsprechenden Funktionalität zurück verfolgbar sein.

4 Anforderungsanalyse

A4	Messbar	Die Zielerreichung und damit verbunden die definierten Ziele in der Methode müssen operationalisierbar sein, um deren Fertigstellungs- und Erfolgsgrad messen zu können.
A5	Ergebnisorientiert	Jede Aktivität im Prozess muss konkrete Artefakte generieren, die einen Beitrag zur Zielerreichung leisten.
Therapeutische Anforderungen		
T1	Datenschutz	Die Methode muss die Analyse, Bewertung und Maßnahmendefinition von datenschutzrelevanten Daten implizieren.
T2	Therapeutische Ausrichtung	Es müssen die individuellen Lebensbedingungen und -abläufe der betrachteten Zielgruppe bei der Definition der Maßnahmen einbezogen werden.
T3	Patienten Adhärenz	Eine Methode muss konkrete Faktoren für die Messung und Bewertung der Therapietreue aufnehmen.
Anforderungen an präskriptive Algorithmen		
P1	Machbar	Die Lösung muss nach aktuellem Stand der Technik sowie nach wirtschaftlichen und sicherheitsrelevanten Aspekten umsetzbar sein.
P2	Nutzenorientiert	Der zu entwickelnde bzw. anzuwendende Algorithmus muss vorrangig zur Zielerreichung des Vorhabens beitragen und entsprechend spezialisiert sein.
P3	Interdisziplinär	Ein solcher Algorithmus muss immer unter Einbezug von Domänenexperten erstellt werden.
P4	Simulativ	Es muss in einer Methode vorgesehen werden, Algorithmen in einer simulativen Umgebung zu testen, bevor diese in Feldstudien an realen Probanden getestet werden.
P5	Anpassbar	Es muss in einer Methode die Möglichkeit der kontinuierlichen Überarbeitung der eingesetzten Technologien und Algorithmen existieren. Sie sollte nicht fest an bestimmten technologischen Konzepten festhalten.
P6	Sicherheitsorientiert	Für einen Algorithmus müssen überprüfbare Messkriterien erstellt und Alternativmaßnahmen definiert werden, um die Sicherheit bei der Ausführung des Algorithmus zu gewährleisten.

4.2 Evaluierung existierender Ansätze

Anhand einer Literaturanalyse wurden eine Reihe von existierenden Ansätzen recherchiert, die unter Einbezug der definierten Anforderungen selektiert und gefiltert wurden. Diese werden unter Berücksichtigung der in Abschnitt 4.1 definierten Anforderungen beschrieben und bewertet.

Im Nachfolgenden werden alle gefundenen Ansätze beschrieben, einzeln nach den Anforderungen bewertet und danach zusammenfassend im Abschnitt 4.3 nochmals strukturiert dargestellt.

4.2.1 A self-adaptive distributed decision support model

Finanziert durch das Science & Technology Departement of Sichuan in China entstand ein generisches Modell zur Erstellung von IoT Anwendungen (vgl. [Zhang et al. 2017]). Dabei unterstützt das Modell Entscheidungsunterstützungs-Funktionalitäten, die sowohl zentral als auch verteilt in dem IoT-System vorhanden und genutzt werden können. Evaluiert wurde das IoT-Modell in einem medizinischen Zentrum für klinische Studien in Sichuan. Hier wurden die automatische Aufnahme aller Aktivitäten der Studienteilnehmer als auch das Management der Medikamenteneinnahme beobachtet und optimiert.

Der allgemeine methodische Ablauf des Vorgehensmodells aus dem Ansatz kann in Abbildung 11 nachvollzogen werden.

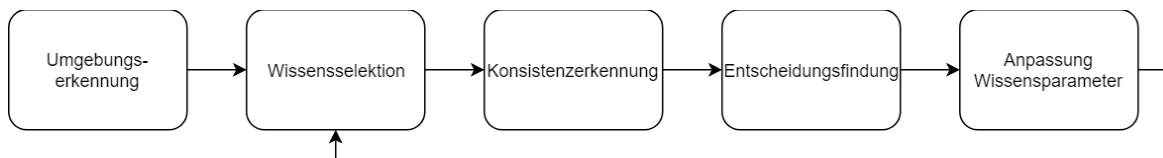


Abbildung 11 Vorgehensmodell nach Zhang et al. [Zhang et al. 2017]

Die fünf dargestellten Prozessschritte führen zu einer kontinuierlichen Erfassung des Kontextes, zur Anpassung und Erweiterung der Wissensbasis und somit einer Anpassung an den aktuellen Kontext. Die Wissensbasis wiederum besteht aus aussage-logischen Formulierungen, die inhärent konsistent sind und zur Entscheidungsfindung genutzt werden.

Der hier vorgestellte Ansatz stellt ein Modell zur Entwicklung von präskriptiven, verteilten Anwendungen dar. Jedoch fehlt es diesem Ansatz bei der Nachvollziehbarkeit der Funktionalität zu den Anforderungen (A3). Vergleicht man die allgemeinen Anforderungen aus Tabelle 3 mit den Eigenschaften des Modells, so werden lediglich der iterative Charakter (A1), wie auch in Abbildung 11 dargestellt, sowie die Messbarkeit der Performance (A4) erfüllt. Gleichwohl kann der Ansatz für verschiedene Domänen in verschiedenen Use Cases genutzt werden (A2) und produziert nach jedem Prozessschritt Artefakte, die in einem folgenden Prozessschritt genutzt werden (A5).

Tabelle 3 Evaluierung Anforderungen Zhang et al. (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	-	+	+	-	-	-	+	+	-	-	+	-

Bei den therapeutischen Anforderungen aus Tabelle 2 wird keine dieser erfüllt, da der allgemeine Charakter sehr generisch gehalten wird, das heißt es werden keine

4 Anforderungsanalyse

therapeutischen Merkmale mit einbezogen (T2, T3), und es wird nicht auf datenschutzrelevante Anforderungen eingegangen (T1). Die Selbstadaptierung unterstellt vermutlich die Anpassung an individuelle Rahmenbedingungen, kann jedoch aufgrund der Komplexität von T2 und T3 diese Rahmenbedingungen nicht implizit verallgemeinern.

Grundsätzlich ist der Algorithmus auf eine große Anwendungsfallzahl adaptierbar und umsetzbar (P1). Auch dient der Algorithmus einem konkreten Zweck (P2) und kann bei neuen Anforderungen angepasst und erweitert werden (P5). Nicht ersichtlich in der Vorgehensbeschreibung des Ansatzes ist der Einsatz eines interdisziplinären Teams zur Entwicklung der Anforderungen (P3), sowie die explizite Betrachtung der Sicherheit der Entscheidungen des Algorithmus (P6). Auch wird das Wissen der Wissensbasis live im Realbetrieb ohne vorhergehende Simulation gesammelt, was eine Reihe von Risiken nach sich zieht (P4, siehe Unterabschnitt 4.1.3).

Insgesamt kann gesagt werden, dass der Ansatz bei der Erstellung des vorgestellten Systems offene Punkte im Bereich Datenschutz und Sicherheit aufzeigt und aufgrund der generischen Aufbauweise keine therapeutischen Anforderungen aufgenommen werden. Weiterhin ist die Architektur fix, worunter die Flexibilität leidet.

4.2.2 IoT-Based Information System for Healthcare Application

Der vorgestellte Ansatz von [Dziak et al. 2017] versucht für die Erstellung von Gesundheitsinformationssystemen einen methodischen Ansatz zu finden, mit den Anforderungen für fachliche und technologische Ansprüche, zusammen mit den späteren Stakeholdern, des Systems definiert werden. Anhand zweier Anwendungsfälle wird der Designprozess durchgespielt.

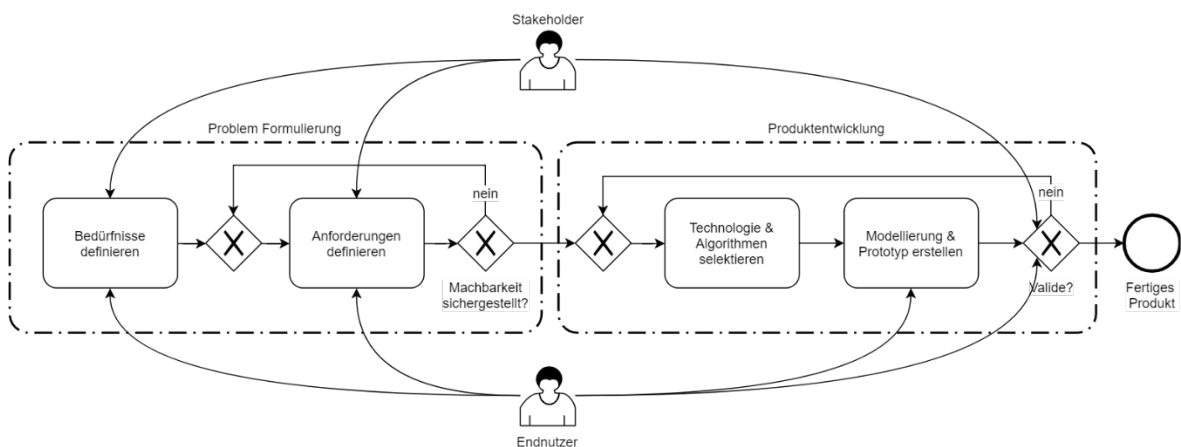


Abbildung 12 Vorgehensmodell nach Dziak et al. [Dziak et al. 2017]

4 Anforderungsanalyse

Der Prozess untergliedert sich in eine Phase der Problemformulierung und eine der Produktentwicklung. Abbildung 12 stellt die einzelnen Schritte der Phase, angelehnt an [Dziak et al. 2017], dar. Stakeholder, inklusive der zukünftigen Nutzer, werden bei der Formulierung der Anforderungen für das System mit eingebunden. Eine Machbarkeitsevaluierung prüft, ob das Vorhaben unter den gegebenen Rahmenbedingungen umsetzbar ist. In einem iterativen Prozess werden Modellierung, Entwicklung und die Technologieauswahl durchgeführt, bis ein anforderungsgerechtes System in der Validierung durch die Endnutzer entstanden ist. Zu bemerken ist, dass auch die Algorithmenauswahl für die Unterstützung separat mit im Designprozess aufgeführt und betrachtet wird.

Tabelle 4 Evaluierung Anforderungen Dziak et al. (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	+	+	+	+	-	-	+	+	+	~	+	~

Wie in der Tabelle 4 zu ersehen, bedient die vorgestellte Methode im vollen Umfang die allgemeinen Anforderungen des Anforderungskatalogs (A1, A2, A3, A4, A5). Auch der Datenschutz (T1) wird explizit angesprochen und als grundlegende Anforderung mit aufgeführt. Trotz der Ausrichtung auf medizinische Produkte wird nur bedingt auf die Therapietreue (T3) sowie nicht auf die individuellen Lebensbedingungen der Alltags der Patienten eingegangen (T2). Unter den Anforderungen für präskriptive Algorithmen werden diese vollständig oder teilweise erfüllt. Während der Technologie- und Algorithmenauswahl werden diese auf Basis der Anforderungen sowie Rahmenbedingungen hinsichtlich technologischer und wirtschaftlicher Betrachtungen gewählt (P1), sodass diese die Anforderungen möglichst genau erfüllen können (P2). Die Definition der Anforderungen und Bedürfnisse erfolgen zusammen mit den Stakeholdern und Endnutzern, welche ebenfalls die Validität des Systems prüfen, wodurch der Algorithmus interdisziplinär getestet wird (P3). Ebenso wird iterativ die eingesetzte Technologie und Algorithmik überprüft und ggf. aktualisiert (P5). Ob die Tests real durchgeführt werden oder simulativ, ist nicht vorgegeben, sollte jedoch eine zwingende Anforderung sein (P4). Ebenso wird zwar die Sicherheit der Lösung adressiert, aber diese nicht explizit geprüft und beschrieben (P6).

In der Summe ist das vorgestellte Vorgehen mit vielen Punkten des Anforderungskatalogs vereinbar, deckt jedoch im Bereich der Therapietreue und Algorithmenauswahl nicht alle Anforderungen ab.

4.2.3 Discovery and Clinical Decision Support

Der hier vorstellte Ansatz von [Yoon et al. 2017] ist ein Entscheidungsunterstützungssystem zur optimalen Empfehlung von Behandlungen für eine nicht eingeschränkte Menge an Krankheiten im stationären Kontext. Im allgemeinen Sinne ist der Ansatz kein Vorgehensmodell, unterliegt aber der Einhaltung der Anforderungen eines Vorgehensmodells zur Erstellung dieses generalistischen Systemansatzes.

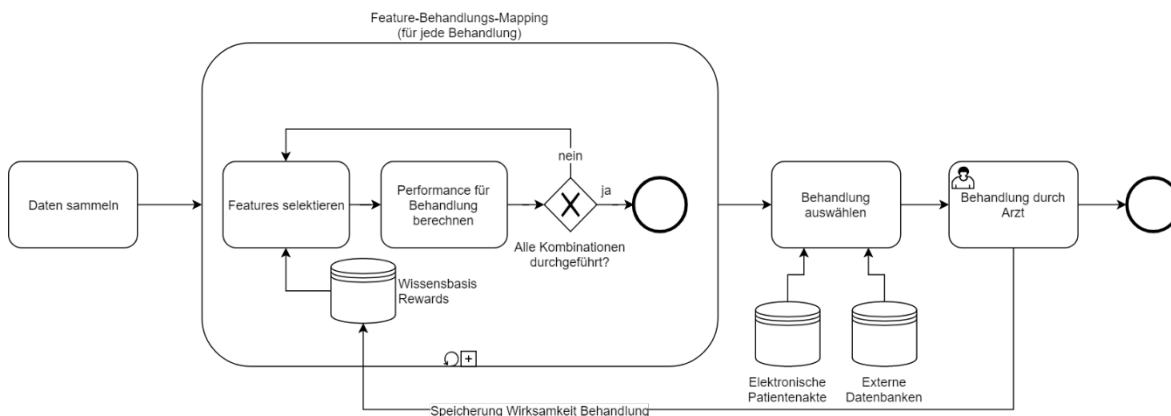


Abbildung 13 Vorgehensmodell Yoon et al. [Yoon et al. 2017]

Abbildung 13 illustriert den groben Ablauf für die Erstellung des hier vorgestellten Ansatzes. Durch den generalistischen Ansatz können für beliebige stationäre Krankheitsbilder (z. B. Brustkrebs) Modelle für die Behandlungsempfehlung erstellt werden. Die gesammelten Daten werden mit den möglichen Behandlungsoptionen bei der gewählten Krankheit verbunden. Dabei wird geprüft, wie relevant die Daten für die einzelnen Behandlungen sind. Hierfür wird auch das Feedback aus bereits durchgeführten Behandlungen mit einbezogen. Daraufhin wird eine Behandlung, basierend auf dem aktuellen gesundheitlichen Zustand des Patienten, ausgewählt, die zusätzlich mit Informationen aus wissenschaftlichen Veröffentlichungen und anderen Datenbanken versorgt wird. Medizinisches Personal führt die Behandlung aus, woraufhin die Behandlungsergebnisse für die iterative Verbesserung des Algorithmus abgespeichert werden.

Tabelle 5 Evaluierung Anforderungen Yoon et al. (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	~	+	+	-	+	-	+	+	-	-	-	-

Trotz der Fokussierung des Ansatzes auf das zu entwickelnde System, hat dieser ausgeprägte Eigenschaften eines Vorgehensmodells (A1, A2, A4, A5). Lediglich die Nachvollziehbarkeit der Umsetzung der Anforderungen ist nicht komplett gegeben, da durch

4 Anforderungsanalyse

die automatisierte Selektion der medizinisch relevanten Daten nicht genau definiert werden kann, woher diese Entscheidung für die Zuordnung der Daten auf die Behandlungen kam (A3). Bei den therapeutischen Anforderungen wird lediglich der aktuelle Zustand des Patienten erfasst, was sich in der Einhaltung der Anforderung (T2) widerspiegelt. Themen wie Datenschutz (T1) und Therapietreue (T3) werden in dem Ansatz nicht betrachtet. Betrachtet man die algorithmischen Anforderungen, erfüllen die eingesetzten Algorithmen sowohl (P1) als auch (P2), da diese dediziert einem bestimmten Zweck dienen und dem Stand der Technik entsprechen. Alle weiteren Anforderungen (P3, P4, P5, P6) werden in dem Ansatz nicht betrachtet und können somit als nicht erfüllt betrachtet werden. Trotz der technischen Ausrichtung des Ansatzes erfüllt dieser fasst alle Anforderungen hinsichtlich allgemeiner Vorgehensmodelle und kann entsprechend als solches betrachtet werden. Lediglich bei den therapeutischen und algorithmischen Anforderungen werden einige Aspekte nicht mit betrachtet bzw. angesprochen und auch nicht indirekt impliziert.

4.2.4 A clinical decision-making mechanism

Der Ansatz verfolgt nach [Forkan/Khalil 2017] die Sammlung von Daten über Wearables im nicht-stationären Bereich. Diese Technik soll die Meldungen von falsch-positiven Ereignissen reduzieren. Unter Nutzung von Anomaliedetektions-Techniken werden die durch die Wearables gesammelten Vitaldaten auf Ausreißer in den Daten untersucht und ggf. wird daraufhin alarmiert. Im finalen Schritt nach der Alarmierung, trifft ein medizinischer Mitarbeiter die Entscheidung über die nächsten Schritte.

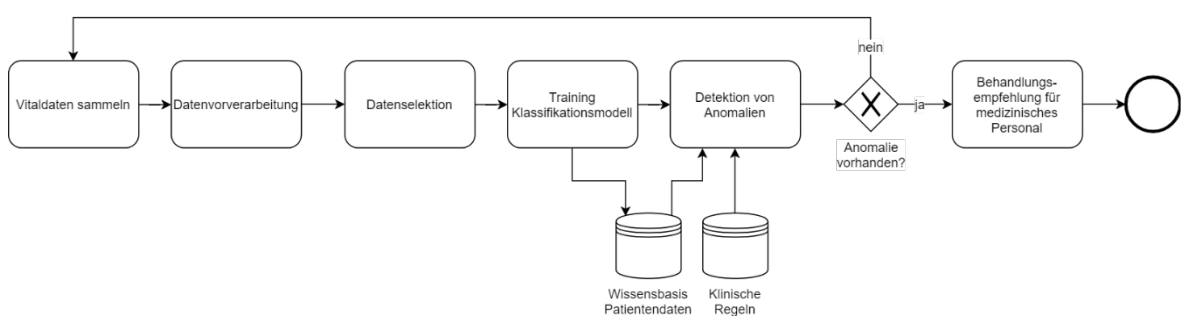


Abbildung 14 Vorgehensmodell Forkan/Khalil [Forkan/Khalil 2017]

Die benötigten Daten werden in einem ersten Schritt durch verschiedene Wearables gesammelt (z. B. Smart Watches), die am Körper des Patienten angebracht sind. Diese werden vorverarbeitet und nach relevanten Daten für die Anomaliedetektion selektiert. Mit der entstandenen Selektion wird ein Modell trainiert, welches die Anomalieerkennung durchführen soll. Unter Einbezug von klinischen Regeln und Grenzwerten werden mit dem

4 Anforderungsanalyse

Klassifikationsmodell die aktuellen Daten über die Vitalsignale geprüft, ob Anomalien vorliegen. Auf dieser Grundlage entscheidet dann das medizinische Personal in der Folge über die weiteren Schritte der Behandlung.

Tabelle 6 Evaluierung Anforderungen Forkan/Khalil (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	+	+	+	-	~	-	+	+	~	~	+	~

Alle allgemeinen Anforderungen für Vorgehensmodelle werden durchgängig vom vorgestellten Ansatz erfüllt (A1, A2, A3, A4, A5). Hingegen werden bei den therapeutischen Anforderungen lediglich die Lebens- und Rahmenbedingungen (T2) vom Patienten teilweise mit einbezogen. Diese werden indirekt über die Vitalsignale aufgenommen und kategorisiert, jedoch nicht weiter mit Kontextdaten über das Verhalten des Patienten oder seine Umgebung angereichert. Datenschutz (T1) oder Therapietreue (T3) spielen im genannten Ansatz keine Rolle. Bei der Betrachtung der algorithmischen Anforderungen wurde im Ansatz auf die Wirtschaftlichkeit und Einfachheit sowie der zielgerichteten Anwendung geachtet (P1, P2). Interdisziplinarität (P3), die die Sicherheit für den Patienten und die anforderungsgerechte Umsetzung sicherstellen soll, wird hier nicht bei der Entwicklung des Algorithmus mit einbezogen. Jedoch wird jede Handlungsempfehlung an das medizinische Personal weitergegeben, die wiederum die finale Entscheidung über eine Maßnahme treffen. Somit gibt es einen indirekten wissenschaftlichen Austausch zwischen Entwicklung und medizinischem Personal. Die Ergebnisse des Algorithmus werden lediglich an einem Testdatensatz geprüft, jedoch nicht in Kombination mit den medizinischen Entscheidungen in einer simulativen Umgebung (P4) validiert. Somit ist diese Anforderung nicht vollständig umgesetzt, kann aber als teilweise erfüllt betrachtet werden, da auch der Zwischenschritt des medizinischen Personals als finalen Vermittler der Maßnahme einen sicherheitsrelevanten Filter verspricht. Auf die Sicherheit der Entscheidungen (P6) des Algorithmus wird nicht konkret eingegangen, ist aber durch die Teilautomatisierung mit dem medizinischen Personal teilweise gegeben. Risiken können dabei in einem möglichen Blindvertrauen des medizinischen Personals in den Algorithmus liegen, das die Entscheidungen nicht mehr hinterfragt. Eine kontinuierliche Verbesserung und Anpassung des Algorithmus ist aber den iterativen Trainingsprozess gegeben (P5).

Der vorgestellte Ansatz bietet eine generalisierte, teilautomatisierte Entscheidungsunterstützung zur Anomaliedetektion. Diskrepanzen zum definierten

4 Anforderungsanalyse

Anforderungskatalog bestehen bei der Therapietreue und dem Datenschutz, sowie bei bestimmten algorithmischen Anforderungen, die auch die Sicherheit tangieren.

4.2.5 Data Analytics Lifecycle

Der Data Analytics Lifecycle (vgl. [EMC Education Services 2015, 25ff.]) ist ein Prozessmodell zur Erstellung von beliebigen Datenanalysemodellen. Dieser wurde speziell für Big Data- und Data Science Projekte entwickelt und besteht aus sechs Hauptphasen, die in Abbildung 15 dargestellt sind. Der Prozess wurde aus einer Komposition bestehender Ansätze zusammengeführt und soll diese in einem übergeordneten Modell kombinieren.

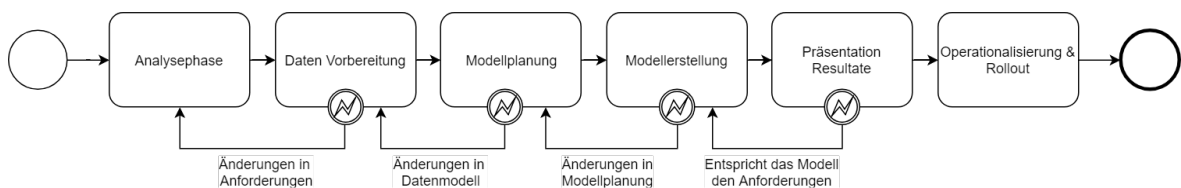


Abbildung 15 Vorgehensmodell EMC [EMC Education Services 2015]

In einer ersten Analysephase werden Stakeholder und deren Anforderungen an das Analysemodell erfasst und definiert sowie die relevanten Datenquellen identifiziert. Daraufhin werden die Daten aus den Datenquellen analysiert und für die Erstellung eines Analysemodells vorbereitet. Zusammen mit den Domänenexperten wird ein Verständnis für die Daten und deren Bedeutung erarbeitet. Bei der Modellplanungsphase werden die Algorithmik definiert, die relevanten Daten für das Modell extrahiert und vorbereitet. Beides fließt in die Modellerstellung ein, in welcher das Modell final entwickelt wird. Die Präsentation der Ergebnisse dient der Validierung des Modells gegenüber den Stakeholdern.

Tabelle 7 Evaluierung Anforderungen EMC Education Services (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	+	+	+	-	-	-	+	+	+	-	+	-

In Tabelle 7 ist ersichtlich, dass alle Anforderungen an ein Vorgehensmodell erfüllt werden (A1, A2, A3, A4, A5). Aufgrund der Ausrichtung des Data Analytics Lifecycle und eine generalisierte Sicht ohne Bezug auf eine Domäne, erfüllt das Modell keine Anforderungen hinsichtlich der Therapietreue (T1, T2, T3). Betrachtet man die Anforderungen an Algorithmen, so werden die technologische und wirtschaftliche Machbarkeit zusammen mit den Stakeholdern bewertet (P1), die Ausrichtung des Ergebnisses auf ein konkretes Ziel (P2) und die interdisziplinäre Erstellung, Weiterentwicklung (P5) und Validierung der Ergebnisse des Modells erfüllt (P3). Nicht explizit gefordert und beschrieben wird die Anforderung,

4 Anforderungsanalyse

dass das Modell mit seinen Ergebnissen vorerst simulativ getestet werden muss (P4). Dies liegt daran, dass das vorgestellte Vorgehen keine präskriptiven Algorithmen abbildet, dafür aber prädiktive. Diese werden i. d. R. von Personen interpretiert und in Handlungen umgewandelt, wodurch eine solche Anforderung nicht notwendig ist. Somit wird auch (P6) nicht betrachtet, da das Modell keine autonomen oder teil-autonomen präskriptiven Algorithmen als Ergebnis definiert.

Das vorgestellte Modell ist ein generisches Modell zur Erstellung von prädiktiven Datenanalysemodellen, ohne Bindung an eine Domäne oder Fachrichtung. Somit erfüllt es nur teilweise die Anforderungen an ein Vorgehensmodell, welche in dieser Arbeit zu Grunde gelegt werden.

4.2.6 ELDAMeth

Die Event-driven Lightweight Distilled state charts-based Agents Methodology, kurz ELDAMeth, ist eine Methode zur Erstellung von agentenbasierten Services. Der Ansatz verbindet die visuelle Service-Modellierung mit einer simulationsgetriebenen Entwicklung in einem iterativen Prozess. Somit soll eine höhere Modularität bei der Entwicklung solcher verteilter Systeme ermöglicht werden.

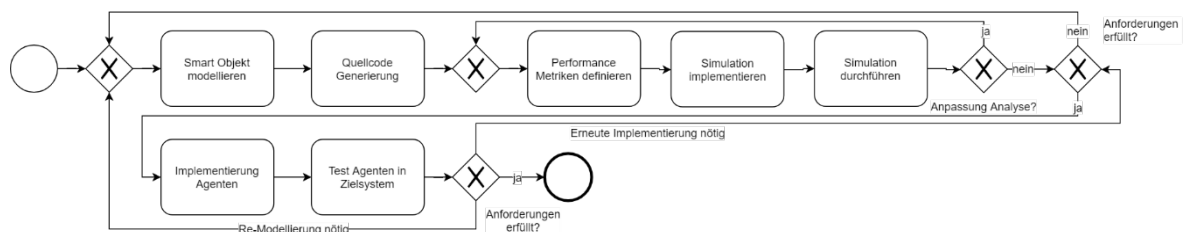


Abbildung 16 Vorgehensmodell Fortino/Russo [Fortino/Russo 2012]

Beginnend mit der Modellierung, wird der Prozessablauf der methode in Abbildung 16 dargestellt. Bei der Modellierung der smarten Objekte werden die zu entwickelnden Services definiert und visuell modelliert. Daraufhin wird automatisiert der Quellcode für die Simulation generiert. Damit das zu entwickelnde System bestimmten Metriken, den Performance-Metriken, und den definierten Anforderungen entspricht, wird eine Simulation des Systems durchgeführt. Sind nach der Beendigung der Simulation die Performance-Metriken oder Anforderungen nicht erfüllt, werden durch einen Rücksprung vorausgegangene Aktivitäten nochmals durchlaufen. Im finalen Part der Implementierung wird der simulierte Prototyp im Zielsystem implementiert und nochmals auf seine Funktionalität hin getestet. Auch hier sind im letzten Schritt noch Rücksprünge bis zum Beginn des Prozesses möglich, sofern die Anforderungen des Service nicht erfüllt werden.

4 Anforderungsanalyse

Tabelle 8 Evaluierung Anforderungen Fortino & Russo (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	+	+	+	-	-	-	~	+	~	+	+	~

Da ELDAMeth als Methode entwickelt wurde, erfüllt diese alle allgemeinen Anforderungen für Vorgehensmodell (A1, A2, A3, A4, A5). Da dieser Ansatz generisch für jede Art von verteilten Agentensystemen entworfen wurde, gibt es bei den therapeutischen Anforderungen keine Erfüllung (T1, T2, T3). Auch wird eine Algorithmenentwicklung (P1) nicht explizit angesprochen, wodurch die Orientierung auf die Machbarkeit bei der Umsetzung der Logik für die Entscheidungen fehlt. Jedoch kann allgemein eine Machbarkeit für ein funktionierendes System ebenfalls angenommen werden. Die Nutzenorientierung (P2) kann mit dem Vorgehen und dessen Anforderungsprüfung unterstellt werden. Eine interdisziplinäre Erstellung der Logik und des Systems ist nicht zu erkennen, könnte jedoch bei Durchführung eines Vorhabens mit betrachtet werden. Ein großer und expliziter Teil der Methode ist das simulative Testen der Logik und des Systems (P4), bevor es zum Anwender kommt und dort getestet wird. Gleiches gilt auch für die Möglichkeit der kontinuierlichen Überarbeitung der Logik (P5). Wie bei einigen bereits genannten Ansätzen wird auch hier nicht direkt auf die Sicherheit der Algorithmen (P6) eingegangen, diese aber durch mehrere Quality Gates und Anforderungs- und Performance-Tests sichergestellt.

Im Ansatz ist dies eine Methode zur Erstellung von agentenbasierten Systemen, wodurch dieser Ansatz im höchsten Maße generisch gehalten wurde. Somit ist keine Ausprägung auf einen therapeutischen Kontext vorhanden. Auch wird die Algorithmik stark generalisiert und lediglich durch Quality Gates kontrolliert aber nicht explizit im Vorgehensmodell betrachtet.

4.2.7 Accelerometer-based step initiation control

Der Ansatz ist ein System mit Vorgehensmodell zur Entwicklung einer therapieunterstützenden adaptiven Neuroprothese. Unter individuellen Gesichtspunkten soll mit dem System Patienten geholfen werden, nach Unfällen wieder richtig zu laufen. Um dies zu ermöglichen, werden mit der Entwicklung eines individuellen Algorithmus einzelne bis mehrere Muskeln stimuliert und auf die jeweilige Person mit ihren individuellen Anforderungen angepasst. Die benötigten Daten werden durch ein Wearable erfasst, welches vom Patienten getragen wird.

4 Anforderungsanalyse

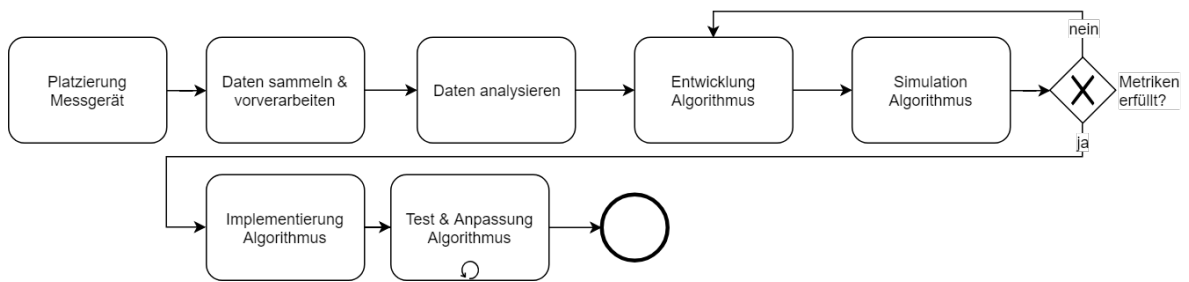


Abbildung 17 Vorgehensmodell Foglyano et al. [Foglyano et al. 2016]

In der Abbildung 17 ist der Vorgehensablauf dargestellt. Je nach Einschränkung muss das Messgerät individuell am Körper platziert werden. Bei der Datensammlung wird ein spezifischer Laufweg eines Individuums aufgenommen und dabei die relevanten Daten über das Messgerät gesammelt. Diese Daten werden in einem darauffolgenden Schritt nach bestimmten Ausschlägen in der Gesamtmenge der Daten beim Laufen des Patienten analysiert. Daraufhin wird ein Algorithmus entwickelt, der diese Ausschläge vorhersagen kann, um entsprechend gegenwirken zu können. Dieser wird mit verschiedenen Simulationen solange verändert, bis er keine falsch-positiven Ereignisse mehr besitzt. Nach der Implementierung des Algorithmus wird dieser weiter unter realen Bedingungen mit dem Patienten zusammen getestet und ggf. weiter feingranular parametrisiert.

Tabelle 9 Evaluierung Anforderungen Foglyano et al. (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	-	+	+	+	-	~	~	+	+	~	+	+	~

Das System bildet den Großteil der allgemeinen Anforderungen an das Vorgehensmodell ab (A1, A3, A4, A5). Lediglich die Flexibilität (A2) ist nicht erfüllt, da das Vorgehen dediziert für die Erstellung von unterstützenden Neuroprothesen erstellt wurde. Datenschutz spielt bei dem aktuellen Ansatz keine Rolle und wurde entsprechend nicht definiert (T1). Der Patientenalltag (T2) sowie die Therapietreue (T3) wurden indirekt einbezogen, in dem der Patient aktiv beim Testen des Systems unter realen Bedingungen mitwirkt und entsprechend Feedback in Form von Verhaltensdaten einbringt. Unter den algorithmischen Anforderungen ist die technische Machbarkeit (P1) und Nutzenorientiertheit (P2) mit im Fokus des Ansatzes und somit erfüllt. Auch das simulative Testen (P4) des Algorithmus vor dem ersten Ausrollen zum Endanwender sowie die ständige Anpassungs- und Überarbeitungsmöglichkeit (P5) sind gegeben. Eine interdisziplinäre Erstellung des Algorithmus wird indirekt durch das Testen unter realen Bedingungen ermöglicht, jedoch nicht vollständig mit Domänenwissen durchgeführt (P3). Die Sicherheit des Algorithmus

4 Anforderungsanalyse

wird lediglich simulativ gewährleistet, jedoch nicht explizit beschrieben oder getestet, wodurch die Anforderung als teilweise erfüllt zu sehen ist (P6).

Der vorgestellte Ansatz ist für die Entwicklung eines spezifischen Typs von Unterstützungssystemen konzipiert. Der Prozess zur Erstellung erfüllt in jeder Anforderungsklasse nur teilweise die Anforderungen an das definierte Vorgehensmodell.

4.2.8 CRISP-DM

Der Cross-Industry Standard Process for Data Mining (kurz CRISP-DM) ist eine Methode aus dem Data-Mining-Bereich zur Durchführung solcher Auswertungen (vgl. [Chapman et al. 2000]). Dabei wurde der Ansatz für deskriptive Verfahren entwickelt, kann aber für eine beliebige Art von Analysetypen genutzt werden (z. B. prädiktive Verfahren). Da der Ansatz als Standardmethode in der Industrie für Data-Mining-Verfahren genutzt wird und in verschiedenen Bereichen und Domänen bereits Anwendung für verschiedene Analysetypen fand, soll dieser Ansatz hier als weit verbreiteter Ansatz mit betrachtet werden.

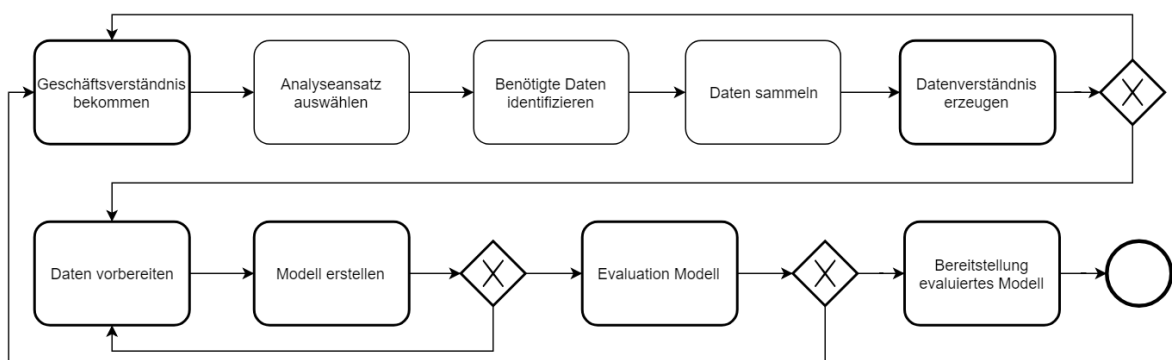


Abbildung 18 Vorgehensmodell Chapman [Chapman et al. 2000]

Die einzelnen Aktivitäten in Abbildung 18 beinhalten weitere Schritte. So enthält der Prozessschritt Modellerstellung die Aktivitäten Modellerstellung, Testen des Modells und Hyperparameter-Tuning. Die dick umrandeten Aktivitäten sind die Kernaktivitäten in der Methode. Hierzu wurden drei zusätzliche Aktivitäten aus den zwei davor und danach stehenden Aktivitäten entnommen, um den Prozess vergleichbarer mit den anderen vorgestellten darstellen zu können (vgl. [Lin 2004]). Die Anforderungsanalyse findet in der ersten Aktivität statt. Danach wird der Analysetyp auf Basis der Zielsetzung ausgewählt (siehe Abschnitt 2.1). Daraufhin müssen die benötigten Daten identifiziert, gesammelt und verstanden werden. Hier können schon Rücksprünge zu den Anforderungen stattfinden, wenn sich eine Anforderung oder das Ziel auf Basis der Daten nicht oder ungenügend lösen lässt. Wenn die Datenqualität genügt, werden die Daten vorverarbeitet und das Modell

4 Anforderungsanalyse

erstellt und gemäß den Anforderungen angepasst. Auch hier ist ein Rücksprung zur Datenvorverarbeitung möglich. Im Anschluss wird das Modell evaluiert und ggf. zurück zur Anforderungsanalyse gegangen, wenn das Ziel nicht erreicht wurde.

Tabelle 10 Evaluierung Anforderungen Chapman et al. (+ erfüllt; ~ teilweise erfüllt; - nicht erfüllt)

A1	A2	A3	A4	A5	T1	T2	T3	P1	P2	P3	P4	P5	P6
+	+	+	+	+	-	-	-	+	+	~	~	+	+

Alle allgemeinen Anforderungen für das Vorgehensmodell werden von der Methode erfüllt (A1, A2, A3, A4, A5). Da die Methode generisch gehalten wird und trotz seiner Adaption auf verschiedene Domänen und Anwendungsfälle eher für die Datenauswertung erstellt wurde, finden sich von (T1-T3) keine Anforderungserfüllungen im Vorgehen wieder. (P1) und (P2) werden beide durch die ganzheitliche Umsetzung und Orientierung an den Anforderungen, mit Möglichkeiten zu Rücksprüngen zur erneuten Anforderungsdefinition, umgesetzt. Auch wenn die Interdisziplinarität (P3) nicht explizit genannt wird, kann diese bei der Ausführung des Prozesses unterstellt und durch die Rücksprünge und Evaluationen umgesetzt werden. Die Simulation (P4) der Ergebnisse vor dem Ausrollen zum Endanwender kann teilweise über die Evaluation angenommen werden. (P5) wird durch die iterative Erstellung des Modells sichergestellt. Die Sicherheit (P6) der Ausgaben der Modelle wird durch die explizite Evaluation dieser erreicht.

Da diese Methode grundsätzlich für die Erstellung von Datenanalyseverfahren erstellt wurde, kann durch die hohe Flexibilität diese auch auf andere Ansätze und Analysetypen adaptiert werden. Die häufigen Rücksprünge, hin zur Anforderungsanalyse, bieten eine hohe Nachvollziehbarkeit und Sicherheit in der Umsetzung der Anforderungen. In den therapeutischen Anforderungen konnte der Ansatz durch seine generische Ausrichtung keine Erfüllung erreichen.

4.3 Zusammenfassung der Evaluierung

Aus der in den voranstehenden Unterabschnitten des Abschnitts 4.2 vorgenommenen Evaluierung der bestehenden Ansätze anhand des Anforderungskataloges aus dem Abschnitt 4.1, lässt sich in der nachstehenden Tabelle 11 eine Übersicht über die Erfüllung (+), nicht-Erfüllung (-) und teilweise Erfüllung (~) erstellen.

4 Anforderungsanalyse

Tabelle 11 Auswertung der Evaluierung bestehender Ansätze

	3.2.1	3.2.2	3.2.3	3.2.4	3.2.5	4.2.6	3.2.7	4.2.8
A1	+	+	+	+	+	+	+	+
A2	+	+	+	+	+	+	-	+
A3	-	+	~	+	+	+	+	+
A4	+	+	+	+	+	+	+	+
A5	+	+	+	+	+	+	+	+
T1	-	+	-	-	-	-	-	-
T2	-	-	+	~	-	-	~	-
T3	-	~	-	-	-	-	~	-
P1	+	+	+	+	+	~	+	+
P2	+	+	+	+	+	+	+	+
P3	-	+	-	~	+	~	~	~
P4	-	~	-	~	~	+	+	~
P5	+	+	-	+	+	+	+	+
P6	-	~	-	~	-	~	~	+

Aus den Tabellendaten wird ersichtlich, dass alle gefundenen Ansätze in jeder definierten Anforderung mindestens eine teilweise Erfüllung erreichen konnten. Schaut man auf die allgemeinen Anforderungen an Vorgehensmodelle (A1-A5), so werden diese bis auf wenige Einzelfälle von allen Ansätzen erfüllt. Ursächlich kann man dies auch auf die Vorauswahl der betrachteten Ansätze reflektieren, da ein Mindestmaß an Übereinstimmung mit den hier in der Zielstellung der Methode definierten Anforderungen vorausgesetzt wurde.

Mit einem Blick auf die therapeutischen Anforderungen (T1-T3) kann, im Gegensatz zu den allgemeinen Anforderungen, festgestellt werden, dass hier nur einzelne Anforderungen vollständig, von einer geringen Menge an Ansätzen, erfüllt wurden. Vor allem in der Betrachtung der Therapietreue muss festgestellt werden, dass diese bei keinem der evaluierten Ansätze im Mittelpunkt steht, in seltenen Fällen allenfalls indirekt mitbetrachtet wird. Die Ausrichtung auf therapeutische Anforderungen lässt sich nur bedingt erklären, jedoch kann gefolgert werden, dass aufgrund der meist technischen Ausrichtung der Lösungsansätze die direkte Zielstellung nicht nachhaltig auf den Patienten ausgelegt ist. Dadurch ist in den seltensten Fällen geklärt, wie weit der Patient und dessen Umfeld mit in die Entwicklung eines solchen Systems einbezogen wird. Strenge Datenschutzanforderungen und fehlende Informationen über die Zielgruppe und deren

5 Methodologische Grundsätze

Verhalten erschweren den Einbezug der therapeutischen Anforderungen, neben (T1), zusätzlich (vgl. [Brown/Bussell 2011]). Ebenso liegt bei den Ansätzen der Fokus auf der Entwicklung eines Systems, jedoch nicht auf dem späteren Betrieb desselben. Deshalb fallen langfristige Konzepte für die Nutzung des Systems, welche durch (T2) und (T3) maßgeblich beeinflusst werden, meist nicht ins Gewicht.

In der letzten Gruppe, den algorithmischen Anforderungen, liegt eine ähnliche Abdeckung wie in den therapeutischen Anforderungen vor. Wobei (P1), (P2) sowie (P5) häufig vollständig bzw. teilweise erfüllt werden, werden die Anforderungen des simulativen Testens (P4), der Interdisziplinarität der Entscheidungsunterstützung (P3) und der expliziten Formulierung der Sicherheit der Entscheidungen (P6) selten bis gar nicht erfüllt. Diese werden meist nur teilweise erfüllt, als inhärent angesehen oder können dem Ansatz nur indirekt angerechnet werden. Gerade im medizinischen Bereich, bzw. in der Therapieunterstützung, ist die Sicherheit der zu treffenden Entscheidungen, wenn diese vor allem voll-automatisch getroffen werden, zum Schutze des Patienten von größter Bedeutung. Einige Ansätze umgehen dieses Problem mit qualitätssichernden Personen, wie bspw. medizinischem Personal, welches die finale Entscheidung trifft, andere Ansätze erwähnen die Probleme gar nicht oder sehen ihr System als nicht kritisch für diesen Sachverhalt an. Interdisziplinarität und Simulation versucht man durch die Anforderungsaufnahme und Testevaluation des Systems abzudecken. Es wird jedoch in den seltensten Fällen explizit beschrieben, wie dies stattfindet.

Schlussfolgernd lässt sich feststellen, dass keiner der betrachteten Ansätze alle Anforderungen für entscheidungsunterstützende Therapiesysteme ganz erfüllt. Eine vollständige Gesamtkombination des Anforderungskataloges wird in keiner Prozessarchitektur schlüssig abgebildet.

5 Methodologische Grundsätze

Im voranstehenden Kapitel der Arbeit wurde ein Anforderungskatalog für die Entwicklung eines entscheidungsunterstützenden Therapieunterstützungssystems entwickelt. Alle verglichenen bereits existierenden Ansätze konnten alle Anforderungen vollständig erfüllen. Somit konnte kein Ansatz identifiziert werden, der bereits alle Zielkriterien erfüllt.

Im folgenden Kapitel werden die methodologischen Grundlagen beschrieben, die für die Entwicklung eines Zielsystems nach den Anforderungskatalog benötigt werden. Hierfür

5 Methodologische Grundsätze

werden verschiedene wissenschaftliche Disziplinen herangezogen, die für die Entwicklung notwendig sind.

5.1 Methoden Engineering

Das Methoden Engineering, ist eine Disziplin innerhalb der Wirtschaftsinformatik und des Software-Engineerings, zur systematischen Erstellung von Entwurfsmethoden. Dabei werden Ansätze aus dem Software-Engineering genutzt, um in einem ingenieurmäßigen Vorgehen Methoden zu entwerfen (vgl. [Gutzwiller 1994, 11ff.]). Vor allem in der Wirtschaftsinformatik hat sich das Methoden Engineering zur Entwicklung neuer Methoden bewährt, da es auf Basis verschiedener Methoden sowohl mit einem theoretischen als auch mit einem praxisbewährten Hintergrund gestaltet wurde (vgl. [Alt/Österle 2004, 258f.]).

Ein Methodenentwurf lässt sich auf fünf Basiskonzepte aufteilen, die in allen Methoden, die mithilfe des Methoden Engineerings entstehen, vorhanden sind (vgl. [Gutzwiller 1994, 11ff.]).

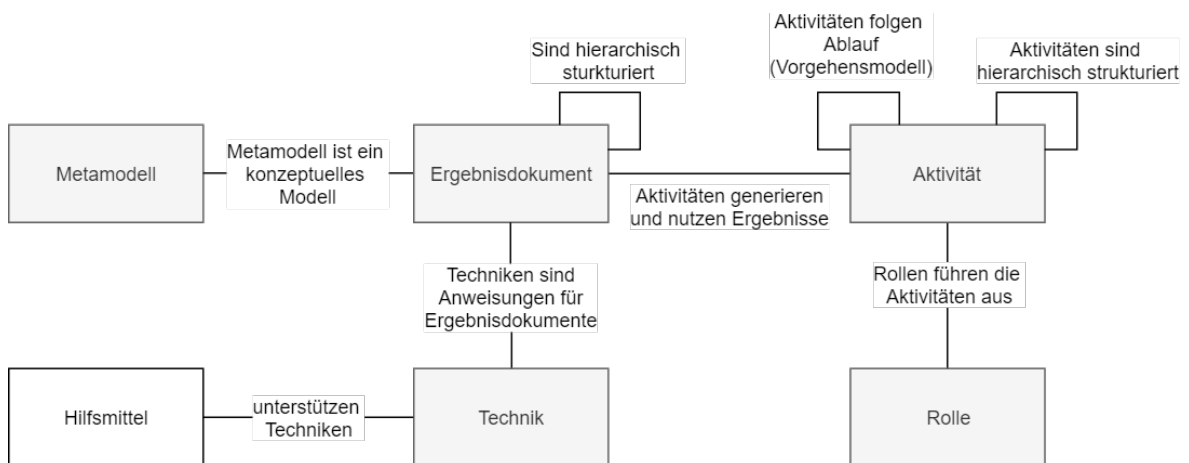


Abbildung 19 Methoden Engineering (in Anlehnung an [Gutzwiller 1994, 13])

Wie aus Abbildung 19 ersichtlich wird, sind alle Konzepte innerhalb einer Methode miteinander direkt oder indirekt verknüpft und somit voneinander abhängig. Die grau hinterlegten Elemente aus der obigen Abbildung sind die Kernelemente einer Methode. Angefangen bei den Aktivitäten können diese hierarchisch strukturiert sein und folgen einem vordefinierten deterministischen Vorgehen, welches als Vorgehensmodell bezeichnet wird. Ausgeführt und bearbeitet werden die Aktivitäten von einer oder mehreren Rollen, die verschiedene Aspekte und Techniken in eine Aktivität einbringen können. Bei Beginn oder Beendigung einer Aktivität werden durch jede Aktivität Ergebnisdokumente erzeugt, die wiederum hierarchisch strukturiert sein können. Um diese Ergebnisdokumente zu erstellen, sind Anweisungen nötig, welche durch die Techniken geliefert werden. Hilfsmittel, wie

Modellierungssprachen etc., können Techniken bei der Erstellung der Ergebnisse unterstützen. Das Metamodell ist ein konzeptuelles Modell, welches die Gestaltungsobjekte für die Erstellung eines Modells beschreibt und diese in Beziehung zueinander setzt. Es gibt die Rahmenbedingungen für die Gestaltung von Modellen in dieser Methode vor (vgl. [Gutzwiller 1994, 11ff.]).

Modernere Weiterentwicklungen des Methoden Engineering referenzieren hauptsächlich auf das sogenannte situationsbedingte Methoden Engineering, in englisch situational method engineering. Dies ist ein Subtyp des Methoden Engineering und beschäftigt sich mit der Anpassung von Methoden auf bestimmte Situationen bei deren Einsatz. Es werden somit einzelne Teile einer Methode situationsbedingt ausgetauscht und modular wieder zu einer Gesamtmethode zusammengeführt (vgl. henderson). Entscheidend für den Einsatz solcher Methoden ist eine Methodenbasis, welche Bausteine für bestimmte Situationen und Domänen beinhaltet (vgl. [Henderson-Sellers et al. 2014, 3ff.]). Da in dieser Arbeit eine allgemeine Methode für die Erstellung von präskriptiven Therapieunterstützungssystemen ohne genauen Situationsbezug entwickelt werden soll, ist das situationsbedingte Methoden Engineering zu speziell. Trotz allem können in Folgeforschungen mehrere situationsbedingte Methoden aus der hier entwickelten allgemeinen Methode entworfen werden, da diese per Definition Subtypen des allgemeinen Methoden Engineerings sind (vgl. [Henderson-Sellers et al. 2014, 3ff.]).

Zur Erstellung des in dieser Arbeit anvisierten Endergebnisses, wird das Methoden Engineering als die geeignetste Methode für die Zielerreichung angesehen und im oben definierten Rahmen und der definierten Art und Weise angewendet.

5.2 Software Engineering

Software Engineering beschreibt das ingenieurmäßige Erstellen von Softwareprodukten, von der Spezifikation bis zur späteren Betreuung der Software, unter Einbezug aller Aspekte der Softwareentwicklung wie Projektmanagement, Techniken und Methoden. Unter den systematischen Ansatz der Erstellung von Software fallen Phasen wie die Software-Spezifikation, -Entwicklung, -Validierung und -Evolution nach dem initialen Rollout (vgl. [Sommerville 2011, 7ff.]). Ähnlich formuliert deckt sich diese zusammenfassende Definition von Software Engineering mit der der IEEE Computer Society (vgl. [Bourque/Fairley 2014, 29]).

5 Methodologische Grundsätze

Die Absicht des Software Engineerings ist, standardisierte, verlässliche und nachvollziehbare Softwareprodukte zu erstellen (vgl. [Sommerville 2011, 8f.]). Mit diesem Prinzip im Vordergrund entstanden verschiedene Arten von Softwareentwicklungsprozessen, von denen drei bekannte und für die Arbeit relevante kurz vorgestellt werden sollen.

Softwareentwicklungsprozesse sollen die Schritte im Software Engineering strukturieren. Die einfachste Struktur ist hier die der sequenziellen Entwicklungsprozesse, wie z. B. das Wasserfallmodell als bekannter Vertreter, welche aufeinander folgend alle Schritte im Prozess nacheinander abarbeiten. Änderungen in bereits abgearbeiteten Phasen fordern ein erneutes Durchlaufen aller Phasen des Prozesses (vgl. [Sommerville 2011, 30ff.]). Zur besseren Einbindung von Kundenfeedback und zur Ermöglichung von Rücksprüngen entstanden inkrementelle Entwicklungsprozesse. Diese erstellen iterativ kleine Softwareelemente, z. B. Prototypen beim Prototyping, die mit jeder Iteration weiter ausgebaut werden. Dabei werden alle Phasen der Softwareentwicklung bei jeder Iteration durchlaufen. Dies ermöglicht eine schnelle Reaktion auf Änderungen und Rücksprünge durch den iterativen Charakter (vgl. [Sommerville 2011, 32ff.]). Parallel zu den bereits genannten Strukturen der Softwareentwicklung bildete sich die agile Entwicklung heraus. Grundsätzlich beschreibt agile Softwareentwicklung eine Ansammlung an Praktiken und Methoden, die auf den Prinzipien des agilen Manifests basieren (vgl. [Agile Alliance (Hrsg.) 2019]). Dabei legt das agile Manifest vier Werte und zwölf Prinzipien fest, woran sich agile Entwicklungsprozesse, wie Scrum oder Extreme Programming, orientieren (vgl. [Mezick et al. 2019, 90f.] & [Beck et al. 2001]). Dabei fokussieren sich agile Prozesse auf Kundeneinbindung, die Einfachheit von Prozessen und Dokumentationen sowie die schnelle Reaktion auf Änderungen in der Entwicklung (vgl. [Sommerville 2011, 58ff.]).

Alle drei genannten Ansätze für die Umsetzung des Software Engineerings sind für die Entwicklung des Vorgehens in der Methode relevant. Aus den sequenziellen Ansätzen wird das grundsätzliche Vorgehen bei Softwareentwicklungsprozessen entlehnt. Aus den inkrementellen Ansätzen werden die Rücksprünge zu vorherigen Phasen und somit die Flexibilität sowie die schnelle Lieferung von Software an den Kunden (in Form von Prototypen) entnommen. Die agilen Ansätze dienen bei der Entwicklung des Vorgehensmodells als Prinzipien, z. B. bei der Kundeneinbindung in allen Phasen des Vorgehensmodells, der ständigen Anpassung der Software an Änderungen, der Freiheit und

Eigenständigkeit des Entwicklungsteams bei der Entwicklung der Software sowie bei dem Versuch des Einfachhaltens des Prozesses und des Dokumentationsvorgehens.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Nachdem im vorangegangenen Kapitel die methodischen Grundlagen für die Durchführung der Entwicklung eines entscheidungsunterstützenden Therapieunterstützungssystems beschrieben wurden, wird die Methode und Vorgehensweise in diesem Kapitel vorgestellt. Somit ist das hier folgende Kapitel der Kern der Arbeit und beschreibt einen Ansatz, basierend auf den Anforderungen aus Unterabschnitt 4.1.4, zur Entwicklung eines entscheidungsunterstützenden Therapieunterstützungssystems und nutzt hierfür die Grundlagen aus Kapitel 2 und die wissenschaftlichen Methoden aus Kapitel 5.

Nach dem Vorgehen im Methoden-Engineering, sind in diesem Kapitel mehrere Abschnitte zu definieren. Diese umfassen das Metamodell der Methode, das Vorgehensmodell sowie die Technik-, Ergebnis- und Rollenmodelle. Das Vorgehensmodell beschreibt das Vorgehen bei der Erstellung eines solchen Systems, indem es Aktivitäten definiert und in einer bestimmten deterministischen Reihenfolge abarbeitet. Jede Aktivität wird mit bestimmten Techniken, definiert im Technikmodell ausgeführt. Dabei sind die Techniken jeweils immer am eigentlichen Vorhaben auszurichten und mehr als Richtung, als ein festes unumstößliches Konstrukt zu verstehen. Gleiches gilt für die Rollen im Vorgehen. Das Rollenmodell beschreibt alle Rollen, die im Vorhaben bei der Erstellung des Zielsystems benötigt werden oder benötigt werden können. Diese sind mit den Aktivitäten, als durchführende oder mitarbeitende Personen, verbunden. Auch hier ist je nach Vorhaben zu prüfen, welche Rollen wirklich benötigt werden, zumal manche Rollen grundsätzlich als optional zu sehen sind und ggf. auch Rollen in ihren Aufgaben zusammengelegt werden können. Zu den Rollen und Techniken besitzt eine Aktivität zusätzlich noch eine Menge an Ergebnissen, was sich auch als Anforderung aus dem Anforderungskatalog ergibt. Diese sind Teilergebnisse des Projektes und finden ihren Mehrwert in einer oder mehreren folgenden Aktivitäten nach ihrer Erzeugung.

Im Folgenden werden, startend mit dem Metamodell der Methode, die genannten Teile des Methoden-Engineerings vorgestellt und beschrieben.

6.1 Überblick Metamodell der Methode

Metamodelle sind nach der Begrifflichkeit Übermodelle, die eine höhere Ebene der Modellierung beschreiben. Ein Metamodell definiert, welche Arten von Modellbausteinen und deren Beziehungen untereinander existieren, um ein konkretes Modell modellieren zu können. Nach der Methode des Methoden Engineering ist das Metamodell das konzeptionelle Datenmodell der Ergebnisse. Aus dem Metamodell lassen sich hier alle Elemente und Verbindungen herauslesen, die benötigt werden, um ein Therapieunterstützungssystem für präskriptive Anwendungen zu erstellen (vgl. [Hars 1994, 11f.], [Gutzwiller 1994, 24ff.] & [Stachowiak 1973, 197]).

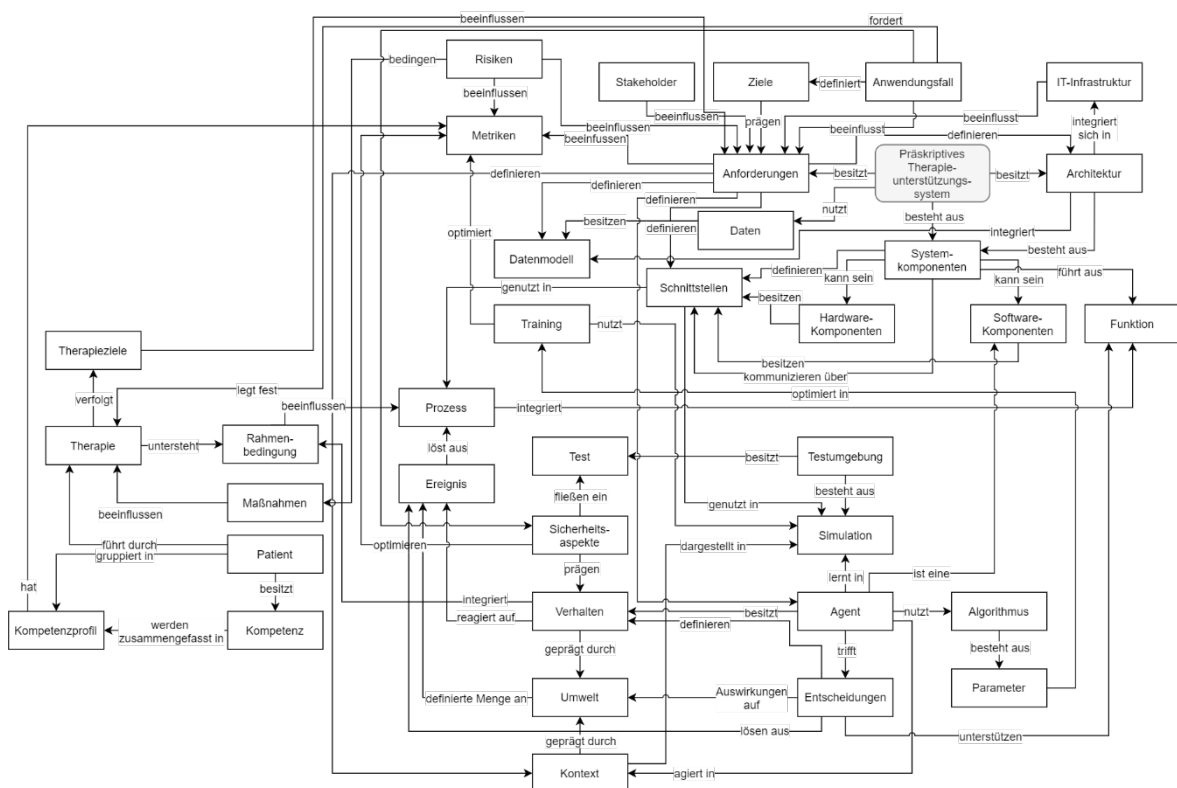


Abbildung 20 Metamodell der Methode (siehe Anhang 2)

Abbildung 20 visualisiert das Metamodell der hier beschriebenen Methode mit seinen Elementen und Verbindungen. Im Zentrum steht das Therapieunterstützungssystem mit seiner präskriptiven Entscheidungsunterstützung. Dieses besteht aus Systemkomponenten und wird durch die Anforderungen und die Architektur aus den Anforderungen definiert. Die Architektur besteht aus den Systemkomponenten und integriert sich in die IT-Infrastruktur. Risiken, Stakeholder, Ziele und Anwendungsfälle fließen in die Anforderungen ein, welche als ein zentrales Element neben der Architektur auch die Sicherheitsaspekte, die Metriken, den Agenten und den Kontext definieren. Der Agent ist

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

mit seinen Entscheidungen das zentrale Element für die Entscheidungsunterstützung. Dessen Entscheidungen orientieren sich, über das Verhalten in der Umgebung und dem Kontext, an den Ereignissen und unterstützen die Funktionalität des Therapieunterstützungssystems. Angelernt und getestet wird der Agent in einer Simulationsumgebung, welche die Umwelt im gegebenen Kontext abbildet. Damit Performance- und Sicherheitsmetriken mitberücksichtigt werden, werden diese sowohl im Training als auch beim Testen mit betrachtet.

Tabelle 12 Übersicht Metamodell-Elemente in einzelnen Phasen

Phase (Haupt- & Teilprozess)	Metamodell-Elemente
Analysephase	Ziele, Anwendungsfall, IT-Infrastruktur, Stakeholder, Risiken, Umwelt, Kontext, Prozesse, Ereignisse
Entwurfsphase	IT-Infrastruktur, Architektur, Anforderungen, Stakeholder, Anwendungsfall, Ziele, Datenmodell, Systemkomponenten, Hardware- / Software-Komponenten, Schnittstellen, Funktion, Agent, Algorithmus, Simulation, Testumgebung, Test, Sicherheitsaspekte, Prozesse, Ereignisse, Kontext, Entscheidungen, Metriken, Risiken
Entwicklungs- & Testphase	Stakeholder, Agent, Simulation, Testumgebung, Test, Parameter, Training, Ereignis, Verhalten, Kontext, Umwelt, Entscheidungen, Funktion, Systemkomponenten, Schnittstellen, Datenmodell, Daten, Risiken
Deployment- & Betriebsphase	Stakeholder, Agent, Systemkomponenten, IT-Infrastruktur, Datenmodell, Prozess, Metriken, Risiken, Umwelt, Kontext

Tabelle 12 stellt noch einmal eine prinzipielle Übersicht der Nutzung der Elemente in den einzelnen Phasen der Haupt- und Teilprozesse des Vorgehensmodells dar. Die Elemente werden mehrmals in den einzelnen Phasen genutzt, um die Methode und das Vorgehen durchführen zu können.

6.2 Überblick Vorgehensmodell der Methode

Das Vorgehensmodell der Methode besteht aus einer Reihe an Aktivitäten, die während des Durchlaufens des Vorgehensmodells einer bestimmten Abfolge gehorchen und aufeinander aufbauen. Dabei werden die einzelnen Aktivitäten in Phasen zusammengefasst, die den Ablauf des Vorgehensmodells in abgeschlossene Abschnitte gliedert. Diese Phasen haben wiederum Vorgehensmodelle inhärent, die nach einer bestimmten Abfolge und Logik ablaufen.

Generalisierte Vorgehensmodelle, wie z. B. das Wasserfallmodell oder das V-Modell XT, sind zwar standardisiert und für eine große Breite an Anwendungsfällen und Domänen einsetzbar, können jedoch die speziellen Eigenheiten und Individualitäten von Projekten mit dem Einsatz von präskriptiven Algorithmen nicht zielführend abbilden. Zielführend bedeutet hier mehr eine Art Leitfaden für die Umsetzung von Projekten und Produkten in der Therapieunterstützung bereitzustellen, der die Entwicklung und Implementierung von sicheren präskriptiven Algorithmen für den Einsatz an Menschen impliziert und anleitet. Gerade der Bereich des Einsatzes von therapieunterstützenden Maßnahmen unter Zuhilfenahme von maschinellen Lernverfahren ist aufgrund der notwendigen Gewährleistung der Sicherheit der Probanden kritisch und muss differenzierter als allgemeine Software-Entwicklungsprojekte betrachtet werden.

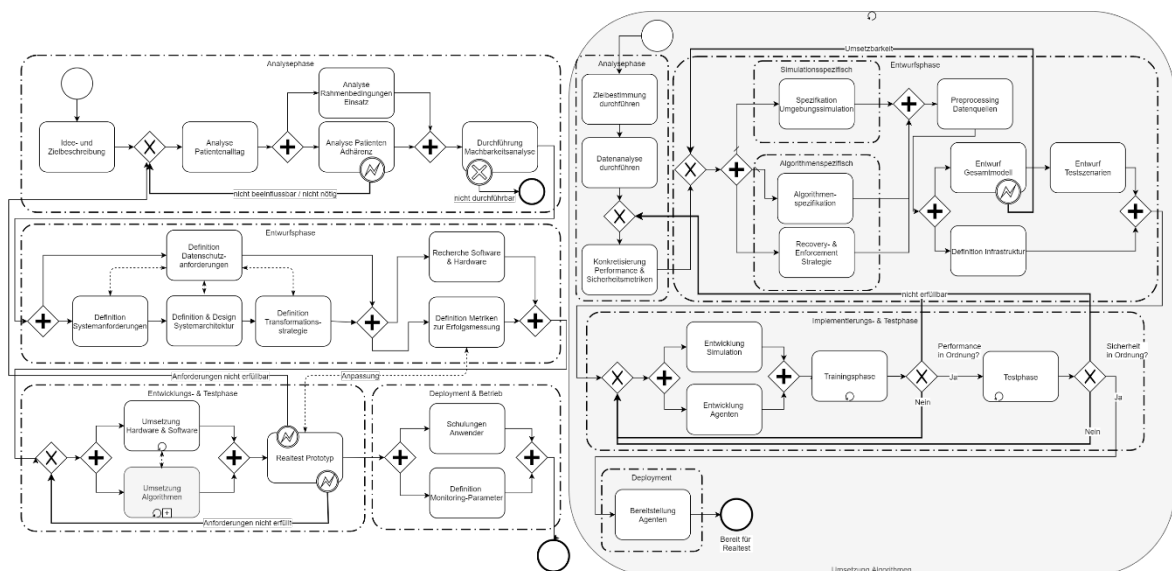


Abbildung 21 Gesamtablauf des Vorgehensmodells (siehe Anhang 1)

Wie in der Abbildung 21 zu erkennen ist, befinden sich im Vorgehensmodell vier klassische Phasen, bekannt und entlehnt aus standardisierten Vorgehensmodellen. Zwischen den Phasen können Rücksprünge möglich sein, wenn sich während des Vorgehens

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Anforderungen geändert bzw. Design-Entscheidungen verworfen wurden. Gleichwohl ist der Prozess ein iterativer, da Aktivitäten und Phasen mehrmals durchlaufen werden können, um Teilergebnisse zu bilden, die separat evaluiert und mit anderen Teilergebnissen zu einem Endergebnis zusammengefasst werden können.

Der Leitgedanke in diesem Vorgehensmodell liegt vor allem bei der Aktivität der Algorithmenumsetzung in der Entwicklungs- & Testphase. Hier wird ein kompletter Sub-Prozess definiert, der parallel zur Entwicklung der Software läuft und die Spezialitäten bei der sicheren Umsetzung von präskriptiven Algorithmen beschreibt und abarbeitet. Somit wird die Algorithmen-Entwicklung von der standardisierten Software-Entwicklung entkoppelt, womit letztere mit gängigen, standardisierten Vorgehensmodellen durchgeführt werden kann, während die Algorithmen-Entwicklung mit dem hier definierten Subprozess umgesetzt wird.

In den folgenden Abschnitten werden als erstes der Hauptprozess, wie in Abbildung 21 dargestellt, beschrieben und danach in einem eigenen Abschnitt der Subprozess durchlaufen.

6.2.1 Analysephase

Zu Beginn des Vorgehensmodells steht die Analysephase, in der der Ist-Zustand aller Rahmenbedingungen und Erwartungen an das System gesammelt und analysiert werden. Hierbei werden die Ziele, der Patientenalltag, die Rahmenbedingungen für den späteren Einsatz des Systems, die Adhärenz und somit auch die therapeutisch relevanten Daten sowie eine Machbarkeitsanalyse erfasst, näher betrachtet und durchgeführt. Das Ziel dieser Phase im Vorgehensmodell ist die vollständige Aufnahme des Ist-Zustandes und der damit verbundenen Rahmenbedingungen (therapeutisch und nicht-therapeutisch), damit diese in späteren Phasen für die Soll-Konzeption des zu entwickelnden Systems übernommen werden können.

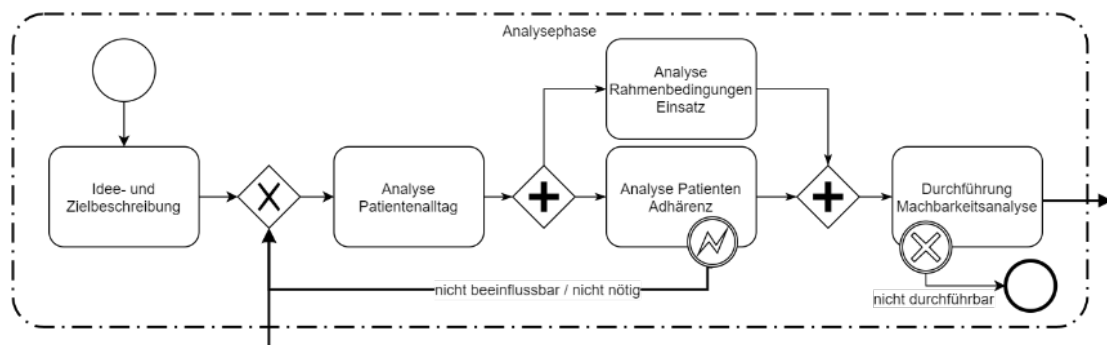


Abbildung 22 Prozessschritte der "Analysephase" im Gesamtprozess

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Abbildung 22 zeigt die logische Abfolge der einzelnen Aktivitäten in der Analysephase auf.

Zu Beginn steht die Aufnahme und Definition von Ideen und Zielen, die das System beschreiben und dessen Zweck definieren. Daraufhin werden bei der Aufnahme des Patientenalltags vor allem der Ist-Zustand über die täglichen Abläufe und Gewohnheiten der Patienten ermittelt. Ist der Alltags-Ist-Zustand ermittelt, müssen zusätzlich zu den täglichen Gewohnheiten und Abläufen weitere Rahmenbedingungen einbezogen werden, die den Patienten direkt oder indirekt betreffen. Solche Rahmenbedingungen können strenge Medikamentenregimes sein, die dem Patienten es nicht oder nur erschwert möglich machen, längere Zeit von seinem Heimatort entfernt zu bleiben. Des Weiteren sind Lebensumfeld, wie soziale oder finanzielle Rahmenbedingungen, entscheidend dafür, wie ein später zu entwickelndes System sich beim Patienten etablieren lässt. Parallel dazu beschäftigt sich die Analyse der Patienten-Adhärenz mit den medizinischen Rahmenbedingungen, welche durch therapeutische Ansätze bestimmte Abläufe oder Einschränkungen diktieren. Hierzu wird die Gruppierung der WHO für Adhärenz Einflüsse genutzt (vgl. [Weltgesundheitsorganisation 2003]). Abschließend wird eine Machbarkeitsanalyse auf Basis der bereits gesammelten Daten und Rahmenbedingungen durchgeführt, um entscheiden zu können, ob das System mit den gegebenen Zielen unter den Gesichtspunkten der Machbarkeitsanalyse realisier- und anwendbar ist.

Abbildung 23 stellt eine Übersicht über die Rollen, Techniken und Ergebnisse der Aktivitäten dieser Phase dar.

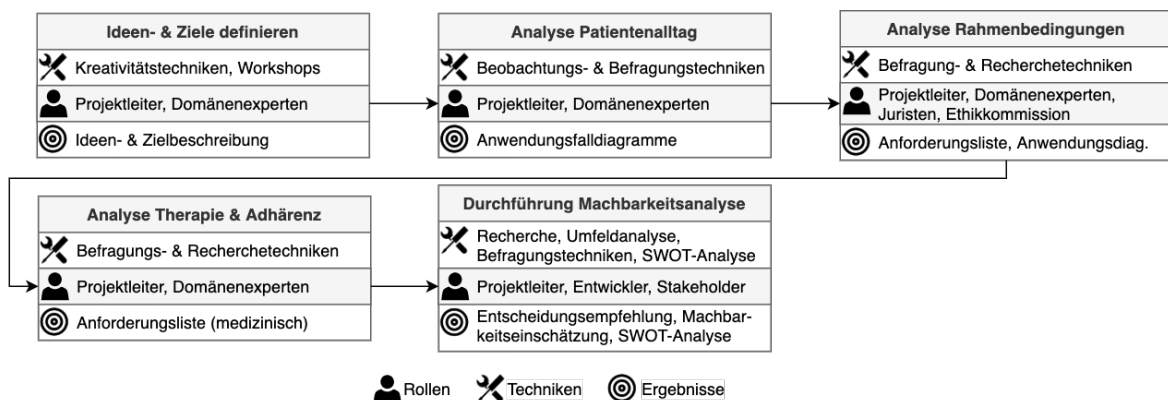


Abbildung 23 Übersicht Aktivitäten für Analysephase

In den folgenden Absätzen werden die genannten Aktivitäten detaillierter ausgeführt und beschrieben.

6.2.1.1 Ideen und Ziele definieren

Zu Beginn der Analysephase steht die Suche nach Ideen und der damit verbundenen Definition von Zielen. Diese Aktivität dient der Klärung der Fragestellung, wie man wo, mit entscheidungsunterstützenden Therapiemaßnahmen, bei den Patienten ansetzen kann. Dafür ist es erforderlich, dass die Domänenexperten der betroffenen Bereiche sowie die Projektleitungsebene zusammenfinden und gemeinsam nach möglichen Anwendungsbereichen und Anwendungsideen suchen.

Da bisher noch keine Definition von Anforderungen oder Rahmenbedingungen stattgefunden hat, können die Ideen mit einem hohen Maß an kreativen Freiheitsgrad gesucht und verfeinert werden. Hierfür bieten sich Kreativitätsmethoden an, welche eine hohe Anzahl an Ideen generieren, die im weiteren Verlauf konsolidiert und verfeinert werden können. Eine dieser Kreativitätsmethoden ist das Brainstorming. Es stellt die Quantität der Ideen vor deren Qualität und bietet somit die Möglichkeit, in kurzer Zeit eine hohe Anzahl an Ideen zu definieren, ohne dabei die Denkweise der Beteiligten einzuschränken (vgl. [Becker et al. 2018]). Die hieraus entstandenen Ideen können daraufhin gemeinsam mit den Domänenexperten reduziert und konsolidiert werden (z. B. in Workshops), sodass eine handelbare Menge an Ansätzen für die Therapieunterstützung übrigbleiben. Bei der Selektion und Reduzierung der Ansätze ist auch zu klären, welche Patientengruppen und Therapien betrachtet werden sollen, da dies später bei der Analyse der Patientenadhärenz von Bedeutung sein wird (siehe Absatz 6.2.1.4). Gleichwohl wird diese Selektion der Patientengruppen dadurch bedingt, dass Art und Umfang der Krankheit sowie der angewandten Therapie, die Schwere der Krankheit und die Dauer der Therapie einen großen Einfluss auf die Therapietreue hat. Um die entstandenen Ideen realistischen Zielen zuordnen zu können, bietet sich die Gruppierung der Ideen nach den Therapienarten an.

Sind die Ideen reduziert und konkretisiert, lassen sich Ziele für die Umsetzung des Vorhabens aus diesen ableiten. Damit die Ziele klar definiert und messbar sind, bietet sich die Anwendung der SMART-Methode, entlehnt aus dem Projektmanagement, an. SMART ist ein Akronym, welches Eigenschaften von Zielen und Zielsetzungen beinhaltet, die für Ziele, u. a. im Projektmanagement, angewandt werden sollten (vgl. [Doran 1981; Tewes 2011]).

S. steht für spezifisch, also die konkrete und eindeutige Beschreibung des Zieles. M. beschreibt die Messbarkeit eines Zieles. Somit muss ein Ziel messbar sein, um dessen Erreichen feststellen zu können. A. steht für *assignable*, also die konkrete Zuweisung, wer

das Ziel umsetzt. R. definiert die Eigenschaft einer Realisierbarkeit unter den gegebenen Einschränkungen für die Zielerreichung. Letztlich beschreibt T. die Eigenschaft der Terminierbarkeit eines Zieles, sodass dieses in einer absehbaren Zeit zu erreichen ist (vgl. [Doran 1981]).

Mit der klaren Definition der Ziele können diese im Verlauf des Vorhabens dezidiert verfolgt und immer wieder messbar abgeglichen werden, was erlaubt, zu jedem Zeitpunkt des Vorhabens genaue Aussagen darüber zu treffen, ob dieses im geplanten zeitlichen und qualitativen Rahmen umsetzbar bleibt oder ob Abweichungen existieren, auf die in der einen oder anderen Form reagiert werden muss.

Als explizites Ergebnis dieser Aktivität steht eine Ideen- und Zielbeschreibung, welche die grundlegenden Ideen und Ziele des Vorhabens exakt beschreibt und klar definiert.

6.2.1.2 Analyse Patientenalltag

Nach erfolgreicher Definition und Festlegung von Ideen und Zielen für das Vorhaben folgt als nächste Aktivität die Analyse des Patientenalltags. Hierbei ist es das Ziel, den aktuellen üblichen Tagesablauf eines Patienten aus der Zielgruppe so genau und relevant als möglich aufzunehmen und zu analysieren. Bei der Analyse liegt die Selektion möglicher Ansatzpunkte für die Verbesserung dieser Abläufe im Fokus der Betrachtung, um therapieunterstützende Maßnahmen zielunterstützend einzubauen. Um diese Ansatzpunkte zu identifizieren, werden die Ideen und Ziele aus der vorherigen Aktivität mit einbezogen.

Zur Erhebung der dafür nötigen Daten können verschiedene Techniken der Befragung und der Beobachtung genutzt werden. Mithilfe von Experteninterviews, als eine Möglichkeit der Befragungstechniken, können Fachexperten zu den Abläufen befragt werden (vgl. [Friedrichs 1990]). Als Fachexperten gelten hierbei nicht nur das medizinische Personal, sondern auch die Patienten selbst, welche einen eigenen subjektiven Blick auf den Tagesablauf, der möglicherweise für den Therapieansatz und die Adhärenz sowie das allgemeine Wohlbefinden wichtigen Einfluss haben kann.

Weiterhin können ergänzend zu den Expertenbefragungen Beobachtungstechniken, wie bspw. eine nicht-teilnehmende unsystematische Beobachtung, durchgeführt werden. Nicht-teilnehmend bedeutet in diesem Kontext, dass der Beobachter den Prozess begleitet, an diesen aber nicht aktiv teilnimmt. Unsystematisch beschreibt hier, dass aufgrund des fehlenden Vorwissens über den Ablauf des Tages diese Beobachtung lediglich Grundregeln der Beobachtung folgt aber weitestgehend frei von methodischen Einschränkungen ist (wie

z. B. in Dokumentationsabläufen). Mithilfe dieser Beobachtung des Patientenalltags können versteckte Abläufe beobachtet werden, die bei den Experteninterviews, aufgrund ihrer bspw. routinierten Vorgehensweise, möglicherweise vergessen worden sind (vgl. [Mayntz et al. 1978; Friedrichs 1990]).

Unter Einbezug der Ergebnisse aus den Beobachtungen und Befragungen können Anwendungsfalldiagramme erstellt werden, die es ermöglichen, eine normierte und strukturierte Sicht auf relevante Bereiche des Patientenalltags zu erstellen, welche fachübergreifend lesbar sind (vgl. [Ambler 2005]). Die Anwendungsfalldiagramme und deren Beschreibungen bilden zudem einen ersten Ist-Soll-Vergleich zwischen dem aktuellen Zustand des Patientenalltags und ersten möglichen Unterstützungsmöglichkeiten, die bei der Aufnahme des Ist-Zustandes ersichtlich wurden. Somit sind die Anwendungsfalldiagramme und deren Beschreibungen die strukturierten Ergebnisse dieser Aktivität, welche später in den folgenden Aktivitäten der Analyse- und Entwurfsphase genutzt werden können.

6.2.1.3 Analyse Rahmenbedingungen

Bei der Analyse der Rahmenbedingungen wird das Ziel verfolgt, eine Anforderungsliste, hinsichtlich rechtlicher, technologischer, organisatorischer und ethischer Beschränkungen, für das Vorhaben zu erstellen. Bei diesen Beschränkungen handelt es sich um Vorgaben, die die funktionalen Anforderungen und die Qualitätsanforderungen im Vorhaben beeinflussen und in der Regel nur schwer geändert werden können (vgl. [Patig/Dibbern 2018]).

Um die Rahmenbedingungen systematisch aufzunehmen bietet sich das methodische Vorgehen des Requirements Engineering an (vgl. [Sommerville 2011]). Dieses gibt einen iterativen Prozess zur Ermittlung, Analyse, Spezifikation und Validierung von Anforderungen vor (vgl. [Sommerville 2011; Patig/Dibbern 2018]). Dabei werden fortlaufend die Anforderungen an das Zielsystem aufgenommen und verfeinert. In der aktuellen Aktivität beschränkt sich die Aufnahme der Anforderungen lediglich auf die vorab benannten Rahmenbedingungen. Funktionale- und Qualitätsanforderungen werden in späteren Aktivitäten, u. a. basierend auf den Rahmenbedingungen, ermittelt und definiert. Spezifikation und Validierung des Anforderungsaufnahmeprozesses dienen der genauen Beschreibung (z. B. durch Use Case-Diagramme (vgl. [Sommerville 2011])) und Verifikation auf Richtigkeit und Realisierbarkeit der Anforderungen.

Zur Ermittlung und Analyse der Rahmenbedingungen bieten sich Techniken der Befragung (z. B. Experteninterviews mit Domänenexperten) und Techniken der Recherche (z. B.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Marktanalysen) an. Je nach Eigenschaften des Vorhabens (z. B. Neuentwicklung oder Altsystemanpassung, Kritikalität der Prozesse und Detaillierungsgrad der Anforderungen) können weitere Techniken wie bspw. Kreativitätstechniken, wie Brainstorming oder Mindmapping, in den Aufnahmeprozess einbezogen werden (vgl. [Rupp et al. 2009]).

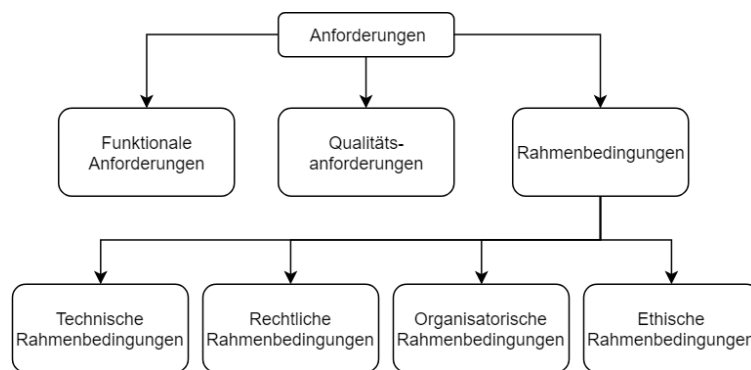


Abbildung 24 Gliederung Anforderungen (in Anlehnung an [Patig/Dibbern 2018])

Wie in Abbildung 24 ersichtlich und bereits erwähnt, sind vor allem Anforderungen im Hinblick auf technologische, organisatorische, ethische und rechtliche Rahmenbedingungen von Bedeutung.

Technologische Rahmenbedingungen an Anforderungen können Vorbedingungen und Einschränkungen durch Altsysteme sein, in die das neue System / Produkt integriert werden muss, bzw. Beschränkungen hinsichtlich der Datensicherheit und des Datenschutzes. Gerade im Bereich Datenschutz sind gemäß der Datenschutzgrundverordnung (DSGVO) und des Bundesdatenschutzgesetzes (BDSG) hohe Anforderungen zu erfüllen, um allgemein personenbezogene und medizinisch insistierte personenbezogene Daten zu verwalten und zu nutzen. Deshalb können in dieser ersten Analyse nur grundlegende Aspekte des Datenschutzes (evtl. anfallende Daten, Schutzbedürfnis dieser Daten, mögliche Analysen und deren Nutzung der Daten) aufgenommen werden, die in einer späteren Aktivität der Entwurfsphase ausgearbeitet und verfeinert werden müssen und ggf. die Anforderungen noch einmal beeinflussen (siehe Absatz 6.2.2.1).

Organisatorische Rahmenbedingungen beziehen sich auf den späteren Einsatz des Systems / Produktes im Unternehmen und dessen Integration in bestehende organisatorische und prozessuale Strukturen. Hierfür müssen bestehende Prozesse, die durch das System beeinflusst werden könnten, identifiziert und danach bewertet werden, ob diese angepasst werden können oder ob deren Ablauf, bspw. rechtlich, vorgegeben ist (z. B. Pflegeprozesse in der Altenpflege). Des Weiteren ist es erforderlich zu analysieren, ob bei oder nach der Einführung der Lösung evtl. neue Stellen geschaffen bzw. Aufgabenverteilungen auf

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

bestehende Stellen durchgeführt werden müssen. Diese Anforderungen werden später vor allem bei der Definition der Transformationsstrategie mitberücksichtigt (siehe Absatz 6.2.2.4).

Ethische Rahmenbedingungen auf Anforderungen beziehen sich sowohl auf die Akzeptanz der späteren Nutzer des Systems als auch auf die Fragestellung, ob ein Ziel des Systems eine forschende Tätigkeit sein soll. Unter dem Aspekt der Akzeptanz können Befragungstechniken wie Interviews genutzt werden, um mögliche Diskrepanzen der späteren Nutzer des Systems mit den definierten Ideen und Zielen zu identifizieren. Diese Rahmenbedingungen können später bei der Definition der funktionalen Anforderungen und der Qualitätsanforderungen mit einfließen. Sollen im Projekt forschende Tätigkeiten, wie bspw. bei der Algorithmenentwicklung, durchgeführt werden, kann der Projektträger die moralische Vertretbarkeit des Systems gegenüber dem Patienten von einer Ethikkommission im Verlauf des Projektes prüfen lassen. Die benannte Kommission kann von einer Universität bzw. einer Berufsvereinigung kommen oder aus einem unternehmenseigenen Gremium bzw. einem externen Ethiker bestehen.

Grundsätzlich besteht kein rechtlicher Zwang für den Einsatz einer Ethikbewertung (vgl. [Bundesministerium der Justiz und für Verbraucherschutz 2017]). Lediglich bei Medizinprodukten und Forschungsvorhaben müssen Ethikkommissionen befragt werden. Da in dieser Methode jedoch ausschließlich nicht-Medizinprodukte betrachtet werden, also Systeme, die zur Steigerung der Therapietreue genutzt werden, die jedoch nicht zur Diagnose, Verhütung, Überwachung, Vorhersage, Prognose, Behandlung oder Linderungen von Krankheiten oder Behinderungen dienen, ist dies nicht notwendig (vgl. [European Union 2017]). Der Einbezug von Medizinprodukten würde weitere Aktivitäten, hinsichtlich Zertifizierungseinschätzungen und den Einbezug von einer benannten Stelle, nach sich ziehen, welche nicht Bestandteil dieser Methode und Arbeit sind.

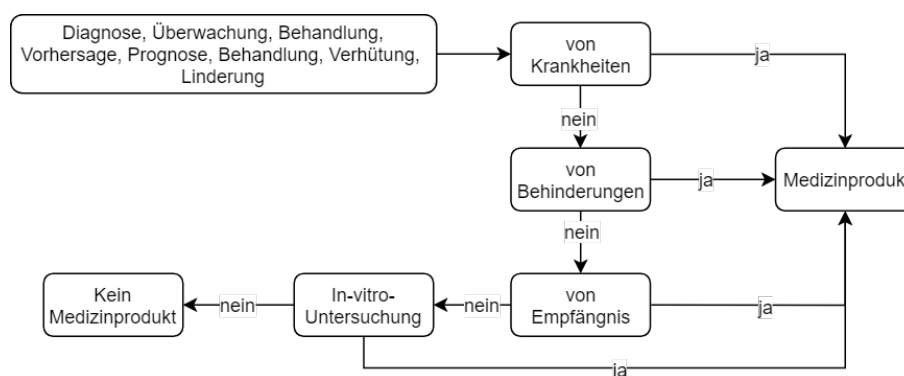


Abbildung 25 Definition Medizinprodukt

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Unter diesem Aspekt der Einordnung des Zielsystems als nicht-Medizinprodukt, muss mithilfe von Recherche- und Befragungstechniken, u. a. auch Beratungsgespräche bei Zertifizierungsstellen, geklärt werden, ob das zu entwickelnde Produkt / System als Produkt des Lebensstils und Wohlbefindens eingeschätzt werden kann oder ob es rechtlich betrachtet unter eine Medizinproduktklassifikation fällt. Abbildung 25 bietet einen allgemeinen Überblick über die Einordnung von Produkten im Medizinbereich, gemäß der EU Richtlinie *2017/746 Medical Device Regulation* (vgl. [European Union 2017]).

Weitere rechtliche Rahmenbedingungen können u. a. aus dem öffentlichen Raum (bei Produkten / Systemen, die öffentliche Installationen benötigen bzw. modifizieren), aus arbeitsrechtlichen Beschränkungen, sozial- bzw. pflegegesetzlichen Vorgaben hinsichtlich der Patienten sowie allgemeinen gesetzlichen Rahmenbedingungen, wie z. B. Einschränkungen bei der Überwachung von Patienten, kommen. Auch rechtliche Einschränkungen hinsichtlich des Einsatzes von lernenden Algorithmen, im Zusammenhang mit Entscheidungsunterstützung, kann von Bedeutung sein (vgl. [Müller-Hengstenberg, 233ff.]). Bei komplexen rechtlichen Fragestellungen bietet es sich an, einen Juristen (intern oder extern) zur Beantwortung der Fragestellungen mit einzubeziehen.

Als Ergebnis dieser Aktivität können Anwendungsfall-Diagramme stehen sowie eine Liste an Rahmenbedingungen an die Anforderungen mit Beschreibung dieser.

6.2.1.4 Analyse Therapie & Adhärenz

Zusätzlich zu den Rahmenbedingungen und Tagesabläufen sind für die Konzeption und Entwicklung des Zielsystems medizinisch relevante Anforderungen hinsichtlich der Patientenadhärenz und der damit verbundenen Therapietreue nötig. Das Ziel dieser Aktivität ist die Erkenntnis, wie es um die Therapietreue der ausgewählten Zielgruppe steht und welche medizinischen Rahmenbedingungen für das Zielsystem existieren. Diese Rahmenbedingungen können z. B. feste zeitliche Abläufe im Tagesablauf durch ein strenges Medikamentenregime oder die tägliche Bindung an eine Behandlung sein. Weiterhin ist es möglich, mit dieser Aktivität, weitere Faktoren für eine schlechte Therapietreue zu identifizieren. Alle identifizierten Rahmenbedingungen und Erkenntnisse können später bei der Definition der Transformationsstrategie genutzt werden, um Maßnahmen zu bewerten und selektieren zu können, die möglicherweise die Therapietreue steigern können.

Um eine systematische Bewertung der Adhärenz zu ermöglichen, bieten sich u. a. Befragungstechniken wie Experteninterviews mit dem betreuenden Personal und vor allem

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

mit den Patienten an (vgl. [Schäfer 2017]). In diesen Befragungen können vom Patienten empfundene Missstände hinsichtlich seiner Therapieabläufe und seine Grundhaltung gegenüber dieser identifiziert werden (siehe Adhärenz / Non-Adhärenz Vergleich in Unterabschnitt 3). Ebenso können Techniken der Recherche, wie bspw. der systematischen Literaturrecherche, genutzt werden, um aktuelle wissenschaftliche Erkenntnisse über die Adhärenz bei Patienten der ausgewählten Zielgruppe zu erhalten (vgl. [Vom Brocke et al. 2009; Feldmann/Gorj 2017]).

Mithilfe der Gliederung der Adhärenz-Faktoren der World Health Organisation (WHO) können Schwerpunkte bei den Befragungen gesetzt werden, um einen möglichst vollständigen Überblick über die Therapietreue des Patienten zu erhalten (vgl. [Weltgesundheitsorganisation 2003]). Parallel zur Konzeption der Interview Inhalte sollte abgegrenzt werden, welche Therapieabläufe genau betrachtet werden sollen, da die Adhärenz auch maßgeblich von der angewandten Therapie und der Schwere sowie der Dauer der Krankheit abhängig ist (vgl. [Schäfer 2017] & siehe Unterabschnitt 3).

Domänenexperten wie Therapeuten, Pfleger und Ärzte können die durch die Befragungen erhobenen Erkenntnisse auswerten und eine Abschätzung der für Therapietreue geben. Für die Betrachtung kann es je nach Krankheit und gewählter Therapie sein, dass die Therapietreue in einem akzeptablen bzw. guten Level liegt (vgl. [Schäfer 2017]), wodurch eine weitere Betrachtung dieser Therapie / Krankheit in der gewählten Patientengruppe nicht mehr nötig ist. In diesem Fall ist zu entscheiden, ob eine erneute Selektion der Patientengruppe nötig ist, was zu einem erneuten durchführen der Analyse des Patientenalltags und der Analyse der Patientenadhärenz führen würde oder ob trotz guter Adhärenz-Werte das System trotzdem mit entsprechender Unterstützung dieser Analysen umgesetzt wird.

Als Ergebnis dieser Aktivität steht eine medizinische Anforderungsliste, die Informationen über die Patientenadhärenz, wie nach der WHO Gliederung segmentiert sein kann, Informationen über den Ablauf der Therapie und mögliche Einschränkungen der Patienten durch die Therapieabläufe beinhaltet. Mithilfe dieser Anforderungsliste kann eine Auswahl der zu unterstützenden Alltagsbeschränkungen erstellt werden, die ebenfalls als Teil des Ergebnisses zu sehen ist.

6.2.1.5 Machbarkeit analysieren

Sind die möglichen Rahmenbedingungen aufgenommen, ist vor der Detaillierung der Anforderungen eine Machbarkeitsstudie durchzuführen. Diese soll die Durchführbarkeit hinsichtlich der Zielsetzung und Rahmenbedingungen bewerten und Empfehlungen über das weitere Vorgehen abgeben (vgl. [Wieczorrek/Mertens 2007; Gudehus 2011; Felkai/Beiderwieden 2015]). Die Begründung für die Notwendigkeit einer Machbarkeitsstudie gibt das hohe Risikopotential, wie z. B. die rechtlichen Aspekte und organisatorischen Einschränkungen, der Neuartigkeitscharakter bei Projekten oder die Unsicherheit bei der Umsetzung der algorithmischen Anforderungen (siehe Abschnitt 6.3), das durch das Projektvorhaben gegeben ist.

Bei der Durchführung der Machbarkeitsstudie sind die technische Machbarkeit, die wirtschaftliche und organisatorische Machbarkeit sowie eine Risikoeinschätzung zu bewerten. Dabei kommen Techniken wie Interviews, Recherchen und Umfeldanalysen zum Tragen.

Während der Prüfung der technischen Machbarkeit muss geklärt werden, ob die Anforderungen und Rahmenbedingungen aus informationstechnischer Sicht realistisch und im Rahmen des Vorhabens (Zeit, Kosten und Qualität) umsetzbar sind. Eine Sicherstellung dieser Bewertung kann durch IT-Fachpersonal (z. B. die künftigen Entwickler) gewährleistet werden, indem diese konzeptuell ein mögliches Zielsystem unter den Anforderungen und Rahmenbedingungen durchspielen und analysieren. Im Vordergrund steht das Zusammenspiel aller möglichen Komponenten und nicht das Design eines detaillierten Zielsystems. Weiterhin kann durch Domänenexperten die Akzeptanz der möglichen technischen Endnutzerschnittstellen (z. B. Smartphones oder Wearables) gegenüber den Patienten geprüft und bewertet werden, ob die Nutzung überhaupt realistisch ist.

Bei der wirtschaftlichen und organisatorischen Machbarkeit gilt es zu bewerten, ob das Kosten-Nutzen-Verhältnis des Projektes stimmt und ob das Projekt unter den gegebenen Anforderungen, Rahmenbedingungen und Ressourcen (Personal, Zeit, Finanzen) realistisch umgesetzt werden kann. Des Weiteren ist zu prüfen, ob eine organisatorische Integration der Ziellösung in die bestehenden Prozesse und Strukturen überhaupt möglich ist. Diese Bewertungen können vom Projektleiter, zusammen mit Entwicklern und relevanten Stakeholdern, getroffen werden.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Als letzter Teil der Machbarkeitsstudie steht eine Risikoeinschätzung, die Chancen und Risiken identifiziert und bewertet. Dabei spielen vor allem auch rechtliche Einschränkungen für die Umsetzbarkeit des Projektes eine Rolle. Weiterhin ist der realistische medizinische Nutzen des Zielsystems, und die damit verbundenen Risiken, durch die Domänenexperten zu bewerten. Als Technik für die Umsetzung dieser Risikoeinschätzung bietet sich eine SWOT-Analyse an (vgl. [Kotler/Keller 2012]). Mit dieser Analysemethode können alle Chancen und Risiken in eine Matrix übernommen werden, um mithilfe der daraufhin entwickelten Strategie über die Machbarkeit valide entscheiden zu können.

In einer zusammengefassten Betrachtung aller Machbarkeitsbetrachtungen obliegt es dem Projektleiter, eine realistische Empfehlung über die Durchführbarkeit zu erstellen und diese den Stakeholdern vorzustellen. Im Ergebnis entstehen dabei eine SWOT-Analyse, als Risikoeinschätzung, eine Machbarkeitseinschätzung aller genannten Gesichtspunkte und eine Entscheidungsempfehlung über den weiteren Verlauf des Projektes oder dessen Abbruch bzw. Neuausrichtung.

6.2.2 Entwurfsphase

In der Entwurfsphase werden die Anforderungen, gemäß der in der Analysephase identifizierten Rahmenbedingungen und Ist-Zustände, definiert. Um das zu entwickelnde System vollumfänglich definieren zu können, müssen eine Transformationsstrategie für die Einbindung des Systems in die bestehende Prozess- und Systemlandschaft sowie für die Erhöhung der Adhärenz, die Datenschutzanforderungen im Projekt, die funktionalen und nicht-funktionalen Anforderungen, die Gesamtarchitektur, die Hard- und Software und die zur Messung des Erfolgsgrades verwendeten Metriken definiert werden. Daraus ergibt sich das Ziel der Entwurfsphase, ein Gesamtkonzept für das zu entwickelnde System zu entwickeln, welches alle in der Analysephase identifizierten Informationen und Rahmenbedingungen einbezieht und ein umsetzbares Zielsystem definiert, welches die selbstgewählten messbaren Ziele erreichen kann.

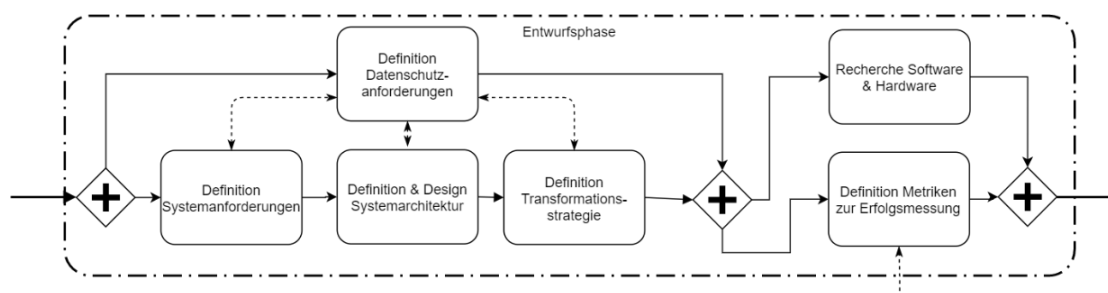


Abbildung 26 Prozessschritte der "Entwurfsphase" im Gesamtprozess

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Wie in Abbildung 26 zu sehen ist, werden in der ersten Aktivität die Systemanforderungen für die funktionalen und nicht-funktionalen Anforderungen definiert. Diese bilden die Grundlage für die Erstellung der Systemarchitektur und den damit verbundenen Software- und Hardware-Komponenten. Mit der Definition der Transformationsstrategie wird ein Konzept vorgegeben, das die Integration der Anforderungen in die bestehenden Strukturen des Gesamtsystems beschreibt. Parallel zu diesen Schritten muss ein Datenschutzkonzept erarbeitet werden, was alle nötigen rechtlichen Aspekte umfasst und ggf. bei der Erstellung die parallel ablaufenden Schritte inhaltlich beeinflusst. Daraufhin können konkrete Hard- und Softwarelösungen recherchiert, bewertet und ausgewählt werden, welche in der Gesamtarchitektur eingesetzt werden sollen. Parallel dazu werden die Metriken definiert, die die Zielerreichung des Systems und des Projektes aus wirtschaftlicher und medizinischer Sicht messbar machen sollen.

Folgende Abbildung 27 fasst die Ergebnisse, Techniken und federführenden Rollen der einzelnen Aktivitäten der Phase zusammen.

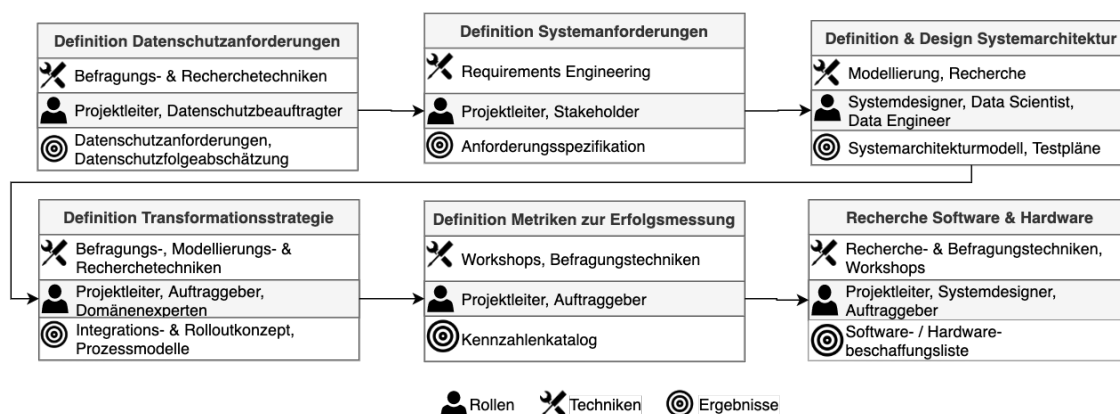


Abbildung 27 Übersicht Aktivitäten für Entwurfsphase

Alle Aktivitäten der beschriebenen Phase werden nachfolgend detaillierter definiert und beschrieben.

6.2.2.1 Definition Datenschutzanforderungen

Ziel dieser Aktivität ist die Definition der Datenschutzanforderungen für das Zielsystem. Weiterhin können die gewonnenen Informationen bei der Definition der Anforderungen für die Integration und das spätere Betreiben des Zielsystems genutzt werden. Da die Datenschutzanforderungen in paralleler Bearbeitung durch bspw. einen Datenschutzbeauftragten aufgenommen, jedoch nicht isoliert von den Systemanforderungen definiert werden können, müssen zwischen dieser und den Aktivitäten aus den Absätzen aus 6.2.2.4, 6.2.2.2 und 6.2.2.3 regelmäßig Absprachen geschehen.

Zur Bewertung des Datenschutzes können Techniken der Befragung und der Recherche angewandt werden. Sofern möglich, sollte ein Datenschutzbeauftragter dem Projektleiter beratend zur Seite stehen bzw. die Anforderungen selbst aufnehmen. Hierfür müssen gesetzgebende Dokumente auf nationaler und internationaler Ebene mit einbezogen werden (z. B. die Datenschutz-Grundverordnung [DSGVO] und das Bundesdatenschutzgesetz [BDSG]). In einem ersten Schritt wird unter Einbezug der zuvor definierten Anforderungen und Anwendungsfälle abgeklärt, ob personenbezogene Daten überhaupt anfallen bzw. verarbeitet werden. Dies sind alle Daten, die individuelle Person identifizieren können. Darunter gelten auch Daten, die indirekt Rückschlüsse zulassen, wie z. B. Bewegungsprofile. Fallen personenbezogene Daten an, muss das Schutzbedürfnis über diese Daten geklärt werden. Dies kann von nationalen und internationalen Richtlinien abhängig sein, ab wann Daten einen hohen Schutz benötigen. Im Rahmen der DSGVO ist zusätzlich zu beachten, dass bei Anwendung neuartiger Technologien die personenbezogenen Daten verarbeiten, wie bspw. Machine Learning, grundsätzlich eine Datenschutzfolgeabschätzung durchgeführt werden muss. Im Gegensatz zur restlichen Definition der Datenschutzerfordernung muss hier der Rat eines Datenschutzbeauftragten eingeholt werden. Es besteht hier zwingend eine Beratungspflicht (vgl. [Bundesministerium der Justiz und für Verbraucherschutz 2018]). Die Durchführung einer solchen Abschätzung ist durch die DSGVO und deren Auslegungen und evtl. Gerichtsentscheidungen geregelt (vgl. [Bitkom e. V. 2016; Bundesministerium der Justiz und für Verbraucherschutz 2018; European Union 2016]).

Das Ergebnis dieser Aktivität sind die Erfassung der Datenschutzerfordernungen an das Zielsystem sowie deren spätere Verwendung im System. Bei Anwendung einer Datenschutzfolgeabschätzung ist diese ebenfalls Teil des Ergebnisses.

6.2.2.2 Definition Systemanforderungen

In der Aktivität der Definition der Systemanforderungen steht als Ziel die Strukturierung, die Konkretisierung und die Ausformulierung der Anforderungen für das Zielsystem. Zur Realisierung der strukturierten Definition der Anforderungen kann die Requirements Engineering Methode genutzt werden (vgl. [Sommerville 2011, 99f.]).

Die in den vorhergehenden Aktivitäten definierten Anforderungslisten und grafischen Darstellungen werden für die Ausführung dieser Aktivität herangezogen. Dies inkludiert die technischen, organisatorischen, rechtlichen und medizinischen Anforderungen und Rahmenbedingungen, die vorab identifiziert und definiert wurden. Ebenfalls fließen in die

Erarbeitung der Anforderungen die Datenschutzanforderungen ein, welche parallel zu dieser Aktivität erarbeitet werden. In dem möglichen Fall, dass mehrere zu unterstützenden Szenarien aus dem Prozessschritt Analyse Patienten Adhärenz entstanden sind, wären diese spätestens in dieser Aktivität nochmals zu selektieren bzw. alle mit einzubinden, da die Systemanforderungen und die spätere Systemarchitektur maßgeblich von den zu unterstützenden Aktivitäten abhängig sind. Dieser Vorgang der Strukturierung der funktionalen und nicht-funktionalen Anforderungen obliegt dem Projektleiter, ggf. in Rücksprache mit verschiedenen Stakeholdern, wie Entwicklern oder Auftraggebern.

Nach Erstellung der Systemanforderungen liegt eine Anforderungsspezifikation mit funktionalen und nicht-funktionalen Anforderungen vor.

6.2.2.3 Definition & Design Systemarchitektur

Nachdem die Anforderungsspezifikation erstellt wurde, können die konkreten Anforderungen in eine Systemarchitektur überführt werden. Dabei beschreibt die Systemarchitektur alle Software- und Hardwarekomponenten, Schnittstellen innerhalb des Systems und zu bestehenden IT-Systemen, die zu verwendenden Architekturprinzipien, auszutauschende Daten und Datenmodelle sowie alle nicht-funktionalen Anforderungen wie Sicherheitskonzepte oder Anforderungen an die Benutzbarkeit des Systems (vgl. [Sommerville 2011, 151]). Weiterhin werden für die Systemarchitektur Testpläne für Integrations- und Softwaretests erstellt (vgl. [Beauftragter der Bundesregierung für die Informationstechnik 2009, 146ff.]). Die vorgesehenen Software- und Hardwarekomponenten sind in dieser Architektur noch abstrakt definiert und nicht mit konkreten Entwicklungen oder Produkten hinterlegt. Bei der abstrakten Definition der Hardware Komponenten ist vor allem auf den Einbezug der eventuellen Einschränkungen der späteren Zielgruppe und auf das Interaktionsdesign zu achten. Die Auswahl dieser Komponenten im Rahmen einer Make-or-Buy Entscheidung findet in der Aktivität der *Recherche & Auswahl Hardware / Software* (siehe Absatz 6.2.2.6) statt.

Die Erstellung der Systemarchitektur obliegt dem Systemarchitekten unter Einbezug des Data Engineers sowie des Data Scientists. Diese nutzen Techniken der Modellierung zur strukturellen Darstellung und Techniken der Recherche für die Anwendung aktueller System- (z. B. Client-Server-Architektur) und Softwarearchitekturprinzipien (z. B. monolithische oder Microservice-Architektur). Die Auswahl der Architekturprinzipien ist anwendungsfallabhängig und muss Vor- und Nachteile der Prinzipien in die Designauswahl mit einbeziehen (vgl. [Richardson 2017, 40ff.] & [Sommerville 2011, 151]).

Als Ergebnis dieser Aktivität steht ein Systemarchitekturmodell, inklusive der Software- und Hardwarearchitektur und der dazugehörigen Integrations- und Softwaretestpläne.

6.2.2.4 Definition Transformationsstrategie

Nach der Fertigstellung der Anforderungen und der Systemarchitektur müssen diese für den späteren Gebrauch im Unternehmen in organisatorisch-strukturelle und technische Begebenheiten integriert werden. Um dies zu erreichen, muss eine Transformationsstrategie erstellt werden. Diese hat zum Ziel, ein Konzept zur Anpassung von Geschäftsprozessen zur Einbindung der späteren Nutzer des Zielsystems und zur Integration des Systems in die bestehenden Alt-Systeme zu erstellen.

Aus technischer Sicht betrachtet ist es wichtig, das Zielsystem ohne Medienbrüche in die Alt-System-Landschaft zu integrieren. Um dies zu ermöglichen, müssen die technischen Schnittstellen zwischen den IT-Systemen identifiziert und spezifiziert werden. Schnittstellen, abgehend vom Zielsystem, wurden bereits in der vorangegangenen Aktivität definiert. Jedoch ist nicht immer sichergestellt, dass die entsprechenden Schnittstellen bei den Alt-Systemen vorhanden sind. Diese müssen ggf. neu spezifiziert und implementiert werden, um überhaupt einen Datenaustausch zwischen Alt- und Zielsystem durchzuführen. Im gleichen Zuge ist auch das Datenaustauschformat ein wichtiger Bestandteil der Betrachtung. Fallen Daten in mehreren Systemen an, ist sowohl über eine einheitliche Datenstruktur als auch über Datenkonsistenzen nachzudenken, wenn die Daten nicht in einer Single-Source of Truth liegen (vgl. [Stimmel 2015, 26]).

Aus organisatorischer Sicht ist vor allem die prozessuale Integration relevant. Hier ist zu überlegen, wer bzw. welche Abteilung später für das System verantwortlich sein soll. Eventuell müssen hierfür neue Stellen geschaffen werden. Prozesse, wie sie bereits bei der Analyse der Rahmenbedingungen (siehe Absatz 6.2.1.3) grob identifiziert wurden, müssen detailliert betrachtet und für die Integration des Zielsystems und dessen Prozesse angepasst werden. Um diese neuen und modifizierten Prozesse sowie die technischen Applikationen auszurollen, braucht es einen geregelten Prozess des Rollouts. Hierfür bietet sich die Erstellung einer Rollout-Strategie an. Diese beschreibt, wo, in welcher Reihenfolge und in welcher Form das Zielsystem mitsamt den modifizierten und neuen Prozessen im Unternehmen implementiert wird (vgl. [Koch 2011, 107] & [Gadatsch 2010, 360f.]).

Unter der Berücksichtigung der Bedienungspraktikabilität und Technikakzeptanz sind die Nutzer des Zielsystems von Bedeutung. Dabei liegt der Fokus auf den Interaktionsprozessen.

Zu diesen zählen die Benutzerführungen und die Benutzeroberflächen sowie die Gesprächsabläufe für bspw. Sprach- bzw. Texteingaben. Eine Erarbeitung der Anforderungen dieser Betrachtung muss sowohl mit systembetreuenden Mitarbeitern als auch mit den zukünftigen Endnutzern erfolgen.

Durchgeführt werden kann dieser Schritt vom Projektleiter sowie den Auftraggebern, die das System einführen möchten inkl. deren Domänenexperten, die beratend zur Seite stehen können. Angewandte Techniken zur Umsetzung der Aktivität können Befragungs- und Recherchetechniken sowie Modellierungstechniken zur visuellen Darstellung des Transformationsprozesses sein.

Resultat und Artefakt dieser Aktivität ist ein Integrations- und Rolloutkonzept sowie eventuelle Prozessmodelle für neue bzw. bestehende aber modifizierte Abläufe im Unternehmen.

6.2.2.5 Definition Metriken zur Erfolgsmessung

Parallel zur Auswahl von Hardware und Software müssen zur Erfolgsmessung des Projektes und Vorhabens Kennzahlen definiert werden. Diese geben Aufschluss über die erfolgreiche Beendigung des Projektes und die Erreichung der Ziele. Dabei sollten sich diese an den für das Projekt definierten Zielen und Anforderungen aus den vorangegangenen Aktivitäten orientieren und messbar, objektiv, reproduzierbar und beeinflussbar operationalisiert sein (vgl. [Fittkau/Ruf 2008, 211ff.] & [Karlstedt 2014, 24ff.]). Die erforderlichen Kennzahlen können vom Projektleiter, unter eventueller Absprache mit dem Auftraggeber, definiert und durch Befragungen oder Workshops identifiziert werden.

Bei der Definition von Kennzahlen für das Vorhaben geht es nicht nur um die Einhaltung des terminlichen und finanziellen Rahmens des Projektes, sondern ebenfalls um Qualitätsbetrachtungen. Können Kennzahlen z. B. die qualitative Messung der Erhöhung der Adhärenz im erfolgreichen Feldtest des Prototyps oder wirtschaftliche Einsparungen, bspw. in der zeitlichen und finanziellen Bindung von Sachgütern oder Personal, sein.

Das Ergebnis dieser Aktivität ist eine Liste an Schlüsselkennzahlen für die Erfolgsmessung des Projektes. Für die Messung von technischen und technisch-qualitativen Kriterien werden in der Aktivität unter Absatz 6.3.1.3 im späteren Verlauf des Vorgehens weitere Kennzahlen definiert, um zusätzlich zum wirtschaftlichen Hintergrund auch technische Aspekte mit einzubeziehen.

6.2.2.6 Recherche & Auswahl Hardware / Software

Anhand der Systemarchitektur, welche die abstrakten Komponenten und Eigenschaften für die Hardware und Software Komponenten beschreibt, können in diesem Schritt die entsprechenden Komponenten konkret für das Projekt beschafft bzw. umgesetzt werden. Ziel ist somit die Auswahl und Beschaffung bzw. Umsetzung der benötigten Komponenten für sowohl Software als auch Hardware.

Bei der Auswahl der Software- bzw. Hardware-Komponenten geht es im ersten Schritt um die Entscheidung, ob die benötigten Komponenten gekauft bzw. auf frei verfügbare zurückgegriffen werden kann oder diese selbst entwickelt werden sollen. Im Folgenden werden Hardware und Software separat betrachtet, da diese unterschiedliche Anforderungen und Vorgehensweisen erfordern.

Bei der Entscheidung, fertige Hardware-Komponenten zu beschaffen oder diese selbst zu entwickeln, spielen verschiedene Faktoren eine Rolle. Zum einen gibt die Budgetierung den Rahmen vor, ob im Projektvorhaben die finanziellen Mittel für solch eine Entwicklung vorhanden sind. Zum anderen ist relevant, ob das benötigte Knowhow und die benötigten zeitlichen und personellen Ressourcen vorhanden sind, um eine solche Entwicklung während des Projektes durchzuführen (vgl. [Angermeier 2009]). Zudem erwächst aus der Eigenentwicklung von Hardware-Komponenten auch ein gewisses Fehlschlagsrisiko, das schwer kalkulierbar ist. Risiken können dabei schlechtes Knowhow, nicht bestandene Zertifizierungsanstrengungen, fehlende Produzenten etc. sein. Daher sollte grundsätzlich beim Vorhandensein von kommerziellen, serienreifen Produkten auf dem Markt die Kaufentscheidung der Eigenentwicklung vorgezogen werden.

Da die Hardware-Entwicklung den fortlaufenden Prozess verändern und weitere Schritte und Artefakte generieren würde, wie bspw. Zertifizierungsschritte für die Verkehrsfähigkeit der Hardware-Komponenten (z. B. CE-Zertifizierungen) (vgl. [Bundesministerium der Justiz und für Verbraucherschutz 2016]), und der Fokus dieser Methode softwaregetrieben auf der Entwicklung sicherer präskriptiver Algorithmen in Therapieunterstützungssystemen liegt, soll die Eigenentwicklung von Hardware folgend nicht mehr im Prozess betrachtet werden. Damit sind in diesem Vorgehensmodell alle Hardware-Komponenten extern bezogene Elemente.

Zum Vergleich verschiedener Hardware-Anbieter auf dem Markt hat sich als Technik die Durchführung einer Nutzwertanalyse bewährt (vgl. [Tiemeyer 2014, 398ff.]). Um diese

durchführen zu können, müssen Zielkriterien erstellt werden, die die Anbieter untereinander vergleichbar machen. Diese können vom Projektleiter zusammen mit dem Systemarchitekten, und ggf. weiteren mit der Hardware agierenden Personen, z. B. Domänenexperten als Vertretung für die späteren Nutzer des Systems, im Rahmen eines Workshops ausgewählt, gewichtet, bewertet und entschieden werden. Unter anderem bieten sich für die Erstellung einer solchen Hardware-Nutzwertanalyse Kriterien wie z. B. Einmal- und laufende Kosten, Bedienbarkeit, Erweiterbarkeit, vorhandene Programmierschnittstellen, zukünftiger Support und Weiterentwicklung etc. an.

Für die Auswahl der Software-Komponenten kann ähnlich wie bei der Selektion der Hardware-Komponenten vorgegangen werden. Neben den klassischen Kaufentscheidungen kann bei Software jedoch auch auf quelloffene Entwicklungen zurückgegriffen werden. Hierbei sind serienreife Produkte Eigenentwicklungen generell vorzuziehen, sofern diese die benötigten Eigenschaften besitzen. Auch in diesem Rahmen bietet sich für die Bewertung aller kommerziellen bzw. quelloffenen Lösungen eine Nutzwertanalyse an. Auch hier kann diese wieder im Rahmen eines Workshops erarbeitet und entschieden werden. Mögliche Kriterien für die Software-Auswahl können z. B. Preis, Folgekosten, Anbieter, Produktreife, Community-Unterstützung, Weiterentwicklung, Support, Dokumentation, Programmierschnittstellen, Integrationsfähigkeit und Bedienbarkeit sein.

Sollte bei der Softwareauswahl trotz Nutzwertanalyse am Ende ein nicht zufriedenstellendes Ergebnis stehen, kann über eine Eigenentwicklung, unter Einbezug der oben genannten Kriterien, nachgedacht werden. Dem Projektleiter obliegt diese Aktivität, unter Einbezug von Auftraggeber und Systemdesigner.

Das Ergebnis dieser Aktivität sind eine Hardware- und Softwarebeschaffungsliste, welche als Grundlage für die Beschaffung der Komponenten dient.

6.2.3 Entwicklungs- & Testphase

In der Entwicklungs- und Testphase werden die Anforderungen gemäß dem Anforderungskatalog sowie die Gesamtarchitektur umgesetzt. Dabei wird die Hard- und Softwareentwicklung von der Algorithmenentwicklung unterschieden, da für algorithmenspezifische Entwicklungen differenziertere Vorgehen benötigt werden, als dies bei der Software oder Hardware Entwicklung üblich ist. Ziel dieser Phase ist die Entwicklung eines voll funktionalen Systems, welches unter Realbedingungen getestet wurde und reif für die Bereitstellung ist.

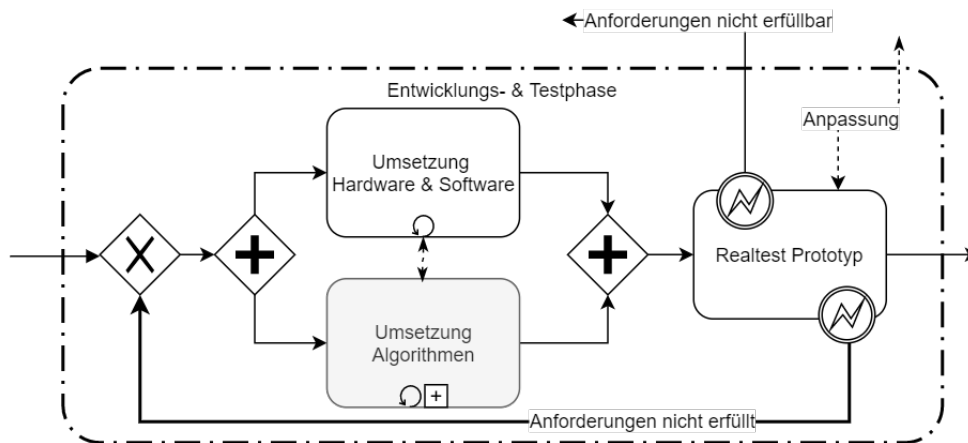


Abbildung 28 Prozessschritte der "Entwicklungs- und Testphase" im Gesamtprozess

Wie Abbildung 28 veranschaulicht, können beide Umsetzungsschritte parallel abgearbeitet werden. Da bei der Umsetzung beider Schritte eine starke wechselseitige Interdependenz besteht, muss dabei stets ein technischer Austausch stattfinden, um ein lauffähiges Gesamtprodukt erstellen zu können. Der dabei entstandene Prototyp wird daraufhin in einer Realtestphase auf die praktische Nutzbarkeit in einer realen Umgebung getestet. Sollten die Ergebnisse den gesetzten Zielen nicht entsprechen, muss in einem iterativen Prozess die Umsetzung weiter verfeinert und angepasst werden. Auch ein vollständiger Abbruch des Realtests, mit Rücksprung zur Analysephase, ist möglich, sollten Anforderungen aus bestimmten Gründen nicht umsetzbar sein.

In der folgenden Abbildung 29 werden die einzelnen Aktivitäten der Phase mit ihren Ergebnissen, Rollen und Techniken dargestellt.

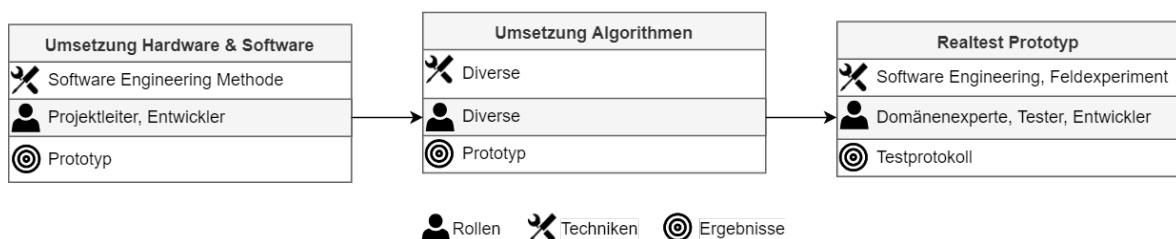


Abbildung 29 Übersicht Aktivitäten für Entwicklungs- & Testphase

Im folgenden Verlauf werden die in Abbildung 28 visualisierten Aktivitäten näher beschrieben.

6.2.3.1 Umsetzung Hardware & Software

Auf Basis des zuvor definierten Anforderungskatalogs und der entwickelten Systemarchitektur werden in diesem Schritt die Anforderungen umgesetzt. Ziel ist somit die Entwicklung und Konfiguration eines lauffähigen Prototyps, welcher alle Anforderungen

umsetzt und bereit für einen Test, unter realen Bedingungen, mit realen Testpersonen, im Alltag ist.

Zu Beginn der Aktivität sind die Infrastruktur und ggf. benötigte Softwareprodukte für die Entwicklung und das Testen der Software- und Hardware-Komponenten aufzusetzen, falls noch nicht vorhanden. Danach können die beschafften Hardware- und Software-Komponenten für das System angepasst werden. Software-Komponenten die gekauft bzw. bezogen und nicht selbst entwickelt werden, müssen auf die Bedürfnisse der Anforderungen angepasst und in die selbstentwickelte Software und ggf. in Alt-Systeme integriert werden. Hardware-Komponenten müssen in das System sowie die zu entwickelnde bzw. zu konfigurierende Software eingebunden und für die Verwendung entsprechend angepasst werden. Weiterhin müssen die Entwicklung und Konfiguration der Software- und Hardware-Komponenten mit der Entwicklung der Algorithmen abgestimmt und abgeglichen werden. Das bedeutet, dass die Algorithmen in die Software und ggf. auf die Hardware integriert werden können. Dies bezieht u. a. Rahmenbedingungen ein, die während der Algorithmenentwicklung definiert (bspw. Fallback-Strategien, siehe Absatz 6.3.2.3) und in die Entwicklung der Software-Komponenten mit eingebracht werden müssen.

Für die Entwicklung der Software bietet sich der Einsatz der Software-Engineering Methode (vgl. [Bourque/Fairley 2014, xxix]), unter Einsatz eines beliebigen Vorgehensmodells, an. Dabei muss das gewählte Vorgehensmodell lediglich die in den allgemeinen Anforderungen definierten Regeln einhalten (siehe Unterabschnitt 4.1.1). Unter diesen Bedingungen können dies bspw. Vorgehensmodelle wie das V-Modell XT (vgl. [Verein zur Weiterentwicklung des V-Modell XT e.V 2006]) oder Scrum (vgl. [Sommerville 2011, 72ff.] & [Gloger 2016]) bzw. beliebig andere iterative, rücksprungfähige Modelle sein. Zusätzlich zur Entwicklung sind Tests der Software- und Hardwarekomponenten erforderlich. Diese werden abhängig von der gewählten Methode, auf Basis der definieren Testfälle, durchgeführt und ggf. durch weitere Tests erweitert.

Die Umsetzung dieser Aktivität wird vom Projektleiter, bzw. Product Owner unter Ansätzen wie Scrum, verantwortet und den Entwicklern (SW-Entwickler, UX-Entwickler etc.) durchgeführt.

Das Ergebnis dieses Vorgehensschritts ist die physische Infrastruktur zum Betreiben der Lösung sowie ein lauffähiger und getesteter Prototyp, der die für einen Realtest nötigen Anforderungen erfüllt.

6.2.3.2 Umsetzung Algorithmen

Die Umsetzung der Algorithmen ist ein Subprozess im Gesamtprozess, der das Design und die Entwicklung von entscheidenden Algorithmen beschreibt. Damit verfolgt diese Aktivität das Ziel, die für das Vorhaben benötigten Algorithmen zu entwerfen, zu entwickeln und zu testen, um diese dann im Realtest, mit der aufgebauten Infrastruktur aus Absatz 6.2.3.1 zu testen.

In diesem Teilprozess werden alle Zielbestimmung und Metriken getroffen, die für die Umsetzung der Algorithmen nötig sind. Weiterhin werden die Daten identifiziert, selektiert und vorverarbeitet, die für die Algorithmen und die Zielerfüllung von Bedeutung sind, und in die Spezifikation des Algorithmus und der Simulationsumgebung eingebunden. Eine Enforcement- und Recovery Strategie stellt sicher, dass auch im Fehlerfall das System nicht schädlich für den Nutzer ist und funktionstüchtig bleibt. Daraufhin werden der Algorithmus und der Agent erstellt und getestet, sowohl auf Zielerreichung als auch auf Sicherheit, und für die Implementierung in die Softwarekomponenten für den späteren Realtest bereitgestellt.

Das Gesamtergebnis dieses Subprozesses ist der final parametrisierte und trainierte Agent für die Realtestphase. Die nähere Beschreibung aller Schritte in diesem Subprozess findet sich im Abschnitt 6.3.

6.2.3.3 Realtest Prototyp

Der Realtest, in der Soziologie / Psychologie auch Feldexperiment genannt (vgl. [Aronson et al. 2011, 46f.]), dient dem Testen des Prototyps unter realen Bedingungen, im realen Umfeld, mit realen Endnutzern. Dieser unterscheidet sich gegenüber den vorherigen Tests dadurch, dass diese i. d. R. unter Laborbedingungen durchgeführt wurden, also in kontrollierten Umgebungen. Der Realtest ermöglicht die unvoreingenommene und unverfälschte Erprobung des Systems mit allen Problemen, die im Alltag bei der Nutzung des Systems durch die Endnutzer oder durch Experten für das Endnutzerverhalten auftreten können.

Um alle grundlegenden Funktionalitäten zu testen, bietet es sich an, Szenarien zu erstellen, die eine gewisse Anzahl durchlaufen werden müssen. Je komplexer das System und die zugrundeliegenden Funktionalitäten und Algorithmen sind, desto länger und aufwändiger kann ein Realtest werden. Dabei sollte die Länge der Realtestphasen, auch im Hinblick auf die Verbesserung der Adhärenz, situativ angepasst werden. Das bedeutet, bei

Langzeitänderungen der Adhärenz aufgrund von langen Zyklen der Messung von Krankheits- / Verhaltensänderungen (bspw. bei langen Zyklen zwischen depressiven und „normalen“ Phasen, vgl. [Risch et al. 2012, 3f.]), muss die Realtestphase ebenfalls einen gewissen Zeitraum abdecken, um den Effekt des Systems auf das Verhalten der Endnutzer messen zu können.

Der Realtest kann beliebig oft im Projektvorhaben durchgeführt werden, z. B. zur Validierung von Meilensteinen bzw. Quality Gates, jedoch muss auf den hohen Aufwand an Vorbereitung für die Testszenarien, Bereitstellung des Systems, Bereitstellung an Zeit und personellen Ressourcen und die Auswertung der Resultate geachtet werden. Liegen die Testresultate außerhalb der erwarteten Grenzen für die Qualität des Systems, kann ein Rücksprung zur Entwicklung des Systems und somit zur Fehlerbehebung und Anpassung durchgeführt werden.

Ein besonderes Augenmerk liegt bei der Durchführung auf dem Zusammenspiel von Software- / Hardware-Komponenten und den Agenten / Algorithmen, die in diesem Test das erste Mal zusammen im Prozess erprobt werden. Unter diesem Gesichtspunkt sollten auch die Recovery- und Enforcement-Strategien getestet werden, um eine fehlerfreie Ausführung aller Komponenten auch im Fehlerfall unter Realbedingungen garantieren zu können.

Als Techniken für die Durchführung dieser Aktivität kommen Methoden des Software Engineering für das Testen von Software und der Soziologie im Rahmen der Feldexperimente (je nach Domänenanforderungen bei der Messung der Funktionalität) zur Anwendung (vgl. [Sommerville 2011, 228ff.] & [Aronson et al. 2011, 46f.]). Die Umsetzung erfolgt durch Tester bei der Vor- und Nachbereitung mithilfe der Entwickler, die das System bereitstellen und Domänenexperten, die im Feld die Anwendung testen.

Das Ergebnis dieser Aktivität ist ein Testprotokoll mit Testergebnissen über den Verlauf des Realtests.

6.2.4 Deployment- & Betriebsphase

Die Deployment- und Betriebsphase des Vorgehensmodells beschäftigen sich mit der Bereitstellung des validierten Prototyps sowie dem nachhaltigen Betrieb dieses Prototyps. Unabhängig voneinander müssen Schulungen durchgeführt sowie Monitoring Metriken und Maßnahmen definiert werden (siehe Abbildung 30). Dabei beschränken sich die Aktivitäten in dieser Phase auf die vorbereitenden Schritte für den Betrieb, jedoch nicht auf den Betrieb selbst (z. B. aktive Betreuung und Weiterentwicklung des Systems). Zielstellung der

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Deployment- und Betriebsphase ist die Befähigung des Auftraggebers, sowohl technisch als auch organisatorisch, die entwickelte Lösung in den laufenden Betrieb zu übernehmen und diese dann weiter zu betreiben.

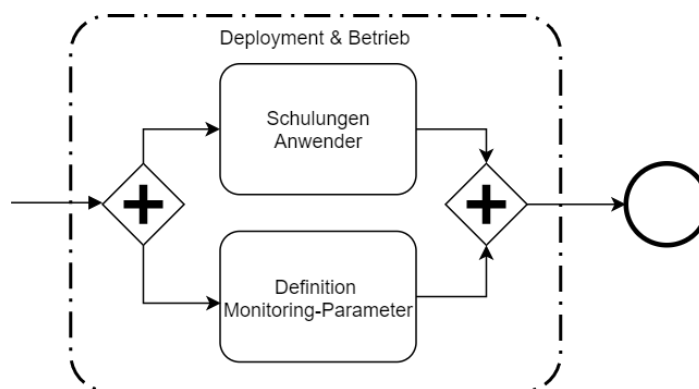


Abbildung 30 Prozessschritte der „Deployment- & Betriebsphase“ im Gesamtprozess

Zur Überwachung der fortwährenden Aufrechterhaltung der Qualität des Systems müssen zudem Monitoring Metriken und Maßnahmen ausgearbeitet werden, die während des Betriebes alle qualitätsrelevanten Funktionalitäten, vor allem die der Algorithmen aufgrund Änderungen in der realen Welt, überwachen. Damit das System auch eingesetzt werden kann, sind Schulungen für die Anwender, sowohl auf Patienten als auch auf Betreuer Seite, durchzuführen.

In der folgenden Abbildung 31 werden die Rollen, Techniken und Ergebnisse der in der Phase vorhandenen Aktivitäten abgebildet.

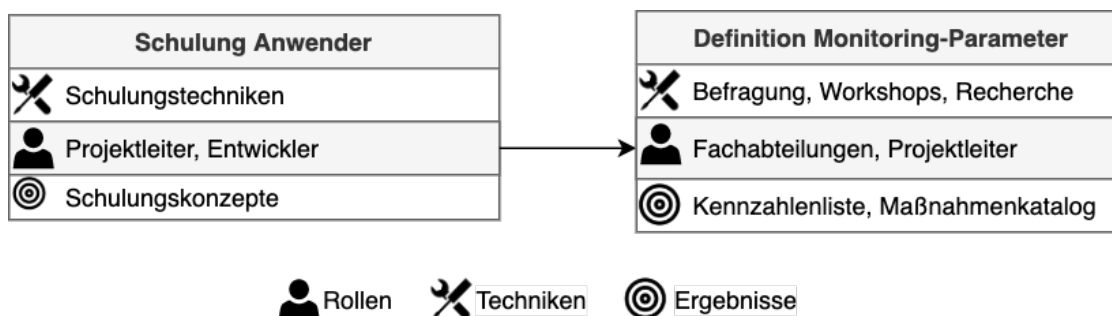


Abbildung 31 Übersicht Aktivitäten für Deployment & Betrieb

Eine detailliertere Ausführung der einzelnen Aktivitäten wird in den nächsten Absätzen beschrieben.

6.2.4.1 Schulung Anwender

Zur erfolgreichen Einführung eines Systems müssen Schulungen für die Betroffenen Mitarbeiter sowie die Endanwender durchgeführt werden. Ziel ist die Schulungen ist die

Befähigung der Endanwender die Technik effektiv zu nutzen sowie diese in die neuen bzw. modifizierten Prozessabläufe einzubringen (vgl. [Abts/Mülder 2009, 378]).

Zur Umsetzung müssen sowohl ein Schulungskonzept für die Anwender des Systems erstellt werden als auch ein Schulungskonzept für die betreuenden Mitarbeiter, die eventuelle administrative und / oder Monitoring Aufgaben übernehmen. Dabei müssen die Schulungsbedarfe festgelegt werden, welche sowohl Technologien und Software als auch Prozessabläufe inkludieren. Zur passenden Ausarbeitung des Umfangs des Schulungsbedarfs, muss der Stand des Wissens der Mitarbeiter und Endanwender erfasst werden. Daraufhin kann ein Schulungsplan erstellt werden, die die Anforderungen an die Schulung impliziert. Hierbei muss auch geklärt werden, ob eventuell für Teile der Schulung externe Anbieter herangezogen werden sollen, vor allem, wenn es um gekaufte Produkte (HW/SW) geht (vgl. [Goltsche 2006, 123ff.] & [Winkelhofer 1997, 459ff.]).

Bei der Festlegung der Schulungsmethoden, können sich diese, je nach zu schulender Anwendergruppe, unterscheiden. Deshalb, sind diese entsprechend auszuwählen und anzupassen. So ist das IT-Fachpersonal aus der IT-Abteilung, in die Software und Hardware, die Wartung und die Betreuung einzuweisen sowie in die eventuell dazu nötigen IT-Service-Operation Prozesse (vgl. [Beims/Ziegenbein 2015, 150ff.] & [Sommerville 2011, 720]). Endanwender wiederum müssen mit den technischen Hintergrundwissen nicht betraut werden, diese müssen mit den Hardware- und Softwareschnittstellen umgehen und diese in ihre alltäglichen Abläufe und Arbeitsprozesse integrieren können.

Durchgeführt wird diese Aktivität vom Projektleiter unter Beteiligung der Entwickler, welche das System designt haben und eventuelle hinzugezogene externe Experten für bestimmte Komponenten des Systems.

Die Ergebnisse dieser Aktivität sind angepasste Schulungskonzepte für die entsprechenden Anwendergruppen.

6.2.4.2 Definition Monitoring Metriken

Zum Betrieb des Systems müssen Metriken und Maßnahmen zum Messen der Qualität und Performance erstellt werden, um einen gleichbleibenden Service zu garantieren (vgl. [Sowa/Fedtke 2011, 4]). Dabei spielen vor allem Metriken hinsichtlich des IT-Systems, der Therapietreue und der eingesetzten Algorithmen eine Rolle. Ziel ist hier, die Erstellung eines Kennzahlensystems bzw. die Erstellung und Integration von neuen Kennzahlen in ein bestehendes Kennzahlensystem oder eine bestehende Ursache-Wirkungs-Kette für

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Kennzahlen, wie eine Balanced-Score-Card (vgl. [Hockenbrink 2012, 1ff.]), um ein gleichbleibendes Qualitätslevel zu garantieren und dieses ggf. an neue Rahmenbedingungen anzupassen (vgl. [Goltsche 2006, 144ff.]).

Die Definition solcher Metriken zur Qualitäts- und Performance-Messung des Systems kann in einem Workshop mit Fachabteilungen bzw. bei Experteninterviews mit dem Servicebetreuenden Personal durchgeführt werden. Ebenso können die zu definierenden Kennzahlen aus den Anforderungen und Zielen des Systems extrahiert und mit den Fachabteilungen abgeklärt werden.

Weiterhin sind für die Metrikdefinitionen eventuelle rechtliche Vorschriften, wie z. B. der Datenschutz, einzubeziehen, welche fortwährend analysiert und ggf. angepasst werden müssen (vgl. [Goltsche 2006, 150ff.]). Eine solche kontinuierliche Anpassung der Kennzahlen und Qualitätskriterien kann unter Zuhilfenahme von Normen wie z. B. der ISO/IEC 27001 geschehen, die durch Maßnahmen wie wiederkehrende Audits durch Externe die Qualität der IT-Systeme und Prozesse zertifizieren und vor dem Gesetzgeber absegnen (vgl. [Sowa/Fedtke 2011, 26ff.]).

Ergebnis ist eine Liste an Kennzahlen zur Qualitätssicherung für das Gesamtsystem sowie ein Maßnahmenkatalog, welcher weiterführende Qualitätssicherungsprozess wie Audits oder Zertifizierungen umfasst.

6.3 Sub-Prozess Umsetzung Algorithmen

Für die Entwicklung von Algorithmen gibt es eine Reihe von allgemeinen Verfahren, die in ihrer Detaillierung nicht genau genug sind, um sichere präskriptive Algorithmen zu entwickeln. Der Sub-Prozess wurde deshalb definiert, weil die Anwendung bzw. Entwicklung von Algorithmen nicht in einer klassischen Software Entwicklung eingebettet werden kann (siehe Abschnitt 4.3). Deshalb wird in den folgenden Punkten ein Teilprozess zur Entwicklung von sicheren präskriptiven Algorithmen beschrieben, der Sicherheit sowohl explizit als auch messbar einbindet.

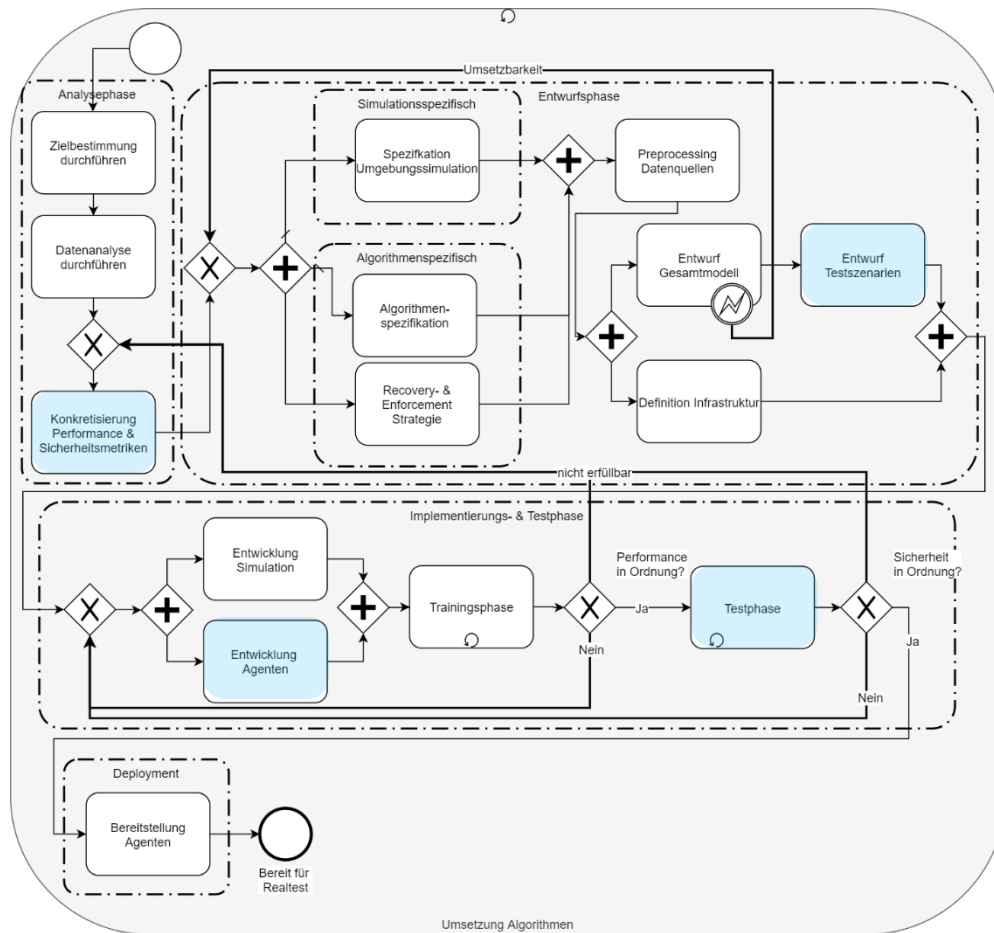


Abbildung 32 Funktionsbaum Teilprozess Algorithmenentwicklung

Die Abbildung 32 zeigt, dass der Teilprozess, ebenfalls wie der übergeordnete Hauptprozess, in einzelne Phasen gegliedert ist. Diese sind aus dem Hauptprozess entlehnt und orientieren sich an klassischen Softwareentwicklungsprozessen. Auch hier sind wie im Hauptprozess Rücksprünge und Iterationen möglich, um eine möglichst flexible Entwicklung zu gewährleisten. Somit können auch während des Entwicklungsprozesses die Anforderungen an das Design und die Infrastruktur mit der parallel ablaufenden Softwareentwicklung ausgetauscht werden, um später einen leicht integrierbaren Prototypen zu designen.

Manche Aktivitäten müssen nicht zwangsläufig bei jedem Projekt und Anwendungsfall durchlaufen werden. So kann bspw. die Simulationsumgebung bereits bestehen und muss lediglich für den Einsatz am Algorithmus konfiguriert, jedoch nicht neu entwickelt werden. Auch kann eine Infrastruktur für die Entwicklung eines Algorithmus bereits bestehen, weshalb dieser Schritt auch nicht unbedingt durchlaufen werden muss. Wann welche Schritte als optional betrachtet werden, ist vom jeweiligen Projektvorhaben und Anwendungsfall abhängig.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Die Besonderheit des Subprozesses, neben der seiner grundsätzlichen Definition innerhalb einer Software-Entwicklung, ist die explizite Betrachtung der Sicherheit bei der Entwicklung einer solchen Klasse von Algorithmen. Aufgrund der hohen Kritikalität beim Einsatz dieser Algorithmen in der realen Welt unter Einbezug von Menschen (z. B. vgl. [Welt Online 2020]), werden mehrere Schritte, verteilt auf mehrere Phasen des Teilprozesses, zur Absicherung definiert. Diese sind in der Abbildung 32 separat hervorgehoben.

In den folgenden Unterabschnitten werden die Phasen dieses Teilprozesses mit ihren Aktivitäten beschrieben.

6.3.1 Analysephase

Die Analysephase im Sub-Prozess der Algorithmenentwicklung betrachtet den bereits in Unterabschnitt 6.2.1 aufgenommen Ist-Zustand sowie die definierten Systemanforderungen aus Absatz 6.2.2.2 und nutzt diese Informationen, um für den zu entwickelnden Algorithmus eine Zielbestimmung, Datenanalyse und die Erstellung von Sicherheits- und Performancemetriken zu definieren. Ziel dieser Phase ist es, eine Aufnahme und Konkretisierung der Voraussetzungen und Rahmenbedingungen für die Algorithmendefinition und Algorithmenentwicklung durchzuführen.

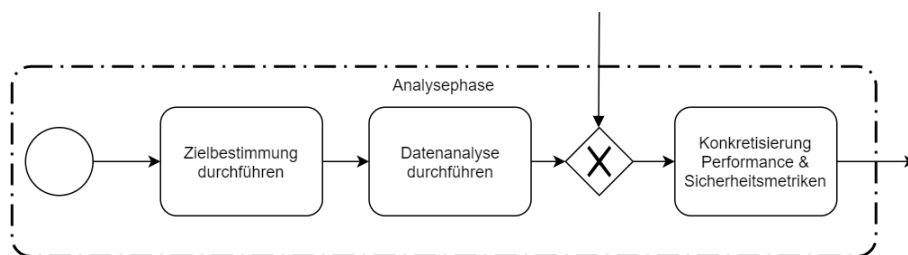


Abbildung 33 Prozessschritte der „Analysephase“ im Sub-Prozess Algorithmenentwicklung

Während der Zielbestimmung werden die Problemstellung aus Sicht der Algorithmenunterstützung nochmals beleuchtet und konkrete Ziele für die Entwicklung des Algorithmus erstellt. Die Datenanalyse als Folgeschritt (siehe Abbildung 33) hat zum Ziel, benötigte und vorhandene Datenquellen für die Entwicklung des Algorithmus zu identifizieren. Weiterhin werden im Schritt der Metrikenkonkretisierung Kennzahlen für die Messung der Algorithmen-Performance und -Sicherheit definiert und mit Zielwerten (bzw. Richtwerten) beschrieben, welche später in der Implementierungs- und Testphase angewandt werden.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Die folgende Aktivitäts-Rollen-Techniken-Ergebnis-Abbildung 34 gibt einen Überblick über alle methodischen Bausteine für jede Aktivität in dieser Phase.

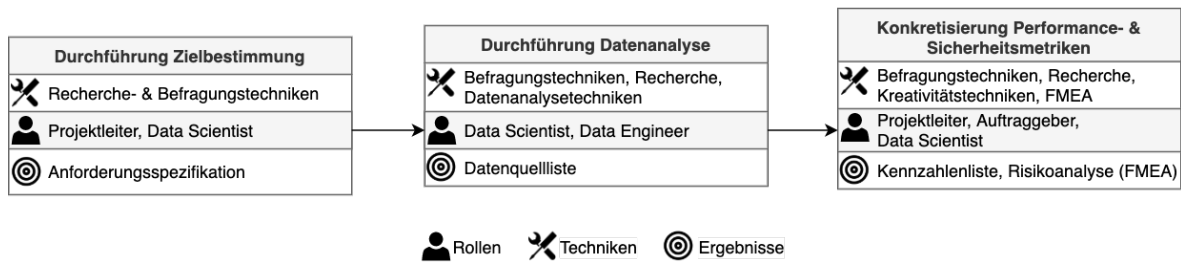


Abbildung 34 Übersicht der Aktivitäten für die Analysephase in der Algorithmusentwicklung

Nachfolgend werden die in Abbildung 33 aufgezeigten Schritte detailliert beschrieben.

6.3.1.1 Zielbestimmung durchführen

In diesem Schritt soll die Definition der Problemstellung und der Ziele für die Algorithmusentwicklung durchgeführt werden. Aus der Anforderungsspezifikation für das Gesamtsystem wird abgeleitet, was konkret unterstützt werden soll und wo die Algorithmusentwicklung ansetzt. Dabei werden die spezifischen Anforderungen an die Algorithmen definiert und in einer eigenen Anforderungsspezifikation festgehalten. Hierfür müssen die konkreten Abläufe und Schritte, in denen der Algorithmus etwas durchführen soll, identifiziert werden. Daraufhin kann analysiert und definiert werden, was für Daten der Algorithmus hierfür benötigt, wie das Design des Agenten und des Algorithmus aussehen soll, wie der Algorithmus funktionieren muss, um diese Unterstützung zu liefern und letztlich was das Ergebnis des Algorithmus ist.

Diese Aktivität kann vom Data Scientist gegebenenfalls unter Einbezug des Projektleiters durchgeführt werden, um eventuelle offene Fragen zu klären. Zur Durchführung können Techniken der Recherche und Befragung genutzt werden.

Das Ergebnis dieser Aktivität ist die Feststellung der Anforderungsspezifikation für den Algorithmus.

6.3.1.2 Datenanalyse durchführen

In der Aktivität der Datenanalyse werden die in dem Vorhaben zur Verfügung stehenden Datenquellen identifiziert und die Qualität der Daten analysiert. Dafür werden die benötigten Daten aus der Anforderungsspezifikation extrahiert und mit den zur Verfügung stehenden Datenquellen abgeglichen.

Dabei können die benötigten Daten von verschiedenen Quellen kommen. So können Daten aus der Domäne, also aus dem zu entwickelnden Zielsystem durch bspw. Smartphone-Sensoren oder den vorhandenen IT-Systemen, sowie aus Drittquellen, wie bspw. Wetterdatenbanken, bezogen werden.

Je nach Anforderungsspezifikation muss entschieden werden, ob die zur Verfügung stehenden Daten ausreichen oder ob ein Zukauf von Daten von kommerziellen Anbietern nötig ist. Um den Bedarf an zusätzlichen Datenquellen zu identifizieren, ist nicht nur die reine Verfügbarkeit von Daten relevant, sondern auch deren Qualität. Alle zur Verfügung stehenden Daten müssen auf Qualität (z. B. Prozentsatz der unvollständigen Datensätze), Menge der Datensätze, Art der vorliegenden Daten (z. B. textuell, binär, visuell etc.), Zeithorizontabdeckung bei Zeitreihen etc. geprüft werden. Entsprechende Kriterien für die Datenprüfung können je nach Anforderungen variieren und müssen vorab durch den Data Scientist und dem Data Engineer definiert werden.

Als Techniken zur Identifizierung der Datenquellen können die Recherche und die Befragung genutzt werden. Die Prüfung der Daten kann durch eine explorative Datenanalyse erfolgen (vgl. [Becker et al. 2016, 27] & [Brownlee 2018, 13]).

Als Ergebnis steht eine Datenquellliste, mit einer Beschreibung der vorhandenen Daten und deren Quellen sowie eventuellen Maßnahmen über die Qualitätsverbesserung bzw. des Fremdbezuges von weiteren notwendigen Daten.

6.3.1.3 Konkretisierung Performance- & Sicherheitsmetriken

Um sowohl die Performance als auch die Sicherheit bei der Entwicklung und dem Einsatz von präskriptiven Algorithmen zu gewährleisten, müssen zur Messung dieser Metriken erstellt werden. Ziel dieser Aktivität ist die Identifikation und Operationalisierung dieser Performance- und Sicherheitsmetriken.

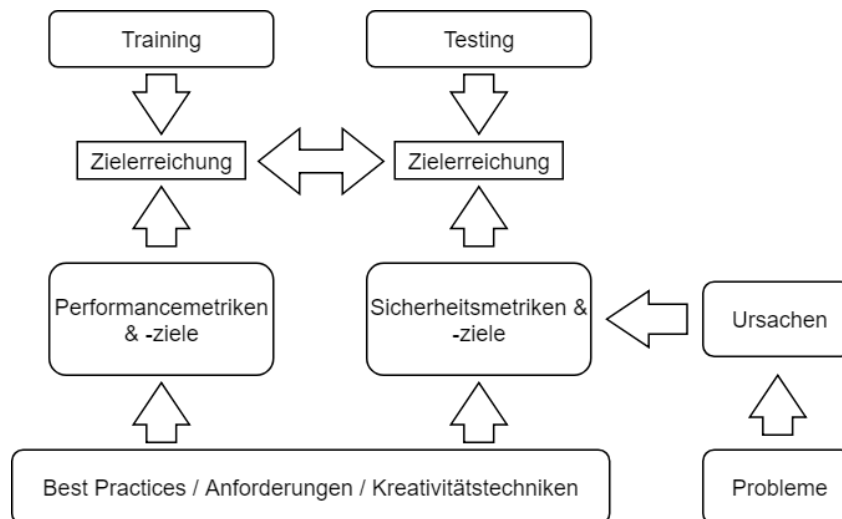


Abbildung 35 Schema Performance- und Sicherheitskennzahlen

In der Abbildung 35 ist der Entstehungsprozess der Metriken schematisch dargestellt. Zu sehen ist u. a., dass Performancemetriken unabhängig von Sicherheitsmetriken erstellt werden. Dies hat den Hintergrund, dass die Einsatzfelder von Performance- und Sicherheitsmetriken unterschiedlich sind.

Performance-Metriken sind technische Kennzahlen, die die Zielerreichung des Algorithmus definieren (z. B. der *mean squared error* [MSE, vgl. [Chollet 2018, 87]]), während Sicherheitskennzahlen auf Basis von Problemen und Ursachen definiert werden und in einer Risikobewertung genutzt werden können (z. B. Eintrittswahrscheinlichkeit). Kennzahlen für Performance werden während der Trainingszeit genutzt, während Sicherheitskennzahlen für die Testphase eingesetzt werden. Ein inhärenter iterativer Prozess zwischen Trainings- und Testphase zeigt, dass Änderungen in beiden Fällen Auswirkungen auf Kennzahlen des jeweils anderen Schrittes bewirken können. Performance- und Sicherheitskennzahlen sind somit grundsätzlich voneinander abhängig und können einander beeinflussen.

Performancekennzahlen

Wie bereits erwähnt, sind Performancekennzahlen technische Kennzahlen, die die Zielerreichung des Algorithmus beschreiben.

Bei prädiktiven Algorithmen können diese z. B. unter der Klasse der Klassifikationsalgorithmen die Area under the curve (AUC, vgl. [Chollet 2018, 112]) sein, unter der Klasse der Regressionsalgorithmen der MSE. In Optimierungsverfahren, wie dem Reinforcement Learning, ist eine bedeutende Kennzahl der Reward, der vom Agenten, je nach Definition, minimiert oder maximiert wird.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Zur Definition der Kennzahlen können Kreativitätsmethoden wie Brainstorming oder eine Recherche durchgeführt werden, um Best-Practice Ansätze und die Anforderungen an den Algorithmus hinzuzuziehen. Je nach Algorithmus-Anforderungen müssen zu den Kennzahlen Zielwerte definiert werden, die eine Zielerreichung messbar machen. Ist für das Projektvorhaben eine feingranulare Unterscheidung der Zielerreichung nötig, können mehrere Grenzwerte für die Zielerreichung definiert werden, z. B. ein Ampelsystem (vgl. [Mettler Toledo]).

Während der Trainingsphase (siehe Absatz 6.3.3.3) werden die Kennzahlen gemessen und bewertet. Sie definieren sogenannte Quality Gates (vgl. [Mensing 2015, 189]), die entscheiden, ob mit der nächsten Aktivität fortgefahren werden kann.

Sicherheitskennzahlen

Sicherheitskennzahlen dienen der Messung von Sicherheitskriterien und zur Reduzierung von sicherheitskritischen Verhalten während des Einsatzes des Algorithmus. Sie sollen die Robustheit des Algorithmus während des Einsatzes in der realen komplexen Welt bis zu einem definierten Maße gewährleisten. Da keine konkreten Sicherheitskennzahlen für präskriptive Algorithmen existieren, wird der Prozess der Erstellung dieser aus dem Prozess- und Qualitätsmanagement entlehnt.

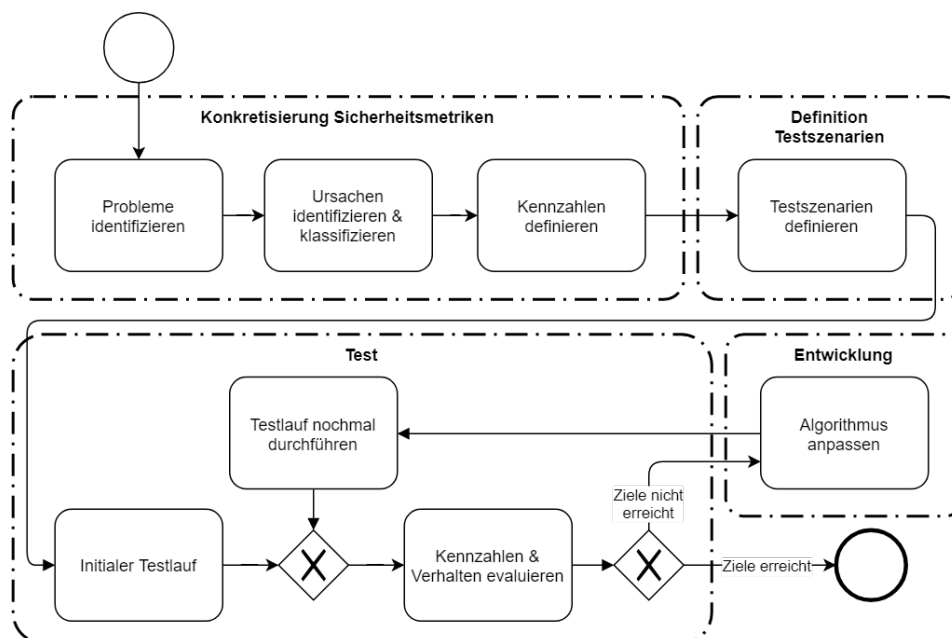


Abbildung 36 Aggregierter Sicherheitsprozess

Wie bereits in Abbildung 32 hervorgehoben wurde, erstrecken sich die Sicherheitsdefinitionen für den Einsatz von präskriptiven Algorithmen über den gesamten

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Teilprozess der Algorithmenentwicklung. Abbildung 36 zeigt in einer komprimierten Darstellung die Abfolge der sicherheitsrelevanten Schritte, von der Identifikation von Problemen bis zur Behebung dieser. Dadurch sollen mit Tests und Interpretierbarkeit Sicherheit geschaffen werden (vgl. [Leike et al. 2018, 16f.]). Ein Teil dieser komprimierten Darstellung findet, wie optisch abgegrenzt, in dieser Aktivität statt.

Wie bereits erwähnt, ist die Erstellung solcher sicherheitsrelevanten Metriken aus den Bereichen der Prozessoptimierung oder dem Qualitätsmanagement entlehnt. Zu Beginn steht die Identifikation der einzelnen Probleme, die während des Einsatzes stattfinden können. Dies kann durch den Data Scientist, den Projektleiter und unter anderem auch mit Domänenexperten, mithilfe von Kreativitätstechniken, wie das Brainstorming, durchgeführt werden. Die spezifischen Probleme sind stark anwendungsfallbezogen und abhängig von den jeweiligen Anforderungen an den Einsatz des Algorithmus. Als Probleme sind hier alle Rahmenbedingungen zu definieren, die während des realen Einsatzes eintreten können und die die Sicherheit der Endnutzer gefährden oder die Funktionalität und Entscheidungsfindung des Algorithmus mittel oder unmittelbar beeinflussen.

Sind die Probleme identifiziert, müssen die möglichen Ursachen für die Probleme bestimmt werden. Dies kann mit einem Ursache-Wirkungsdiagramm-Diagramm in einer Kreativrunde durchgeführt werden. Hierbei müssen die Ursachen konkreten Fehlerklassen zugeordnet werden. Dies dient der späteren Maßnahmendefinition, da bestimmte Fehlerklassen ähnliche Maßnahmen zulassen.

Die definierten und klassifizierten Probleme und Ursachen werden danach in eine FMEA (Fehlermöglichkeits- und Einflussanalyse) eingetragen. Diese hat gleich mehrere Vorteile für die spätere Auswertung der Kennzahlen. Probleme und Ursachen können anhand von Eintrittswahrscheinlichkeit und Auswirkung quantifiziert werden und bilden damit eine Kennzahl. Diese Kennzahl kann darauffolgend mit Grenzwerten und Maßnahmen versehen werden. Somit werden alle Probleme und Ursachen mit Kennzahlen, die die Qualität und damit die Sicherheit des Algorithmus beschreiben, versehen. Weiterhin können die einzelnen Probleme anhand ihrer Kennzahlen in Risikomatrizen abgebildet werden, um eine visuelle Auswertung durchführen zu können. Der größte Vorteil dieser Herangehensweise ist jedoch, dass die Kennzahlen der Auswirkung und der Eintrittswahrscheinlichkeit nicht nur subjektiv in Form eines Workshops festgelegt werden können, sondern sich während der Testphase objektiv in den einzelnen Testszenarien messen und aktualisieren lassen.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

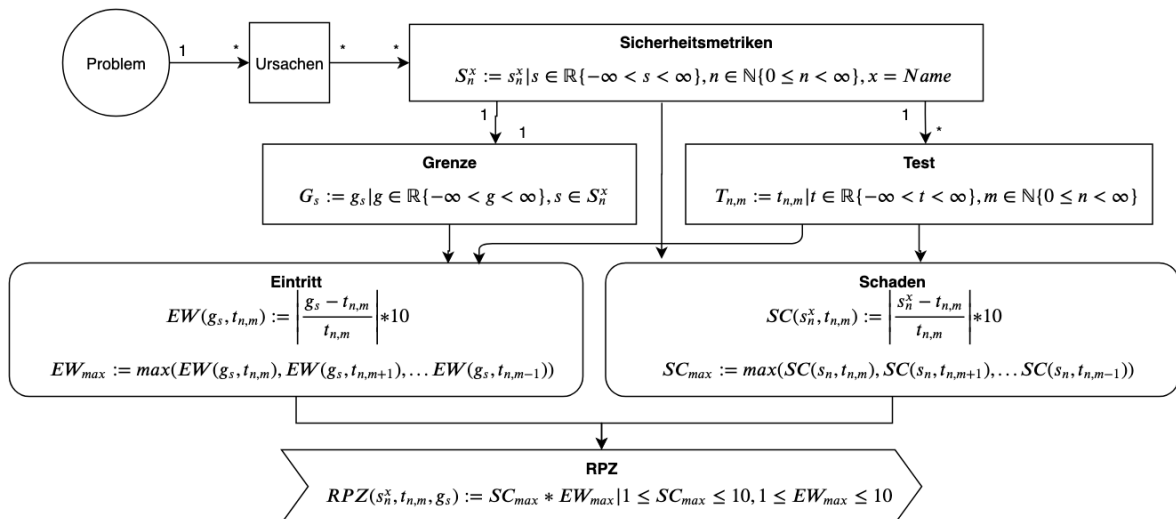


Abbildung 37 FMEA Berechnungen Sicherheitsmetriken

Abbildung 37 beschreibt die mathematische Vorgehensweise zur Berechnung der FMEA. Hierbei werden die Sicherheitsmetriken den Ursachen zugeordnet, um diese messbar zu machen. Sind mehrere Sicherheitsmetriken für die gleiche Ursache nötig, so werden die resultierenden Werte für Schaden und Eintritt der einzelnen Metriken addiert. Danach werden Grenzen je Sicherheitsmetrik und Ursache definiert, welche bei der Berechnung der Eintrittswahrscheinlichkeit als Maßeinheit dienen. Dabei ist, je nach Sicherheitskennzahl, die Bestimmung der Grenzen individuell durchzuführen. Nutzt man eine prozentuale Abweichung vom Normalwert für eine Kennzahl als Grenze, kann für eine andere Kennzahl eine Grenze mit festem Wert von Vorteil sein. Für jede Sicherheitsmetrik werden wiederum Testfälle durchgeführt, deren Ergebnisse in die Berechnung der Eintrittswahrscheinlichkeit und Schadenshöhe einfließen. Am Schluss wird aus dem Produkt der Eintrittswahrscheinlichkeit und der Schadenshöhe die Risikoprioritätszahl (RPZ) berechnet, welche, in gleicher Skalierung, vergleichbar mit den FMEA Bestandteilen jener Ursachen ist, die nicht in der Simulation, sondern auf herkömmliche Art und Weise berechnet wurden.

Die Erstellung der Performance- und Sicherheitsmetriken findet unter Anleitung und Überwachung des Projektleiters statt und bezieht den Data Scientist sowie bei Bedarf den Auftraggeber ein.

Das Ergebnis dieser Aktivität ist in der Summe die Kennzahlenspezifikationen für die Performance- und Sicherheitsmetriken sowie eine Risikobewertung und Analyse anhand einer FMEA.

6.3.2 Entwurfsphase

Die Entwurfsphase beschreibt die Zielumgebung sowie den Zielalgorithmus zur Lösung des zuvor definierten Problems, mithilfe der zur Verfügung stehenden Metriken und Datenquellen, und trifft vorbereitende Designentscheidungen für die folgende Implementierungs- und Testphase. Hierzu zählen, wie in Abbildung 38 dargestellt, die Datenvorverarbeitung, die Selektion und Entwicklung von Algorithmen und der Simulationsumgebung, die Entwicklung einer Recovery- und Enforcement-Strategie, die Zusammenführung der Simulationsumgebung und des Algorithmus zu einem Gesamtmodell sowie die Definition der Infrastruktur für das Training und Testen und der Entwurf der Testszenarien auf Basis der zuvor identifizierten Probleme und Ursachen.

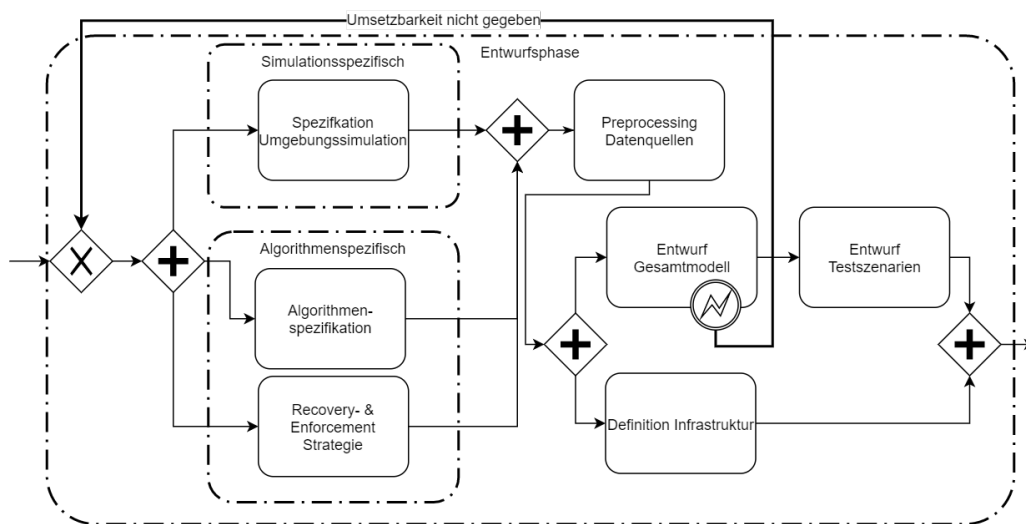


Abbildung 38 Prozessschritte der „Entwurfsphase“ im Sub-Prozess Algorithmenentwicklung

Bei der Datenselektion werden die relevanten Datenquellen ausgewählt und für die Nutzung in der Simulation und in den Algorithmen vorbereitet. Daraufhin können sowohl die Algorithmen- als auch die Simulationsumgebungsentwicklung parallel zueinander betrieben werden. In beiden Fällen werden bestehende Möglichkeiten und Ansätze eruiert und ggf. angepasst und übernommen. Eigenentwicklungen sind in beiden Fällen ebenfalls möglich. Die Recovery- und Enforcement-Strategie soll Vorgehen und Alternativepläne definieren, welche später im laufenden Betrieb Notfallsituationen abfangen und den Regelbetrieb aufrecht erhalten. Nachdem der Algorithmus und die Simulation definiert worden sind, kann in einem Gesamtmodell beides zusammengeführt werden. Hieraus entsteht ein Gesamtbild des Agenten und dessen Simulationsinteraktion, was u. a. für die Definition der Testszenarien, neben den identifizierten Problemen und Ursachen, genutzt werden kann. Sollte das Gesamtmodell aus technischen oder anderen Gründen nicht oder nur suboptimal

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

umzusetzen sein, kann ein Rücksprung zum Beginn der Phase durchgeführt werden. Abschließend wird in der Entwurfsphase die Infrastruktur definiert, welche für das Training und die Tests benötigt wird.

Die nachfolgende Abbildung 39 soll einen Überblick über die benötigten Rollen, genutzten Techniken und produzierten Ergebnisse liefern.

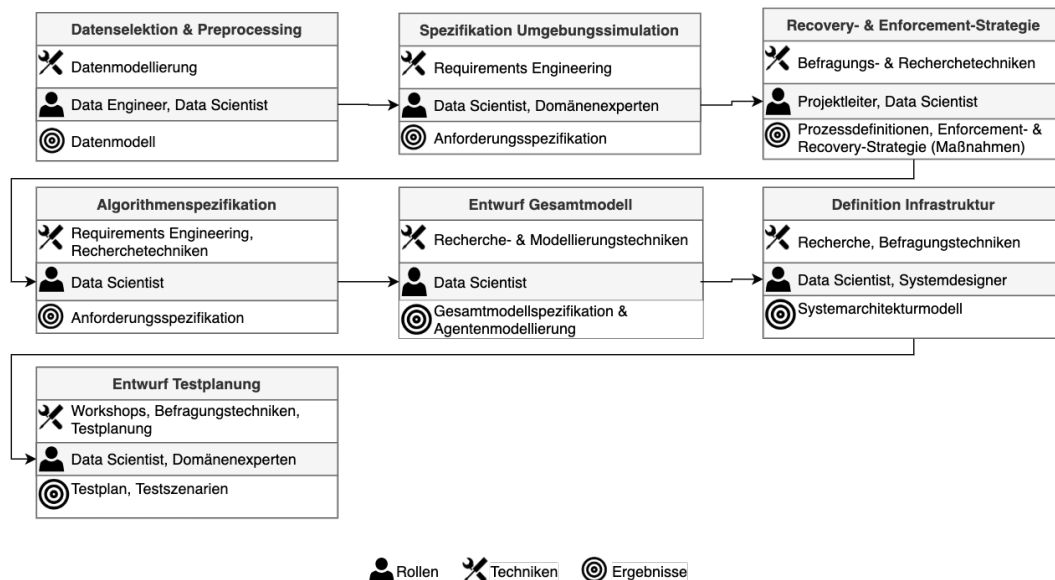


Abbildung 39 Übersicht der Aktivitäten für die Entwurfsphase in der Algorithmenentwicklung

In den folgenden Absätzen werden die aus Abbildung 38 ersichtlichen Schritte noch einmal detailliert erörtert.

6.3.2.1 Datenselektion und Preprocessing

Nachdem in der Analysephase die Datenquellen mit ihren Dateneigenschaften aufgenommen worden sind, werden diese nach den Anforderungen an den Algorithmus und die Simulationsumgebung selektiert und vorverarbeitet. Ziel ist die Bereitstellung der benötigten Daten, für den zu definierenden Agenten und die Simulationsumgebung, mit den benötigten Eigenschaften (z. B. Verfügbarkeit, Qualität und Menge).

Grundsätzlich gibt es keine Notwendigkeit, vorhandene Daten nutzen zu müssen. Abhängig vom Anwendungsfall können diese in der Simulation und dem Algorithmus mit eingebunden werden.

Die Verarbeitung der Datenquellen ist je nach vorliegender Art der Daten und deren Eigenschaften unterschiedlich. Beispielfhaft müssen Bilddaten für den Algorithmus auf die passende Eingabegröße skaliert und mit der richtigen Bildtiefe (Anzahl der Kanäle im Bild, z. B. Farben oder Graustufen) versehen werden (vgl. [Fricklas/Shukla 2018, 173ff.]). Bei

allgemeinen, häufig strukturierten Daten wie Adressen etc., sind fehlende Werte zu ergänzen oder kategorische Daten für die Eingabe in den Algorithmus zu encodieren (z. B. one-hot-encoding, vgl. [Chollet 2018, 101ff.] & [Deshpande/Kumar 2018, 171f.]).

Diese Schritte der Datenvisualisierung, Vorverarbeitung und Bereitstellung dieser, kann durch einen Data Engineer oder einem Data Scientist, unter Anwendung von Datenmodellierungstechniken, durchgeführt werden.

Als Ergebnis dieser Aktivität steht am Ende ein Datenmodell, mit einer für das weitere Vorgehen vorbereiteten Datengrundlage mit entsprechender Dokumentation, zur Verfügung.

6.3.2.2 Spezifikation Umgebungssimulation

Parallel zur Spezifikation des Algorithmus wird die Spezifikation der Simulationsumgebung durchgeführt. Die Simulationsumgebung wird zum späteren Trainieren und Testen des Agenten genutzt und soll somit die Entwicklung unterstützen. Vorteilhaft hierbei ist die sich eröffnende Möglichkeit, je nach Aufgabe, das virtuelle Simulationen kosteneffizienter gegenüber Realtests sind und zum einen sicherer in Testsituationen, in welchen der Agent noch keine klaren Regeln gelernt hat und eventuelle Fehler gefährlich für die Zielgruppe sein könnten und zum anderen schneller in Testerfordernissen, in welchen es sich anbietet, Simulationen parallel auf mehreren Maschinen auszuführen, was in der Realität vielleicht nicht möglich ist (vgl. [Sutton/Barto 2018, 476]).

Als erstes müssen die Rahmenbedingungen für die Simulationsumgebung aus den allgemeinen Anforderungen und Zielstellungen an den Algorithmus entnommen werden. Daraus müssen das Ziel und die spezifischen Anforderungen der Simulation definiert werden. Dies betrifft vor allem die Frage, was überhaupt simuliert werden soll. Hieraus ergeben sich weitere Anforderungen, die durchgegangen werden müssen. Als Technik kann das Requirements Engineering genutzt werden.

Bspw. muss geklärt werden, ob die Simulation und das Verhalten des Agenten ausschließlich über die Metriken zugänglich ist oder auch grafische Ausgaben möglich sind, um weitere Verhaltensanalysen durchführen zu können. Auch muss definiert werden, welche Informationen über die definierten Performance- und Sicherheitsmetriken hinaus für Analysezwecke mitgeteilt werden.

Bereits vorhandene Daten aus den Datenquellen können, für eine realitätsnähere Ausrichtung der Simulation, mit eingebunden werden oder ggf. durch ein Modell erlernt und

danach generiert werden (vgl. [Tremblay et al. 2018]). Die Selektion, welche Daten für die Simulation als relevant zu erachten sind oder ob überhaupt Daten einbezogen werden sollen, wird vom Data Scientist, unter Beteiligung des Data Engineers, durchgeführt.

Weiterhin ist es erforderlich zu definieren, wie umfangreich die Simulation wird. Dies hat verschiedene Hintergründe. Je nach Anwendungsfall muss abgewogen werden, wie realitätsgetreu oder approximativ die Simulation werden soll. Je genauer die Simulation an die Realität anschließen soll, desto aufwändiger und ressourcenintensiver kann diese werden. Dies muss anhand der Anforderungen und Ziele abgewogen werden. Die Komplexität der Simulation stellt hiermit die Entscheidung in den Vordergrund, ob eigene Hardware für den Betrieb der Simulationsumgebung nötig ist und ob es möglich sein muss, dass mehrere Agenten gleichzeitig eine Simulation nutzen oder jeder Agent jeweils seine eigene. Dabei spielt die Skalierbarkeit der Simulation für die Verwendung in parallelen Lernverfahren eine entscheidende Rolle, die mitgedacht werden muss.

Ein sehr wichtiger Punkt bei der Definition der Simulationsumgebung ist die Festlegung des State- und Action Spaces. Wie bereits in Abschnitt 2.2 beschrieben, können der State- und der Action Space kontinuierlich oder diskret sein. Diese können verschiedene Komplexitäten besitzen, die sich auf die Eigenschaften des zu definierenden Algorithmus auswirken. Die Definition des State- und des Action Spaces entscheidet über die Rückgabefunktionen (States) für den Agenten und die möglichen Aktionen (Actions) die der Algorithmus durchführen kann (vgl. [Lillicrap et al. 2015, 1] & [Dulac-Arnold et al. 2015, 1]).

Zusätzlich zum State und Action Space, muss die relevante Performance-Metrik (Reward), passend nach Definition, umgesetzt werden. Die Umsetzung dieser Anforderung ist ebenfalls entscheidend für die Wahl des Algorithmus (vgl. [Sutton/Barto 2018, 1ff.]). Wo die Anforderungen an die Performance-Metriken das *Was* definiert haben, muss nun konkret das *Wie* spezifiziert werden. Entsprechende Techniken für das Reward-Design wie z. B. Scaling (vgl. [Henderson et al. 2017, 3]) oder hierarchische Reward-Strukturen (vgl. [Gudimella et al. 2017, 15]) müssen nach dem State of the Art recherchiert und angewendet werden, um die bestmöglichen Lernbedingungen zu schaffen.

Auch die Wahl der Programmiersprache und der möglichen zu verwendenden Frameworks, sollte bei der Erstellung der Simulation mit einbezogen werden.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Sind die Rahmenbedingungen und Anforderungen an die Simulation geklärt, sollte vor einer eigenen Umsetzung dieser eine Recherche nach bestehenden Simulationsumgebungen durchgeführt werden. In die Betrachtung sollten sowohl kommerzielle als auch freie Simulationsumgebungen einbezogen werden. Eventuell sind bereits Simulationen oder Simulationsframeworks vorhanden, die eine grundlegende Neuentwicklung unnötig machen und lediglich für den Anwendungsfall konfiguriert werden müssen.

Die Spezifikation kann vom Data Scientist, bei Notwendigkeit der Klärung offener Fragen unter Einbezug von Domänenexperten, durchgeführt werden.

Ergebnis dieser Aktivität ist eine Anforderungsspezifikation der Simulationsumgebung.

6.3.2.3 Enforcement & Recovery Strategie

Sicherheit beim Einsatz von präskriptiven Algorithmen ergibt sich nicht nur durch das Testen der Entscheidungen des Agenten, sondern ebenso durch die Absicherung der interagierenden Prozesse. Da präskriptive Algorithmen im Grundsatz auch nur technische Services sind, kann bei der Prozessbetrachtung die Information Technology Infrastructure Library (ITIL) genutzt werden. Die folgenden Erläuterungen beziehen sich deshalb auf durch ITIL definierte Prozesse.



Abbildung 40 Enforcement- & Recovery-Strategie ITIL Einordnung

Um die Prozesse besser zusammenfassen zu können, vor allem ohne Hintergrund von ITIL, können diese in Enforcement-Strategie und Recovery-Strategie basierte Prozesse unterteilt werden. Die Enforcement-Strategie beinhaltet das Monitoring sowie alle Prozesse, die das Verhalten des Algorithmus beschränken und kontrollieren sollen. Dies bezieht sich auf das Design-, Release- und Monitoring-Management. Die Recovery-Strategie wiederum lässt sich auf das Incident-Management beziehen und soll Fehlerfälle abfedern.

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Bei der Absicherung von präskriptiven Algorithmen spielen vor allem das Design- und das Release- sowie das Monitoring- und Incident-Management eine wichtige Rolle.

Im Design-Management des Algorithmus muss auf die Informationssicherheit im Entwicklungsprozess geachtet werden. Konkret bezieht sich dies auf die Absicherung der geistigen Inhalte des Algorithmus, wie Quellcode oder Agentendesigns. Hierbei spielen Zugriffskontrollen und Änderungsrechte eine Rolle. Es muss hierbei sichergestellt werden, dass nur Personen auf den Algorithmus direkten Zugriff haben können, die sowohl fachlich qualifiziert als auch organisatorisch berechtigt sind. Wird der Algorithmus bzw. der Agent verändert, kann dies zur Invalidität der Testszenarien und Testergebnisse führen, wodurch hierfür ebenfalls ein entsprechender Prozess im Rahmen des Change Managements definiert werden muss, also im Umgang mit Änderungen an dem Service (vgl. [Persse 2012, 187ff.]).

Während des Bereitstellens (Release- und Deployment-Management) einer neuen Version des Algorithmus muss entsprechend der definierten Metriken (siehe Absatz 6.3.1.3) und Quality Gates sichergestellt werden, dass die Qualität des Algorithmus gleich bzw. besser der vorhergehenden Version ist. Eine Absicherung, mit automatischen Tests und Reviews durch entscheidungsbefugte Mitarbeiter, ist durchzuführen und, je nach Anwendungsfall und Kritikalität, einzusetzen.

Im Rahmen des Monitoring- und Incident-Managements liegt der Fokus vor allem auf dem Betrieb des Agenten. Um eine gleichbleibende Qualität des Agenten zu gewährleisten, vor allem bei Änderungen der Umwelt und somit dem Auftreten von Fehlerquellen durch ungekannte Zustände, müssen in einem kontinuierlichen Logging und Monitoring relevanter Zielgrößen, wie derjenigen aus Absatz 6.3.1.3, Hard- und Soft-Constraints für die Genzwertüberwachungen als auch Metriken zum Einbezug von Nutzer-Feedback (direkt durch Feedback-Möglichkeiten oder indirekt durch Verhaltensmonitoring) durchgeführt werden. Die Identifikation dieser Monitoring Metriken kann zusätzlich im Rahmen eines Workshops mit dem Data Scientist und den Domänenexperten durchgeführt werden. Umfang und Aufwand der Monitoring Parameter sollten sich am Anwendungsfall ausrichten und nach Bedarf und Kritikalität gewählt werden.

Außerhalb der ITIL Definition für den Monitoring Prozess, aber innerhalb der Enforcement-Strategie, liegt die Sicherstellung der Unterbrechbarkeit und Eingriffsmöglichkeit in den Agenten. Unabhängig der Kritikalität und Komplexität des Systems sollte technisch und organisatorisch die Möglichkeit geschaffen werden, in die Entscheidungen des Agenten

eingreifen und diese ggf. überschreiben bzw. unterbrechen zu können. Damit kann sichergestellt werden, dass durch kontinuierliches Monitoring Fehler nicht nur erkannt, sondern auch direkt auf diese reagiert werden kann.

Tritt im laufenden Betrieb ein Fehler auf oder werden definierte Grenzwerte überschritten, muss im Rahmen des Incident-Managements bzw. der Recovery-Strategie reagiert werden. Dies bezieht sich auf die Fehlerbehebung in der Ausführung des Agenten. Ein besonderes Augenmerk muss hierbei auf ein Fallback-Handling und die Reversibilität / Error Correction gelegt werden.

Das Fallback-Handling beschreibt eine Rückfalllogik oder einen Rückfallprozess der greift, wenn der Agent aus Fehlergründen nicht mehr zur Verfügung steht. Dies kann sowohl eine deskriptive Programmlogik sein als auch ein Mensch, der den Agenten im Fehlerfall mit seinen Entscheidungen ersetzt.

Reversibilität sollte in die Betrachtung mit einbezogen werden, da das Wissen über die Reversibilität der Entscheidungen durch den Agenten Aussagen über die Tragweite der aufgetretenen Problemstellung zulässt. Je nach Design des Algorithmus und des Agenten können Fehler bereits früh identifiziert und entsprechend rechtzeitig darauf reagiert werden, wodurch eine hohe Reversibilität besteht. Die Bedeutsamkeit und die Umsetzung der Reversibilität, ist durch den Data Scientist und den Domänenexperten zu definieren und sollte in die Aktivität der Algorithmenspezifikation (siehe Absatz 6.3.2.4) einfließen.

Eine, je nach Kritikalität des Anwendungsfalles, geeignete Umsetzung der Recovery-Strategie, wird vom Projektleiter zusammen mit dem Data Scientist beschrieben.

Am Ende der Aktivität wird sowohl eine Enforcement- als auch eine Recovery-Strategie in Form eines Maßnahmenkatalogs definiert und beschrieben, welche verschiedene Prozessdefinitionen aus dem ITIL-Framework beinhalten.

6.3.2.4 Algorithmenspezifikation

In dieser Aktivität wird der Algorithmus ausgewählt und definiert. Dies beinhaltet die grundsätzliche Technik, die eingesetzt werden soll, als auch die konkrete Ausprägung sowie die dazugehörigen Parameter und Erweiterungen dieser Ausprägung. Techniken können zum Beispiel logikbasierte Systeme wie Modallogiken und Fuzzy-Logiken, genetische Algorithmen wie evolutionäres Deep Learning, Reinforcement Learning sowie weitere sein.

Um eine Auswahl zwischen diesen und ggf. anderen Techniken durchführen zu können, kann vom Data Scientist, neben einer Recherche, eine Nutzwertanalyse durchgeführt werden. Diese beinhaltet als gewichtete Parameter Anforderungen aus dem Anwendungsfall, der Sicherheit und der Performance, der Enforcement & Recovery Strategie, der Skalierbarkeit sowie der Simulations- und Zielumgebung.

Wurde die Technik passend für den Anwendungsfall ausgewählt, kann durch eine Recherche durch den Data Scientist, unter Berücksichtigung des State of the Art sowie unter Einbezug aller zuvor benannten Anforderungen, eine konkrete Ausprägung der Technik, also der konkrete Algorithmus, bestimmt werden. Dabei ist immer auf die grundsätzliche Machbarkeit und Verhältnismäßigkeit, bezogen auf den Anwendungsfall, zu achten. Zudem besteht die Möglichkeit, eine Reihe von potenziellen Algorithmen auszuwählen, die zusammen in der Trainings- und Testphase verglichen werden können. Dies erhöht zwar den Aufwand in der Implementierung und dem explorativen Vorgehen, kann aber zu besseren Ergebnissen, aufgrund einer größeren Versuchsbasis, führen.

Anzumerken sei, dass in der Vorhabensdurchführung die Entwicklung grundsätzlich neuer Algorithmen (Grundlagenforschung) nicht mit betrachtet wird. Dies erklärt sich aus dem zu erwartenden Aufwand in Zeit und Personalkosten, die eine solche Grundlagenforschung mit sich bringt. Unberührt hiervon bleiben die Modifikationen bestehender Algorithmen, die zur Performance oder Sicherheit des Agenten beitragen und während der Implementierungs- und Testphase entwickelt werden.

Am Ende der Aktivität wird eine Anforderungsspezifikation für den Algorithmus, ggf. unter Einsatz des Requirement Engineerings bzw. von Rechartechniken, erstellt, welche für die Definition des Gesamtmodells und den daraus entstehenden Agenten genutzt wird.

6.3.2.5 Entwurf Gesamtmodell

Mit den fertigen Spezifikationen von Simulationsumgebung und Algorithmus kann der Agent eingebettet in das Gesamtmodell erstellt werden. Das Gesamtmodell beschreibt die Beziehung aller Komponenten untereinander sowie den Agenten in seiner Struktur und seinen Interaktionsmöglichkeiten. Ziel ist es hierbei, die einzelnen Komponenten aus der Simulationsumgebung, den Daten, den Algorithmen und den Zielen zusammenzuführen, was in die Definition des Agenten mündet. Der Agent ist somit eine frei agierende Einheit, welche in der Simulationsumgebung zur Erreichung seines definierten Zieles Aktionen durchführt und unter Einsatz eines Algorithmus Zustände auswertet (vgl. [Russell/Norvig

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

2003, 194ff.] & [Sutton/Barto 2018, 2]). Gleichwohl kann das Gesamtmodell als Architekturüberblick genutzt werden und kann z. B. unter Einsatz von Modellierungstechniken zur einheitlichen Kommunikation mit Fachexperten umgesetzt werden, wenn hierfür eine Zweckmäßigkeit gegeben ist.

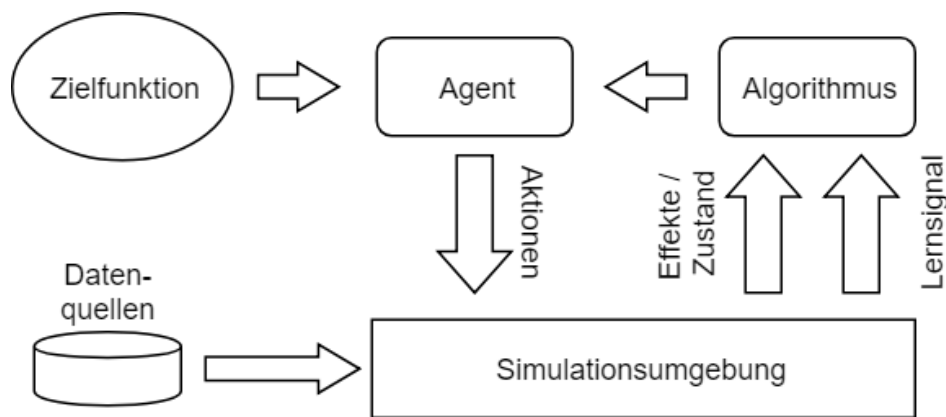


Abbildung 41 Gesamtmodell Komponentenüberblick

Abbildung 41 zeigt den Zusammenhang der einzelnen Komponenten im Gesamtmodell. Bereits im Vorfeld definierte Komponenten sind die Datenquellen (siehe Absatz 6.3.2.1), die Simulationsumgebung inkl. Zustands- und Aktionsraum (siehe Absatz 6.3.2.2), die Zielfunktion als Performance-Metrik (siehe Absätze 6.3.1.3 und 6.3.2.2) sowie der Algorithmus (siehe Absatz 6.3.2.4). Diese fließen in die Definition des Agenten mit ein.

Der Agent muss hinsichtlich seiner internen Struktur sowie in seiner Systemgestaltung beschrieben werden, in Abhängigkeit von allen anderen relevanten Bausteinen.

In der internen Agentenstruktur wird festgelegt, wie der Zustandsraum mit seinen Elementen erfasst und verarbeitet wird, wie die Interaktion mit eventuellen weiteren Agenten durchgeführt und verarbeitet wird und wie die Aktionen passend auf dem Aktionsraum der Simulationsumgebung erreicht werden. Diese interne Struktur muss technisch beschrieben und definiert werden und kann zusätzlich visuell durch Modellierungstechniken, vor allem bei komplexen Agentenarchitekturen, umgesetzt werden.

Die Agentensystemgestaltung beschreibt im Vergleich zur internen Agentenstruktur nicht einen Agenten selbst, sondern den Systemraum. So kann in der Interaktion mit der Simulationsumgebung lediglich ein Agent als globaler und allumfassender Agent existieren oder ein gekoppelter oder loser Verbund aus mehreren Agenten. Ein solcher Verbund ist unter dem Begriff der Multiagentensysteme bekannt und beschrieben. Diese können hierarchisch, kooperativ oder konkurrierend miteinander interagieren und entsprechend

Informationen sowie erlerntes Verhalten untereinander austauschen (vgl. [Weiss 2000, 4ff.]).

Alle Entscheidungen, hinsichtlich Struktur und Eigenschaften des Gesamtmodells, sind auf den Anwendungsfall anzupassen und vom Data Scientist zu erstellen.

Als Resultat dieser Aktivität steht das Gesamtmodell sowohl als Beschreibung als auch als visuelle Modellierung zur Verfügung für die Entwicklung des Agenten und dessen Komponenten.

6.3.2.6 Definition Infrastruktur-Anforderungen

Parallel zum Entwurf des Gesamtmodells und der Testfälle ist die Definition der benötigten Infrastruktur zum Training und Testen des Agenten nötig. Hier entfalten sowohl die Anforderungen an die Simulationsumgebung als auch die Anforderungen an das Gesamtmodell ihre zentrale Bedeutung. Es müssen die Skalierbarkeit und die benötigte Rechenleistung sowie das Vorgehen bei der iterativen Entwicklung des Agenten mit einbezogen werden. Alle zu betrachtenden Punkte beziehen sowohl Software- als auch Hardwarelösungen mit ein. Dabei ist auch zu entscheiden, ob eine eigene Infrastruktur genutzt oder z. B. eine externe Cloud-Lösung bevorzugt werden soll. Dies kann vom Data Scientist, in Rücksprache mit den Systemarchitekten, durchgeführt werden. Weiterhin müssen erste Betrachtungen für das Bereitstellen des trainierten und getesteten Agenten angestellt werden, um diesen für den später anstehenden Realtest (siehe dazu Absatz 6.2.3.3) zur Verfügung stellen zu können.

Die Aktivität erzeugt nach Beendigung ein Systemarchitekturmodell, mit einer Beschreibung der benötigten Komponenten und Dienste.

6.3.2.7 Entwurf Testplanung

Zur Gewährleistung der Sicherheit des Agenten nach erfolgreichem Abschluss des Performance-Trainings (siehe Absatz 6.3.3.3), müssen für die darauffolgende Testphase Testfälle definiert und ein Testplan erstellt werden. Auf Basis des Simulationsumgebungs-Entwurfs und der Sicherheitsanforderungen mit deren definierten Problemen und Ursachen, können die benötigten Testfälle erstellt werden, die eine möglichst vollständige Abdeckung der identifizierten Probleme und deren Ursachen ermöglichen sollen.

Diese Testfälle können durch die Einarbeitung verschiedener Situationen bzw. Zustände innerhalb einer Simulation umgesetzt werden. Dabei ist darauf zu achten, dass sowohl Testsituationen einzeln in einer Simulation als auch im Verbund mit anderen Testsituationen

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

in einer Simulation getestet werden. Inwieweit Kombinationen getestet werden sollen, ist je nach Anwendungsfall und möglicher Eintrittswahrscheinlichkeit in der Kombination beider Aspekte zu beurteilen und kann vom Data Scientist, in Zusammenarbeit mit den Domänenexperten, geschehen.

Nach der Erstellung der Testfälle, sind diese in einen Testplan einzubetten, der zusätzlich zu den Testfällen noch deren Reihenfolge und Ablaufplanung sowie die Protokollierung und Auswertung beinhaltet. Zur Auswertung der Testfälle sind die Sicherheitsmetriken (Absatz 6.3.1.3) zu nutzen, die im Vergleich zu einem Basislauf (einem optimalen Durchlauf der Simulation) die Auswirkungen der Testszenarien operationalisieren.

Zur Erstellung des Testplans kann auf Techniken der Testplanung aus der Software-Entwicklung zurückgegriffen werden (vgl. [Sommerville 2011, 208ff.]). Dieser wird vom Data Scientist, zusammen mit Domänenexperten, erstellt.

Ergebnis dieser Aktivität ist der definierte Testplan mit detaillierter Beschreibung aller definierten Testfälle.

6.3.3 Entwicklungs- & Testphase

Die Entwicklung und das Testen von Simulationsumgebung und Agenten findet in der Implementierungs- und Testphase statt. In dieser werden sowohl die Simulation nach den vorausgegangenen Spezifikationen als auch der Agent erstellt und umgesetzt. Zusammen mit der Simulation wird der Agent hinsichtlich Performance und Sicherheit getestet. Ziel ist hierbei, gemäß den vorab definierten Metriken (siehe Absatz 6.3.1.3), den Agenten umzusetzen und zu testen und diesen somit auf die Anwendung im Realtestumfeld (siehe Absatz 6.2.3.3) vorzubereiten.

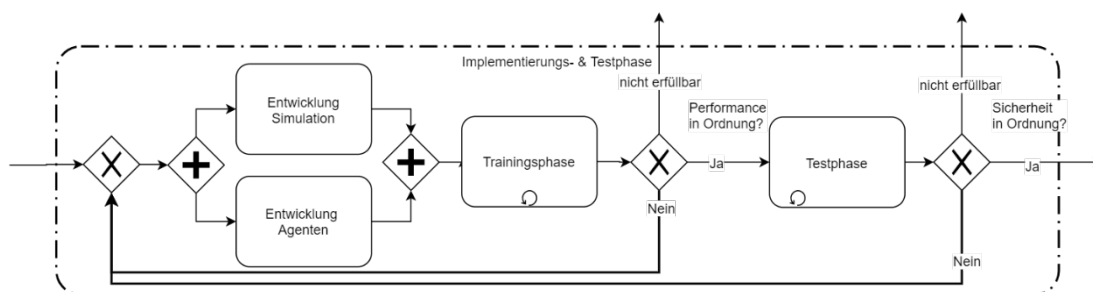


Abbildung 42 Prozessschritte der „Implementierungs- & Testphase“ im Sub-Prozess Algorithmenentwicklung

Mit der Entwicklung der Simulationsumgebung wird die Trainings- und Testumgebung geschaffen, in welcher der Agent antrainiert und getestet wird. Dabei wird eine Simulation für die Performance Messung und weitere, an den Testplanungen orientierte

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Testsimulationen für das Testen der Sicherheitsmetriken erstellt. Die Entwicklung des Agenten kann parallel zur Entwicklung der Simulation durchgeführt werden, da beide wohl definiert über standardisierte Schnittstellen beschrieben sind und somit softwareseitig getrennt betrachtet werden können. In der Trainingsphase werden danach beide Artefakte zusammengeführt. Die zuvor definierten Performance-Metriken (vgl. Absatz 6.3.1.3) kommen beim trainieren des Agenten zum Einsatz, um dessen Grad der Zielerfüllung zu bewerten. Daraufhin werden die Sicherheitsmetriken bei der Ausführung des Agenten in den Testumgebungen aufgenommen und evaluiert. Erst wenn sowohl die Performance- als auch die Sicherheitsziele erreicht wurden, kann die Phase abgeschlossen und der Agent in den Realtest übergeben werden. Ebenso kann es vorkommen, dass die gewählten Performance- und oder Sicherheitsmetriken nicht erfüllbar sind. Unter diesem Aspekt können Rücksprünge in die Analysephase durchgeführt werden.

Folgende Abbildung 43 visualisiert, die für die Umsetzung dieser Phase benötigten Techniken und Rollen sowie die daraus entstehenden Ergebnisse.

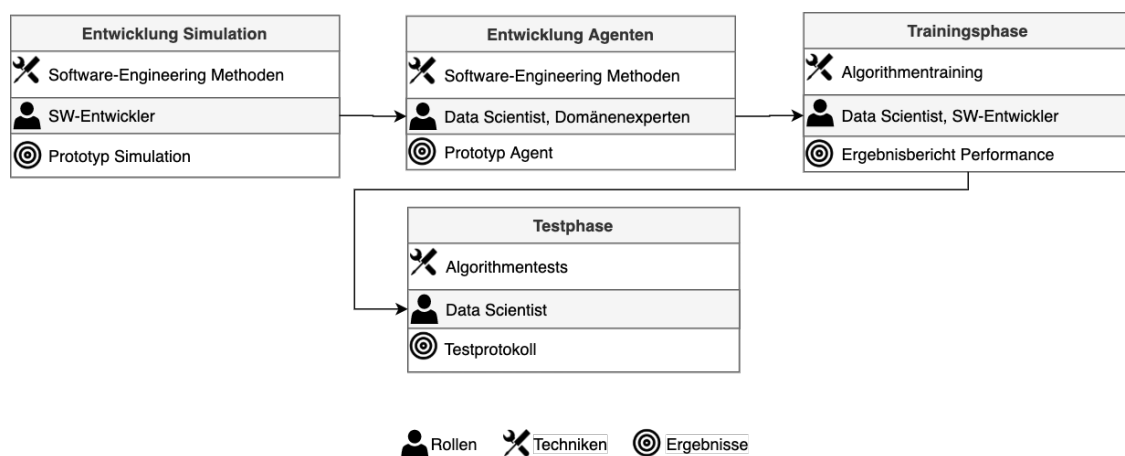


Abbildung 43 Übersicht der Aktivitäten für die Implementierungs- & Testphase in der Algorithmenentwicklung

Nachfolgend werden die einzelnen Schritte aus Abbildung 42 näher definiert und erörtert.

6.3.3.1 Entwicklung Simulation

In dieser Aktivität wird auf Basis der Anforderungsspezifikation die Simulation umgesetzt. Zur softwareseitigen Umsetzung der Simulationsumgebung kann ein beliebiges Entwicklungsvorgehen für Software-Entwicklungsprojekte genutzt werden, sofern dieses die Anforderungen für Vorgehensmodelle (siehe Unterabschnitt 4.1.1) erfüllt. Beispielfähig können dies das V-Modell XT oder Scrum sein, wie bereits in Absatz 6.2.3.1 erläutert.

Zur Entwicklung können auch Softwareentwickler ohne explizite Kenntnisse über Machine Learning oder andere Lernverfahren herangezogen werden, da die Simulationsumgebung

ein Abbild der realen Welt erzeugen soll, jedoch keine spezielle Logik des Algorithmus beinhaltet.

Das Ergebnis dieser Aktivität ist die Herstellung eines funktionsfähigen Software-Prototyps der Simulationsumgebung gemäß deren Spezifikation.

6.3.3.2 Entwicklung Agenten

Der Agent kann unter Betrachtung des Gesamtmodells und der Algorithmenspezifikation umgesetzt werden. Das Ziel hier ist, wie in der Implementierung der Simulationsumgebung, die Entwicklung eines lauffähigen Prototyps. Dieser Agenten-Prototyp soll in der Trainings- und Testphase in der Lage sein, mit der Simulationsumgebung interagieren und währenddessen sein Verhalten in dieser anpassen zu können.

Für die Entwicklung des Agenten-Prototyps kann, wie auch schon bei der Entwicklung der Simulation, ein beliebiges Vorgehensmodell zur Softwareentwicklung genutzt werden, sofern dieses die allgemeinen Anforderungen gemäß Abschnitt 4.1.1 erfüllt. Im Unterschied zur Simulationsumgebung kann für die Entwicklung dieser Software-Komponente jedoch nicht jeder Software-Entwickler herangezogen werden.

Aufgrund der speziellen fachlichen Anforderungen an den Entwickler für diesen Software-Prototypen, hinsichtlich der Kenntnisse spezieller mathematischer Modelle und deren softwareseitige Umsetzung durch den Einsatz spezieller Software-Frameworks (z. B. Tensorflow für neuronale Netzwerke u. a. [vgl. [Fricklas/Shukla 2018, 21f.]]), hat die Implementierung durch den Data Scientist oder ähnlich ausgebildete Software Entwickler zu erfolgen.

Nach Beendigung der Entwicklung des Agenten-Prototyps steht ein lauffähiger Software-Prototyp, welcher ausgereift für den Einsatz zum Training in der Simulationsumgebung ist.

6.3.3.3 Trainingsphase

Die Trainingsphase dient der grundsätzlichen Umsetzung des definierten Verhaltens, unabhängig von der Betrachtung etwaiger Sicherheitsrisiken. Unter Betrachtung der bereits in einer vorherigen Aktivität definierten Performance-Metriken (siehe Absatz 6.3.1.3), wird dem Algorithmus in einer Schleife aus Ausführen und Anpassen das zu erzielende Verhalten antrainiert.

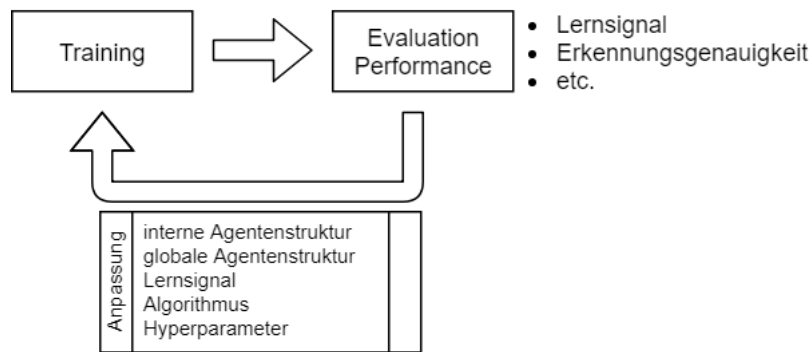


Abbildung 44 Trainingsphase Agent Vorgehen

In Abbildung 44 ist das Vorgehen noch einmal übersichtlich illustriert. Mit der Durchführung des Ausführens und Trainings des Algorithmus können, basierend auf dessen Entscheidungen, Verläufe von Handlungsketten beobachtet werden, die das Verhalten des Agenten beschreiben. Diese Handlungsketten erzeugen nach den zuvor definierten Performance-Metriken, wie bspw. den Lernsignalen, Werte, die entweder zufriedenstellend sind oder Anpassungen benötigen. Mit einer iterativen Anpassung der internen Struktur des Agenten, der globalen Agentenstruktur, der Anpassung des Lernsignals, des Algorithmus oder von Hyperparametern des Algorithmus kann ein Tuning vollzogen werden, um die Werte der Performance-Metriken zu verbessern.

Die Durchführung des Trainings des Agenten obliegt dem Data Scientist, unter eventuellem Einbezug von Software-Entwicklern bei Notwendigkeit einer Anpassung des Lernsignals in der Simulationsumgebung.

Stimmen die Performance-Messungen mit den gewünschten Zielwerten überein, kann der trainierte und angepasste Agent in die Testphase zur Gewährleistung der Sicherheit übergeben werden.

6.3.3.4 Testphase

Die grundlegende Erfüllung der Performance-Metriken gibt lediglich Aufschluss darüber, wie genau der Algorithmus die definierten Ziele verfolgt, jedoch nicht, ob alle Schritte zur Erreichung dieser Ziele legitim stattgefunden haben.

Grundsätzlich ist eine Simulation eine approximierende Abbildung der Realität, welche der realen Welt in einem gewissen Maße nahe kommen soll, aber diese nie exakt abbilden wird (vgl. [Sutton/Barto 2018, 476]). Ebenso ist es mit der Definition der Ziele für den Agenten. Diese werden so genau wie möglich und nötig definiert, können jedoch immer ein Verhalten

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

des Agenten nach sich ziehen, welches nicht explizit gewünscht ist aber legitim nach Art und Weise der Definition der Ziele ist (vgl. [Leike et al. 2018, 11]).

Um diese Effekte des ungewollten Verhaltens zur Zielerreichung zu verhindern, können Tests, Verhaltensanalysen und formale Verifikationen durchgeführt werden. Während die Tests die grundsätzlichen Auswirkungen des Fehlverhaltens auf die definierten Sicherheitsmetriken aufzeigen sollen, kann mithilfe von Verhaltensanalysen sowie formalen Verifikationen das Verhalten des Agenten näher betrachtet werden, um Anpassungen durchführen zu können.

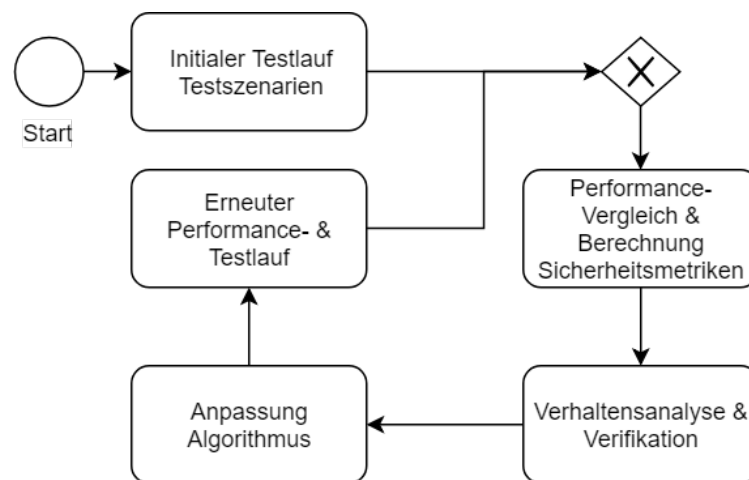


Abbildung 45 Übersicht Testphase Agentensicherheit

Wie auch schon in Abbildung 36 dargestellt, wird in Abbildung 45 das Vorgehen beim Test der Agenten nochmals hervorgehoben. In einem initialen Testlauf werden die Testszenarien (siehe Absatz 6.3.2.7) durchlaufen und deren Performance-Metriken bestimmt. Im Vergleich, mit den bestmöglichen Performance-Messwerten aus der vorausgegangenen Trainingsphase, können die Sicherheits-Metriken berechnet werden. Weitere Sicherheits-Metriken, wie bspw. Intervallgrenzen (vgl. [Gowal et al. 2018, 3f.]), können je nach Definition zusätzlich in die Bemessung einbezogen werden. Daraufhin kann mithilfe verschiedener Techniken zur Betrachtung und Bewertung des Verhaltens der Agenten eine Analyse über die Handlungsketten durchgeführt werden, was in einer Anpassung des Algorithmus bzw. des Lernverhaltens mündet. Dabei ist es auch möglich, mehrere Tests verkettet und simultan durchzuführen, um somit eine Test Pipeline zu erstellen.

Diese Iteration wird solange durchlaufen, bis die Sicherheits-Messungen zufriedenstellende Ergebnisse liefern. Bei den Anpassungen ist ebenfalls auf die Performance-Metriken zu achten, da diese ggf. ebenfalls nachgebessert werden müssen.

Durchgeführt vom Data Scientist, ergibt sich nach erfolgreicher Durchführung dieser Aktivität ein performanter, zielorientierter und auf Sicherheit getesteter Agent, der für den Realtest (siehe Absatz 6.2.3.3) bereitgestellt werden kann.

6.3.4 Deployment-Phase

In der Deployment-Phase wird die Bereitstellung des trainierten und getesteten Agenten-Prototyps beschrieben. Ziel hierbei ist die Zugänglichmachung des Agenten als technischer Service für den Einsatz im Realtest (siehe Absatz 6.2.3.3) des Hauptprozesses, sowie für verschiedene andere technische Services, die hierauf zugreifen sollen. Abbildung 46 stellt die Deployment-Phase mit ihrem Prozessschritt dar.

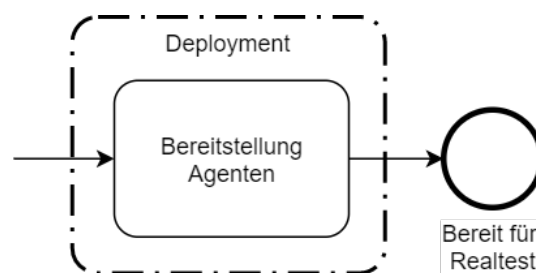


Abbildung 46 Prozessschritt der „Deployment-Phase“ im Sub-Prozess Algorithmenentwicklung

Wie aus der obenstehenden Abbildung 46 ersichtlich, besteht die Phase der Bereitstellung ausschließlich aus einer Aktivität, der Bereitstellung des Agenten Prototyps. Dies beinhaltet die Definition von technischen Service-Schnittstellen, sofern diese noch nicht definiert oder automatisiert erstellt wurden, die Anwendung des definierten Release-Prozesses in der Enforcement-Strategie und die eigentliche Zugänglichmachung für die technischen Konsumenten des Service.



Abbildung 47 Übersicht der Aktivitäten für die Deployment-Phase in der Algorithmenentwicklung

Abbildung 47 stellt die Rollen, Techniken und Ergebnisse für den Prozessschritt der Bereitstellung des Agenten-Prototyps noch einmal kurz zusammengefasst dar.

Nachfolgend wird die Aktivität der Deployment-Phase näher beschrieben.

6.3.4.1 Bereitstellung Prototyp Agenten

Die Aktivität der Bereitstellung des Agenten-Prototyps dient der Zugänglichmachung des Agentens, in Form eines technischen Service, zur Nutzung in der Realtest-Aktivität und später zur Nutzung im finalen Produkt des Vorhabens. Um die Verfügbarmachung durchführen zu können, sind verschiedene Schritte nötig.

Zu Beginn steht die Definition eines Release-Prozesses, der eventuelle automatisierte oder manuelle Quality-Gates sowie Rollen und Autorisierungen beschreibt. Dies wurde bereits während der Erstellung der Enforcement-Strategie (siehe Absatz 6.3.2.3) durchgeführt und kann entsprechend genutzt werden. Ein weiterer Aspekt, der in diesem Zusammenhang zu betrachten ist, ist die Definition des Agenten als technischer Service und die bereits im Absatz 6.2.2.3 definierte Integration dieses Service in die Ziellandschaft.

Als technischer Service muss das Service-Level Agreement, also die Dienstgütevereinbarung des technischen Service, definiert werden. Dies kann mit dem Projektleiter und den dafür relevanten Stakeholdern, wie bspw. den IT-Leiter, geschehen.

Ein weiterer Punkt ist die Schnittstellendefinition des Service zur Integration in die Zielumgebung. Dies kann durch die Systemarchitekten, unter Einbezug des Data Scientist, geschehen, die bereits für bestehende oder zu entwickelnde Services die entsprechenden Rahmenbedingungen konzeptioniert haben. In manchen Fällen bringen technische Lösungen aus kommerziellen oder nicht-kommerziellen Bereichen bereits automatisierte Schnittstellen für eine solche Art von Services mit. In diesen Szenarien gibt es keine Notwendigkeit für eine explizite Definition dieser Schnittstellen.

Je nach Systemdefinition und Entwicklungsvorgehen, steht ein DevOps-Prozess zur eigentlichen Bereitstellung des technischen Service zur Verfügung.

Das Ergebnis dieser Aktivität ist ein bereitgestellter technischer Service mit definierten Schnittstellen für die Nutzung durch weitere Services.

6.4 Überblick Ergebnismodell der Methode

Das Ergebnismodell ist die Summe aller Ergebnisse aus den einzelnen Aktivitäten im Vorgehensmodell. Zu den Ergebnissen zählen sowohl Dokumente als auch Software-Implementierungen, da letztere durch den Quellcode ebenfalls als Dokumente gelten. Ergebnisse sind somit der Output von Aktivitäten, wie es auch in den Anforderungen beschrieben wurde, und dienen anderen Aktivitäten als Input. Einige Dokumente werden im

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Laufe des Vorgehens detaillierter ausgearbeitet oder fließen in die Erarbeitung anderer Dokumente ein. Tabelle 13 stellt für jede Aktivität die zu erarbeitenden Einzelergebnisse vor.

Tabelle 13 Ergebnismodell der Methode

<i>Analysephase – Hauptprozess Vorgehensmodell</i>	
Ideen und Ziele definieren	Ideen- u. Zielbeschreibung
Analyse Patientenalltag	Anwendungsfalldiagramm
Analyse Rahmenbedingungen	Anwendungsfalldiagramm, Anforderungsspezifikation
Analyse Therapie & Adhärenz	Anforderungsspezifikation
Machbarkeit analysieren	Entscheidungsempfehlung, Machbarkeitseinschätzung, SWOT-Analyse
<i>Entwurfsphase – Hauptprozess Vorgehensmodell</i>	
Definition Datenschutzerfordernungen	Datenschutzerfordernungen, Datenschutzfolgeabschätzung
Definition Systemanforderungen	Anforderungsspezifikation
Definition & Design Systemarchitektur	Systemarchitekturmodell, Testplan
Definition Transformationsstrategie	Integrations- und Rollout-Konzept, Prozessmodell
Definition Metriken zur Erfolgsmessung	Kennzahlenkatalog
Recherche & Auswahl Hardware / Software	Software- & Hardwarebeschaffungsliste
<i>Entwicklungs- und Testphase – Hauptprozess Vorgehensmodell</i>	
Umsetzung Hardware & Software	Prototyp
Umsetzung Algorithmen	Diverse – siehe unterer Abschnitt
Realtest Prototyp	Testprotokoll
<i>Deployment-Phase – Hauptprozess Vorgehensmodell</i>	
Schulung Anwender	Schulungskonzept
Definition Monitoring Metriken	Maßnahmenkatalog, Kennzahlenkatalog
<i>Analysephase – Teilprozess Algorithmenentwicklung</i>	
Zielbestimmung durchführen	Anforderungsspezifikation
Datenanalyse durchführen	Datenquellliste
Konkretisierung Performance- & Sicherheitsmetriken	FMEA, Kennzahlen
<i>Entwurfsphase – Teilprozess Algorithmenentwicklung</i>	
Datenselektion und Preprocessing	Datenmodell
Spezifikation Umgebungssimulation	Anforderungsspezifikation
Enforcement & Recovery Strategie	Maßnahmenkatalog
Algorithmenspezifikation	Anforderungsspezifikation
Entwurf Gesamtmodell	Gesamtmodellspezifikation, Agentenmodell
Definition Infrastruktur-Anforderungen	Systemarchitekturmodell
Entwurf Testplanung	Testplan

Implementierungs- und Testphase – <i>Teilprozess Algorithmenentwicklung</i>	
Entwicklung Simulation	Prototyp
Entwicklung Agenten	Prototyp
Trainingsphase	Ergebnisbericht
Testphase	Testprotokoll
Deployment-Phase – <i>Teilprozess Algorithmenentwicklung</i>	
Bereitstellung Prototyp Agent	Technischer Service

Je nach Projektgröße und Zielsetzung des Projektes ist individuell zu entscheiden, welche Ergebnisdokumente und Ergebnisse vorliegen müssen. Die in Tabelle 13 dargestellte Auflistung ist lediglich eine Blaupause, welche die idealerweise zu produzierenden Ergebnisse bezeichnet, diese jedoch nicht gänzlich abschließt oder fest vorschreibt.

Im Folgenden werden die einzelnen Ergebnisse nach den einzelnen Phasen gegliedert und detailliert beschrieben.

6.4.1 Analysephase

Der Unterabschnitt Analysephase soll die sowohl im Haupt- als auch im Teilprozess entwickelten Ergebnisse beschreiben.

Ideen- u. Zielbeschreibung

Die Ideen- und Zielbeschreibung ist eine Definition der grundlegenden Ideen und Ziele für das Vorhaben. Diese dienen der Definition der Anwendungsfälle und der Anforderungen sowie der späteren Erfolgsmessung des Vorhabens.

Anwendungsfalldiagramm

Ein Anwendungsfalldiagramm beschreibt die nötigen Schritte, um ein definiertes Resultat zu erzeugen. Aufgaben und Akteure werden miteinander verbunden, um deren Zugehörigkeit zu definieren und ggf. hierarchisch Definitionen von Aufgaben zu modellieren. Somit können Anwendungsfälle auf ihre einzelnen Akteure und Tätigkeiten heruntergebrochen werden (vgl. [Seemann/Gudenberg 2000, 18ff.]).

Anforderungsspezifikation

Die Anforderungsspezifikation ist eine Liste an Anforderungen, die das zu entwickelnde Zielsystem beschreibt. Anforderungen sind in der Regel verbal beschrieben und können unterstützend durch Dokumente (wie Diagramme etc.) erweitert werden. Die hierin enthaltenen Anforderungen können funktionale als auch nicht-funktionale

Rahmenbedingungen für die Umsetzung des Zielsystems sein (vgl. [Bourque/Fairley 2014, 42]).

Entscheidungsempfehlung

Die Entscheidungsempfehlung beinhaltet eine Zusammenfassung der Machbarkeitseinschätzung und der Risikoanalyse sowie eine Empfehlung des einbezogenen Fachpersonals, ob die Durchführung des Projektes unter den gegebenen Rahmenbedingungen möglich ist. Die Entscheidungsempfehlung dient als Management Summary, auf dessen Basis der Auftraggeber die Durchführung beauftragt, Änderungen anordnet oder das Vorhaben nicht durchführt.

Machbarkeitseinschätzung

Die Machbarkeitseinschätzung resultiert aus der Machbarkeitsanalyse und gibt Aufschluss über die Durchführbarkeit des Vorhabens unter den gegebenen Rahmenbedingungen.

SWOT-Analyse

Die Strength, Weaknesses, Opportunities and Threats Analyse, kurz SWOT-Analyse, ist sowohl eine Technik als auch ein Ergebnis, welche die vier genannten Punkte für das konkrete Vorhaben definieren soll. Als Ergebnis entsteht eine vierteilige Darstellung der genannten Punkte, welche zu vermeiden oder zu verstärken bzw. zu halten sind (vgl. [Kotler/Keller 2012, 48ff.]).

Datenquellliste

Die Datenquellliste beinhaltet alle zur Verfügung stehenden Datenquellen, eine Beschreibung dieser sowie der darin enthaltenen Daten und deren Qualität. Sie soll als kurzer Bericht über die Datenanalyse dienen und kann ggf. auch automatisiert erstellt werden. Zusätzlich kann diese definierte Maßnahmen beinhalten, wie z. B. die Qualität oder andere Eigenschaften der Daten verändert werden können oder müssen.

FMEA

Die Fehlermöglichkeits- und Einflussanalyse, kurz FMEA, ist wie die SWOT-Analyse sowohl eine Technik als auch ein Ergebnisdokument. Das Ergebnisdokument der FMEA beinhaltet alle möglichen identifizierten Fehlerfälle mit den Folgen und den Ursachen dieser, deren Bewertung hinsichtlich Eintrittswahrscheinlichkeit, Häufigkeit und Schwere der Auswirkung sowie den möglichen Maßnahmen zur Reduzierung der drei

Bewertungskriterien. Die FMEA kann als Messinstrument für die Sicherheit und Qualität eines Systems oder Prozesses genutzt werden (vgl. [Pfeufer 2015, 5ff.]).

Kennzahlen

Kennzahlen sind quantifizierbare Größen, die messbar sind und der Zielerreichung bzw. zur Zieleinhaltung dienen. Mit diesen können Ziele verfolgt und deren Erfüllungsgrad dargestellt werden. Kennzahlen können sich wiederum aus verschiedenen einzelnen Kennzahlen zusammensetzen und ggf. auch nur aufschlussreich über einen gewissen Sachverhalt unter Einbezug anderer Kennzahlen sein (vgl. [Karlstedt 2014, 24ff.]). Ein Kennzahlenkatalog, bzw. eine Kennzahlenliste, zählt mehrere Kennzahlen auf.

6.4.2 Entwurfsphase

Der Unterabschnitt Entwurfsphase soll die sowohl im Haupt- als auch die im Teilprozess entwickelten Ergebnisse beschreiben, sofern diese nicht bereits in einer vorausgehenden Phase beschrieben wurden.

Datenschutzanforderungen

Die Datenschutzanforderungen sind eine Liste an relevanten und einzuhaltenden rechtlichen Datenschutzrichtlinien und -gesetzen, die auf das Zielsystem zutreffen. Diese können sowohl nationale als auch internationale Anforderungen beinhalten. Das Dokument dient als Grundlage für die Definition von Maßnahmen in den Anforderungen zur Einhaltung dieser Richtlinien und Gesetze.

Datenschutzfolgeabschätzung

Die Datenschutzfolgeabschätzung kann im Rahmen einer Datenschutzbewertung erstellt werden. Der Begriff beschreibt sowohl eine Technik zur Bewertung als auch das entstehende Ergebnisdokument. Diese Datenschutzfolgeabschätzung ist gemäß Art. 35, DSGVO von Art und Umfang klar definiert und beinhaltet mindestens die Notwendigkeit der Anwendung der Techniken, die Risiken für die Daten der Betroffenen, eine Beschreibung der Datenverarbeitungsvorgänge und Abhilfemaßnahmen für die Bewältigung der entstehenden Risiken durch die Anwendung der Techniken zur Datenverarbeitung (vgl. [European Union 2016]).

Systemarchitekturmodell

Das Systemarchitekturmodell beinhaltet alle Komponenten (Software- als auch Hardware-seitig) der Systemarchitektur sowie die Schnittstellenbeschreibungen zwischen den einzelnen Komponenten (vgl. [Beauftragter der Bundesregierung für die Informationstechnik 2009, 146ff.]).

Testplan

Der Testplan beinhaltet alle für die Durchführung der Testszenarien benötigten Schritte und Maßnahmen. Dazu zählen die Testszenarien an sich, der Ablauf des Testprozesses sowie die dazu nötigen Ressourcen und Abfolgen (vgl. [Sommerville 2011, 208ff.]). Ein Testszenario wiederum beinhaltet die einzelnen Testanwendungsfälle für einen bestimmten Sachverhalt. Beide sind im Testplan inkludiert.

Integrationskonzept

Das Integrationskonzept beinhaltet alle Informationen über zu verändernde Prozesse, zu schaffende oder zu ändernde Stellen oder Stellenbeschreibungen sowie über technisch notwendigen Schnittstellen zur Ankopplung des Zielsystems an die bestehenden Alt-Systeme.

Rollout-Konzept

Das Rollout-Konzept beschreibt das Vorgehen und die Rahmenbedingungen sowie die benötigten Ressourcen und Reihenfolgen zum Verteilen des zu entwickelnden Zielsystems im Unternehmen bzw. bei den Kunden (vgl. [Gadatsch 2010, 360f.]). Hierin sind alle Aufgaben beschrieben, die einen ausfallsicheren Übergang des Zielsystems in den Regelbetrieb gewährleisten.

Prozessmodell

Ein Prozessmodell ist eine strukturierte Abfolge von Aktivitäten. Die Aktivitäten werden in einem deterministischen Ablauf durchgeführt und ggf. mit Bedingungen oder weiteren externen Effekten, wie eingehende Nachrichten als Auslöser, verbunden. Die Beschreibung eines Prozessmodells kann sowohl textuell als auch grafisch, durch die Anwendung von Modellierungstechniken, durchgeführt werden (vgl. [Gadatsch 2010, 82f.]).

Software- / Hardwarebeschaffungsliste

Die Software- und Hardware-Beschaffungsliste ist eine Auflistung aller benötigten Komponenten, die für das Zielsystem zu beschaffen sind. Diese können in der notwendigen Anschaffung am Markt sowohl frei als auch kommerziell zu erstehen sein. Zu den Informationen aus der Liste zählen neben der Beschreibung der jeweiligen Komponente die Bezugsquelle, die Anzahl und eine eventuelle Kostenkalkulation.

Datenmodell

Ein Datenmodell beschreibt die Strukturierung und Organisation der Daten untereinander und über Datenquellen hinaus. Es bildet ein Schema, wie die Daten gehalten und abgelegt werden sollen. Gleichwohl beschreibt es kurz die vorhandenen Daten und das erstellte Schema. Ein Datenmodell kann, begleitend mit einer kurzen Beschreibung visuell mithilfe von Modellierungstechniken, aufbereitet werden (vgl. [Ponniah 2007, 5f.]).

Maßnahmenkatalog

Der Maßnahmenkatalog beinhaltet eine Liste von Maßnahmen zu einem bestimmten Sachverhalt und deren Beschreibung. Die jeweilige Beschreibung beinhaltet das Vorgehen der Maßnahme und deren Notwendigkeit. Weiterhin ist jede Maßnahme mit mindestens einem Auslöser für die Notwendigkeit ihrer selbst verbunden und beschrieben. Maßnahmen können beliebig umfangreich sein und ggf. Rollen, Arbeitsschritte oder ganze Projekte beschreiben oder verlinken.

Gesamtmodellspezifikation

Die Gesamtmodellspezifikation beschreibt den Software-Agenten mit all seinen Komponenten aus dem definierten Gesamtmodell. Diese beinhaltet die Simulationsumgebung, die Zielfunktion, den zu nutzenden Algorithmus, den Rückgabestatus der Simulation, eventuelle Effekte, die vom Agenten ausführbaren Aktionen und dessen interne Struktur sowie die externen einzubeziehenden Datenquellen.

Agentenmodell

Das Agentenmodell ist eine visuell modellierte, aussagekräftige Darstellung der internen Struktur des Agenten und dessen Interaktions-Schnittstellen mit der Umwelt und ggf. weiteren Agenten. Die zu modellierenden Informationen, können je nach Anwendungsfall und Notwendigkeit variieren. Das Modell soll zum einfacheren Verständnis und zur besseren Kommunikation, bzw. zum Austausch zwischen Fachexperten, dienen.

6.4.3 Entwicklungs- & Testphase

Im Unterabschnitt der Entwicklungs- und Testphase sollen die sowohl im Haupt- als auch im Teilprozess entwickelten Ergebnisse beschrieben werden, sofern diese nicht bereits in einer vorausgehenden Phase beschrieben worden sind.

Prototyp

Ein Prototyp ist ein erster Entwurf einer Software bzw. eines Systems, um dessen Funktionalität so früh wie möglich im Entwicklungsprozess testen zu können. Er dient zur Validierung der umgesetzten Anforderungsspezifikationen und kann iterativ stetig erweitert und angepasst werden. Zudem dienen Prototypen zur Validierung des Systems unter realen Bedingungen schon in einer frühen Phase der Umsetzung, um ggf. Fehlentscheidungen beim Design und der Konzeption des Systems erkennen und beheben zu können (vgl. [Sommerville 2011, 45f.]).

Testprotokoll

Ein Testprotokoll beinhaltet die Ergebnisse der einzelnen Testszenarien und die damit verbundenen Abweichungen oder Einhaltungen im Abgleich zu den erforderlichen Zielwerten. Auf Basis des Testprotokolls können bei Abweichungen von den Idealwerten weitere Iterationen in der Entwicklung sowie weitere ergänzende Maßnahmen geplant werden. Das Testprotokoll beinhaltet zudem die Fehlerberichte aus den Testszenarien, um diese für die Entwicklung als Eingabe zu liefern (vgl. [Sommerville 2011, 208ff.]).

Ergebnisbericht

Der Ergebnisbericht enthält eine Auswertung über die Leistungsfähigkeit des Agenten. Dieser beinhaltet die Darstellung des Verlaufs der Leistungsfähigkeit des Agenten als auch eine zum Anwendungsfall passende Beschreibung. Dies können Metriken, wie die Genauigkeit oder die Anzahl der Abweichungen vom Idealvorgehen, sein. Eine genaue Definition der Kennzahlen, ist im Projekt vorab zu treffen. Das Augenmerk dieses Berichtes liegt auf der Validierung der Performance des Agenten gegenüber dem Auftraggeber sowie auf den vergleichbaren Austausch mit Fachexperten.

6.4.4 Deployment- & Betriebsphase

Im letzten Phasenabschnitt der Deployment-Phase des Haupt- und Teilprozesses werden die verbleibenden Ergebnisse beschrieben, die noch nicht in den vorangegangenen Phasen aufgetreten bzw. entwickelt wurden.

Schulungskonzept

Ein Schulungskonzept beinhaltet die benötigten Schulungen für die späteren Anwender des Zielsystems. Dabei ist das Schulungskonzept auf die entsprechenden Anwendergruppen (Techniker, Endanwender, anderes Fachpersonal) anzupassen und ggf. zu untergliedern. Darin enthalten sind die Schulungsbedarfe und -anforderungen für die einzelnen Anwendergruppen. Weiterhin beschreibt das Schulungskonzept den gesamten Schulungsablauf mit den benötigten Ressourcen, Tätigkeiten und Methoden zur Umsetzung der Anwenderschulungen (vgl. [Goltsche 2006, 123ff.]).

Technischer Service

Ein technischer Service soll abgegrenzt von einer Dienstleistung einen Softwaredienst darstellen. Dieser ist über wohl definierte Schnittstellen für andere Dienste erreichbar und liefert zu einer definierten Eingabe eine definierte Ausgabe. Wo dieser technische Service bereitgestellt wird, ist abhängig vom Projekt und den daraus sich ergebenden Anforderungen.

6.5 Überblick Technikmodell der Methode

Im Vorgehensmodell der Methode werden die groben Aktivitäten beschrieben und wie diese aufeinander aufbauen. Jede Aktivität kann durch eine oder mehrere Techniken durchgeführt werden, um das benötigte Ergebnis zu erzielen. Techniken geben konkrete Abläufe und Handlungen zur Erreichung eines konkreten Zieles vor. Damit beschreiben Techniken, im Gegensatz zu den Aktivitäten aus dem Vorgehensmodell der Methode, das Vorgehen im Kleinen (vgl. [Gutzwiller 1994, 11ff.]).

Je nach Aktivität können verschiedene Techniken zur Anwendung kommen. Tabelle 14 stellt einen Überblick über die je nach Aktivität angewandten Techniken dar. Diese werden oftmals in mehreren Aktivitäten genutzt, wodurch diese nur bei der ersten Verwendung beschrieben werden.

Tabelle 14 Übersicht Technikmodell

<i>Analysephase – Hauptprozess Vorgehensmodell</i>	
Ideen und Ziele definieren	Kreativitätstechniken, Workshops
Analyse Patientenalltag	Beobachtungstechniken, Befragungstechniken
Analyse Rahmenbedingungen	Befragungstechniken, Recherchetechniken
Analyse	Befragungstechniken, Recherchetechniken

6 Methode zur Konzeption präskriptiver Patienten- und Therapiesysteme

Machbarkeit analysieren	Umgebungs- und Umfeldanalyse, SWOT-Analyse, Befragungstechniken, Recherchetechniken
<i>Entwurfsphase – Hauptprozess Vorgehensmodell</i>	
Definition Datenschutzerfordernungen	Befragungstechniken, Recherchetechniken
Definition Systemanforderungen	Requirements Engineering
Definition & Design Systemarchitektur	Modellierungstechniken, Recherchetechniken
Definition Transformationsstrategie	Befragungstechniken, Modellierungstechniken, Recherchetechniken
Definition Metriken zur Erfolgsmessung	Befragungstechniken, Workshops
Recherche & Auswahl Hardware / Software	Befragungstechniken, Recherchetechniken, Workshops
<i>Entwicklungs- und Testphase – Hauptprozess Vorgehensmodell</i>	
Umsetzung Hardware & Software	Software Engineering
Umsetzung Algorithmen	Diverse – siehe unterer Abschnitt Teilprozess Algorithmenentwicklung
Realtest Prototyp	Feldexperiment, Software Engineering
<i>Deployment-Phase – Hauptprozess Vorgehensmodell</i>	
Schulung Anwender	Schulungstechniken
Definition Monitoring Metriken	Befragungstechniken, Workshops, Recherchetechniken
<i>Analysephase – Teilprozess Algorithmenentwicklung</i>	
Zielbestimmung durchführen	Recherche-, Befragungstechniken
Datenanalyse durchführen	Befragungstechniken, Recherchetechniken, Datenanalysetechniken
Konkretisierung Performance- & Sicherheitsmetriken	Befragungstechniken, Recherchetechniken, Kreativitätstechniken, FMEA
<i>Entwurfsphase – Teilprozess Algorithmenentwicklung</i>	
Datenselektion und Preprocessing	Datenmodellierung
Spezifikation Umgebungssimulation	Requirements Engineering
Enforcement & Recovery Strategie	Recherchetechniken, Befragungstechniken
Algorithmenspezifikation	Requirements Engineering, Recherchetechniken
Entwurf Gesamtmodell	Recherchetechniken, Modellierungstechniken
Definition Infrastruktur-Anforderungen	Recherchetechniken, Befragungstechniken
Entwurf Testplanung	Workshops, Befragungstechniken, Testplanung
<i>Implementierungs- und Testphase – Teilprozess Algorithmenentwicklung</i>	
Entwicklung Simulation	Software Engineering
Entwicklung Agenten	Software Engineering
Trainingsphase	Algorithmentraining
Testphase	Algorithmentests
<i>Deployment-Phase – Teilprozess Algorithmenentwicklung</i>	

Bereitstellung Prototyp Agent	DevOps Techniken
-------------------------------	------------------

Im Nachfolgenden werden die in der Tabelle 14 genannten Techniken, gegliedert nach den benannten Phasen, beschrieben.

6.5.1 Analysephase

In diesem Unterabschnitt werden die Techniken des Haupt- und Teilprozesses detailliert beschrieben.

Kreativitätstechniken

Kreativitätstechniken sollen bei der Ideenfindung unterstützen. Sie sollen es ermöglichen, von strukturierten Denkweisen auf chaotische und somit kreative wechseln zu können. Hierbei gibt es verschiedene Techniken für kreative Denkweisen. Bekannte Vertreter sind zum Beispiel das Brainstroming oder das Brainwriting als sogenannte Assoziationstechniken. Ziel dieser Techniken ist die Verknüpfung von Ideen und Gedanken, was ohne die Anwendung dieser Techniken eventuell nicht geschehen würde. Kreativitätstechniken werden vorrangig in einer Form von Gruppenarbeit durchgeführt, um möglichst unterschiedliche Eingaben und Denkweisen in den Lösungsansatz mit einzubeziehen. Mithilfe von Ideenvisualisierungen vor der Gruppe können diese diskutiert und ausgewertet werden (vgl. [Becker et al. 2018]).

Datenanalysetechniken

Techniken der Datenanalyse werden oft unter dem Begriff der explorativen Datenanalyse verstanden und bilden ein Teilgebiet der Statistik ab. Ziel hierbei ist es, einen Einblick in die aktuelle Struktur und die Verteilungen der vorliegenden Daten zu erhalten. Ferner ermöglicht die Technik einen Einblick in die Verwertbarkeit der Daten für den angestrebten Nutzen (vgl. [Polasek 1994, 3ff.] & [Becker et al. 2016, 27ff.]).

Befragungstechniken

Befragungstechniken werden zur Erkenntnisgewinnung genutzt. Eine definierte Menge an Teilnehmer wird nach einer bestimmten Technik befragt und die gegebenen Antworten auf die Fragen werden daraufhin ausgewertet. Diese Techniken können qualitativ oder quantitativ sein. Qualitative zielen auf die Motivation, Ansicht oder auf bestimmtes Wissen von Experten ab. Quantitativ angelegte Techniken versuchen dagegen, über eine größere Menge an Befragten und große erhobene Datensätze, eine möglichst aussagefähige

Auswertung und Verallgemeinerung der Antworten über statistische Verfahren zu erreichen. Bekannte Techniken in der Befragung sind z. B. Interviews. Bei diesen gibt es wiederum Unterschiede in der Vorbereitung, Durchführung und Auswertung. Je nach Zielfragestellung ist die richtige Variante der Technik zu nutzen. So bieten Experteninterviews einen speziellen Leitfaden zur Erarbeitung des Interviews für die Befragung von Fach- und Domänenexperten (vgl. [Pepels/Bruns 2008, 203ff.] & [Preißner 2008, 35ff.]).

Beobachtungstechniken

Beobachtungstechniken sind visuelle Auswertungen, die durch einen Externen von einem Prozess, Verhalten etc. erstellt werden. Zum Beispiel können Prozesse hierbei objektiv von außenstehenden geschulten Beobachtern betrachtet werden, wodurch eventuelle Probleme oder Verbesserungsmöglichkeiten auffallen bzw. genauere Beschreibungen der Prozesse möglich werden, die unter normalen Umständen den direkt involvierten Prozessteilnehmern nicht mehr auffallen. Beobachtungstechniken untergliedern sich wiederum in unterschiedliche Varianten. Je nach Ziel der Beobachtung, ist die passende Variante zu wählen. Für Prozessbetrachtungen bieten sich bspw. teilnehmende Beobachtungen an, wo der Beobachter beim Prozessgeschehen dabei ist. Nicht-teilnehmende Beobachtungen werden z. B. durch technische Geräte aufgenommen und erfordern somit keine direkte Präsenz (vgl. [Preißner 2008, 41f.]).

Recherchetechniken

Recherchetechniken dienen der strukturierten Suche von spezifischen Informationen aus verschiedenen Quellen. Quellen können hierbei Zeitschriften, Bücher, das Internet etc. sein. Im wissenschaftlichen Rahmen bietet sich bspw. die Recherchetechnik von vom Brocke (vgl. [Vom Brocke et al. 2009, 2213ff.]) an. Teilweise kann diese Technik auch auf beliebige Quellen und Verzeichnisse angewandt werden, ohne dabei explizit nach wissenschaftlichen Publikationen zu suchen. Recherchen können von einer oder mehreren Personen parallel erfolgen und ggf. strukturiert in einer gemeinsamen Datenbank als Wissensbasis abgelegt werden.

Umfeldanalyse

Die Umfeldanalyse wird auch Stakeholderanalyse genannt. Ziel dieser Technik ist die Analyse der Personengruppen, die auf das Projekt und dessen Gelingen Einfluss nehmen können. Sie zählt mit zur Risikoanalyse und kann parallel zu dieser durchgeführt werden (vgl. [Pfetzing/Rohde 2009, 207ff.]).

SWOT-Analyse

Die Strength, Weaknesses, Opportunities and Threats Analysis, kurz SWOT-Analyse, dient als Risikoanalyse und versucht die Stärken, Schwächen, Chancen und Risiken eines Projektes bzw. eines Vorhabens aufzudecken. Die gewonnenen Erkenntnisse können zu einer effektiveren Lösungsfindung in Bezug auf auftretende Schwächen, zur Vermeidung oder Reduktion von Risiken sowie zur Identifizierung und Nutzung von Chancen und Ausbau vorhandener Stärken genutzt werden. Die Technik ist gleichzeitig ein Ergebnis im Ergebnismodell (siehe Unterabschnitt 6.4.1). Sie kann in einer Gruppenarbeit oder von einer Person allein erstellt werden (vgl. [Kotler/Keller 2012, 48ff.]).

Requirements Engineering

Mit dem Requirements Engineering, also die Anforderungsaufnahme, wird eine strukturierte Aufnahme von Anforderungen an ein bestimmtes System oder einen bestimmten Sachverhalt versucht. Hierzu zählen die Ermittlung der Anforderungen, sowie die Prüfung, Spezifikation und Dokumentation dieser Anforderungen. Für die Durchführung der Anforderungsaufnahme für Software-Systeme existieren eine Vielzahl an Regelungen, Best Practices und Vorgehensmodellen, die für diese Technik eingesetzt werden können (vgl. [Sommerville 2011, 99f.] & [Rupp et al. 2009, 94ff.]). Auch bereits benannte Techniken können für die Anforderungsaufnahme von Nutzen sein. Bspw. bietet sich die Recherche und die Befragung für die Anforderungsidentifikation an.

Fehlermöglichkeits- und Einflussanalyse

Die Fehlermöglichkeits- und Einflussanalyse, kurz FMEA, beschreibt eine strukturierte Technik zur Risikoaufdeckung, -bewertung und Maßnahmenplanung. Es ist zudem auch ein Monitoring mit der FMEA der bewerteten Risiken ist möglich. Dabei werden zu einem bestimmten Prozess oder Produkt / System die möglichen Fehler und ihre Quellen identifiziert (mit beliebigen anderen Techniken unterstützend möglich) und eine Bewertung dieser Fehlerfälle durchgeführt. Diese werden hinsichtlich Eintritts- und Entdeckungswahrscheinlichkeit sowie der Schadenshöhe bewertet. Daraufhin können Maßnahmen zur Reduktion der Fehler und Fehlerquellen aufgrund der vorangegangenen Bewertungen erstellt werden. Nach der Maßnahmendurchführung oder Ermittlung ihres Ausmaßes wird eine erneute Bewertung durchgeführt, um den verbesserten Zustand zu ermitteln. Die FMEA kann zusammen in einer Gruppe von Personen erstellt werden, um

gerade bei der Bewertung der Fehler möglichst heterogene Meinungen einzuholen (vgl. [Pfeufer 2015, 5ff.]).

6.5.2 Entwurfsphase

Dieser Unterabschnitt beschreibt die Techniken des Haupt- und Teilprozesses jeweils für die Entwurfsphase, wenn einzelne Techniken noch nicht im vorherigen Unterabschnitt detailliert beschrieben wurden.

Modellierungstechniken

Modellierungstechniken dienen der Visualisierung durch Grafiken von Prozessen, Strukturen u. Ä. Für die Durchführung von solchen Modellierungen ist der Einsatz von standardisierten Modellierungssprachen möglich. Beispielweise können für solche Modellierungssprachen, die Unified Modeling Language (UML) und die Business Process Model and Notation (BPMN) genannt werden. Diese unterstützen die fachübergreifende Lesbarkeit der Grafik und können somit dem Verständnis und dem Fachaustausch dienen.

Workshops

Ein Workshop ist ein Rahmen, in dem eine Gruppe aus Personen intensiv und unter Einbezug mehrerer Techniken versucht, ein bestimmtes Ziel zu erreichen oder ein bestimmtes Problem zu lösen. Dieser sollte in der Regel von einer Person moderiert werden, welche die Tätigkeiten der Gruppe strukturiert, dokumentiert, zusammenfasst und zum Ziel geleitet. Wie bereits erwähnt, können in Workshops mehrere bereits genannte Techniken eingesetzt werden. Somit ist ein Workshop eine Mischform aus verschiedenen Techniken, welche aber durch die spezielle Form und Organisation der Zusammenkunft separat als eigenständige Technik aufgeführt wird (vgl. [Kaune 2010, 124ff.]).

Datenmodellierung

Datenmodellierung ist eine Technik zur Spezifikation, der vom Zielsystem verwalteten und genutzten Daten. Die Spezifikation kann nach einem statischen Schema oder schemafrei geschehen. Es entsteht eine Ordnung für die Speicherung, Verarbeitung und das Auslesen dieser Daten. Dabei werden nicht nur die Daten für das Zielsystem strukturiert, sondern auch untereinander verknüpft, wodurch sich ein zusammenhängendes Bild von Strukturen und Abhängigkeiten zwischen den verwalteten Daten ergibt. Somit greift das Zielsystem auf eine einheitliche Datenstruktur zu (vgl. [Gadatsch 2017, 4ff.]).

Testplanung

Die Testplanung dient der Erfassung aller notwendigen Tätigkeiten zur Durchführung von Softwaretests. Hierunter zählen die Erstellung eines Ablaufplans für die Durchführung der Tests, die nötigen Testszenarien und die Erstellung der einzelnen Tests sowie die für die Durchführung benötigten Ressourcen (vgl. [Sommerville 2011, 208ff.]).

6.5.3 Entwicklungs- & Testphase

In der Entwicklungs- und Testphase fallen unterschiedliche Techniken zur Durchführung der einzelnen Aktivitäten an. Diese werden in diesem Unterabschnitt, sowohl für den Hauptprozess als auch für den Teilprozess, beschrieben, sofern diese nicht bereits in einem vorherigen Unterabschnitt von Abschnitt 6.5 beschrieben wurden.

Software Engineering

Software Engineering ist eine Disziplin, die alle möglichen Techniken und Vorgehen für die Softwareentwicklung zusammenfasst. In diesem Rahmen werden darunter alle Techniken verstanden, die genutzt werden können, um Softwareprodukte mit oder ohne den Einsatz von Hardware-Komponenten, nach bestimmten Vorgaben, zu erstellen. Hierzu zählen z. B. die Durchführung der Softwareentwicklung und -tests mithilfe von Vorgehensmodellen, die Nachvollziehbarkeit von Funktionen aus den Anforderungen und die Erstellung von eigenständigen Prototypen (vgl. [Sommerville 2011, 7ff.]).

Feldexperiment

Das Feldexperiment soll das zuvor entwickelte System unter Realbedingungen in einer minimal kontrollierten Umgebung testen. Im Unterschied zu Tests in Simulationen oder in Laboren, wird hierbei das Zielsystem unter realen Bedingungen in der späteren Einsatzumgebung getestet. Dabei können viel mehr Faktoren für den Test einbezogen werden, die eine Simulation oder starr definierte Testfälle aufgrund der Komplexität nicht abbilden können (vgl. [Aronson et al. 2011, 46f.]).

Algorithmentraining

Das Algorithmentraining ist ein exploratives Anpassen eines Algorithmus und seiner Hyperparameter zur Erreichung eines bestimmten Zieles. Für das Durchführen dieser Technik ist kein genauer Ablauf nötig. Ziel hierbei ist die Erreichung eines bestmöglichen Wertes für eine Anzahl an Metriken, die die Performance des Algorithmus beschreiben sollen. Die Technik basiert auf starken explorativen und erfahrungsbasierten

Vorgehensweisen und profitiert massiv von der Erfahrung der durchführenden Person. Zur Durchführung gehören u. a. Schritte wie die Parametrisierung von Hyperparametern, die Anpassung von Strukturen oder Datenflüssen und -formaten.

Algorithmtests

Ähnlich dem Algorithmtraining, werden beim Algorithmtest vorab definierte Testfälle am Algorithmus getestet und verglichen, ob die Testresultate auf die definierten Zielvorgaben passen. Zu den Tätigkeiten des Testers zählen hierbei nicht nur die Durchführung und die Protokollierung der Tests, wie es bei Softwaretests teilweise der Fall ist, sondern auch die Anpassung des Algorithmus zur Erreichung der Zielvorhaben der Testszenarien.

6.5.4 Deployment- & Betriebsphase

Als letzter Unterabschnitt im Technikmodell der Methode, werden die Techniken der Deployment-Phase nachfolgend beschrieben.

Schulungstechniken

Schulungstechniken bestehen aus verschiedenen Schritten zur Anwendungsschulung von späteren Nutzern des Zielsystems. Je nach Zielgruppe für die Schulung, müssen andere Methoden und Techniken zur Wissensvermittlung angewandt werden. Hierzu zählt nicht nur die an die verschiedenen Zielgruppen angepasste Wissensvermittlung, sondern auch die Identifizierung des Schulungsbedarfs, die Planung des Ablaufs der Schulung und die dafür benötigte Ressourcenplanung (vgl. [Goltsche 2006, 123ff.]).

DevOps Techniken

Unter DevOps-Techniken werden Techniken verstanden, die zur agilen und kontinuierlichen Auslieferung von Softwareaktualisierungen genutzt werden, um eine möglichst schnelle und kontinuierliche Weiterentwicklung eines Softwareproduktes zu erreichen. Hierbei steht der Entwickler oder Data Scientist im Vordergrund, der selbstbestimmt neue Versionen der von ihm entwickelten Software ausliefern kann. Diese Tätigkeiten beziehen neben automatisierten Tests, Versionierungen und Monitoring-Maßnahmen, das Bereitstellen der Software für Konsumenten der Software mit ein (vgl. [Kim et al. 2016, 1ff.]).

6.6 Überblick Rollenmodell der Methode

Rollen dienen zur Strukturierung des eingesetzten Personals in Gruppen mit bestimmten Eigenschaften, Tätigkeiten und Verantwortlichkeiten. Diese können von einer oder mehreren Personen wahrgenommen werden.

Ein Rollenmodell beschreibt die einzelnen Rollen und deren Verbindung zu den einzelnen Aktivitäten innerhalb des Vorgehensmodells. Dabei können Aktivitäten auch von mehreren Rollen durchgeführt werden, je nachdem, welche Tätigkeiten in der Aktivität benötigt werden.

Je nach Vorhaben und Projekt können Rollen mit einer oder mehreren Personen besetzt werden. Gleichwohl können Rollen sowohl intern als auch durch extern eingekaufte Personen besetzt werden. Je nach Vorhaben können auch einzelne Rollen wegfallen bzw. deren Tätigkeit durch andere vorhandene Rollen übernommen werden.

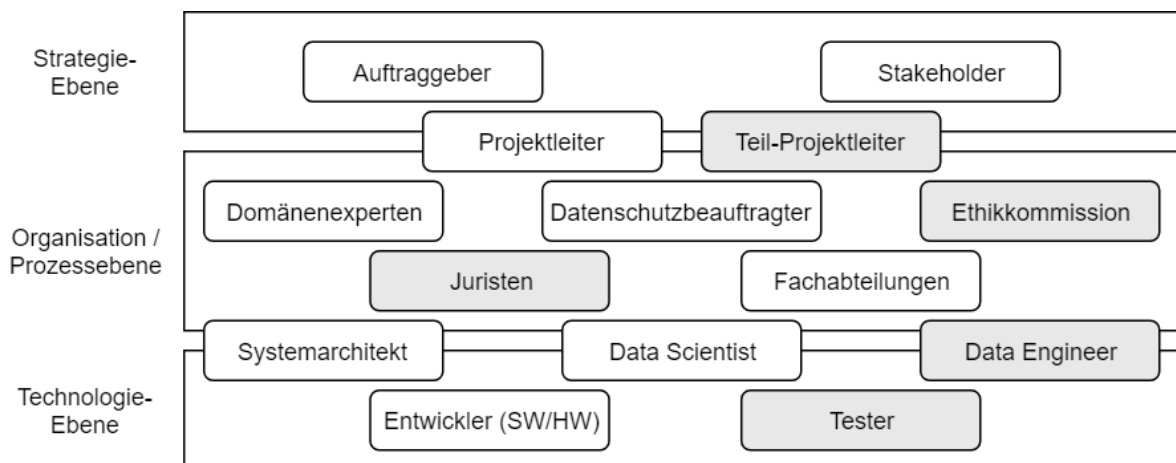


Abbildung 48 Übersicht Rollenmodell nach Ebenen

Abbildung 48 zeigt eine mögliche Strukturierung der Rollen innerhalb eines Projektes. Weiterhin ist die Entbehrlichkeit mancher Rollen (durch eine dunklere Färbung) ersichtlich. Diese optionalen Rollen können je nach Vorhaben und Größe des Vorhabens separat besetzt, in andere Rollen integriert oder ausgelassen werden.

Im Folgenden werden die in der Abbildung dargestellten Rollen näher beschrieben.

Auftraggeber

Der Auftraggeber gibt die Ziele für das Projektvorhaben vor und nimmt die Ergebnisse des Projektes ab. Er setzt den finanziellen und zeitlichen Rahmen fest und wird vom Projektleiter über den Stand des Projektes regelmäßig informiert. Gleichwohl ist der Auftraggeber auch

ein Stakeholder, da er ein bestimmtes Interesse am Gelingen des Projektes hat (vgl. [Alpar et al. 2016, 355]).

Stakeholder

Stakeholder sind alle Personen im Projekt, die ein berechtigtes Interesse am Gelingen und den Ergebnissen des Projektes haben. Somit ist der Begriff Stakeholder ein Oberbegriff für verschiedene Rollen im Projekt und dient der Zusammenfassung dieser (vgl. [Burghardt 2012, 545f.]). Implizierte Rollen sind neben dem Auftraggeber bspw. der Projektleiter, die Fachabteilungen aber auch die Projektmitarbeiter und ggf. Auftraggeber vom Auftraggeber, die nicht direkt im Projekt in Erscheinung treten oder aktiv werden (z. B. Pflegekassen als finanzielle Auftraggeber für Altenheime).

Projektleiter

Der Projektleiter leitet das Vorhaben i. d. R. fachlich, finanziell und disziplinarisch. Er organisiert und leitet das Projekt von der Aufnahme über die Umsetzung der Anforderungen bis hin zur Übergabe des Projektergebnisses in den Regelbetrieb. Er stimmt sich mit dem Auftraggeber ab, berichtet diesem und betreibt Stakeholder-Management. Die Einhaltung der Rahmenbedingungen während des Projektes ist ebenso eine Aufgabe wie auch das fortwährende Kontrollieren der Risiken für das Projekt (vgl. [Alpar et al. 2016, 355]). Je nach Größe und fachlicher Komplexität des Projektes können dem Projektleiter **Teilprojektleiter** untergeordnet werden, die bestimmte Teilprozesse des Projektvorhabens verantworten.

Domänenexperten

Domänenexperten sind Personen, die die Fachkompetenz in das Projektvorhaben aus der Domäne, in der das Vorhaben angesiedelt ist und genutzt werden soll, einbringen. Sie unterstützen das Projekt mit ihrem Fachwissen bei der Anforderungsdefinition und bei der Evaluierung der Projektergebnisse. Sind bestimmte Informationen nicht von Domänenexperten nach oben genannter Definition zu erschließen, kann auch die eigentliche Zielgruppe als Domänenexperten einbezogen werden.

Datenschutzbeauftragter

Ein Datenschutzbeauftragter berät in Datenschutzfragen und dient zur datenschutzkonformen Durchführung des Projektes und Zielergebnisses. Er schätzt das Risiko für datenschutzrechtliche Anforderungen ein und gibt Maßnahmen vor, die eine

rechtlich sichere Umsetzung der Anforderungen gewährleisten. Seine Tätigkeit hat eine fortwährende Notwendigkeit im Projekt, da mit jeder Änderung der rechtlichen Situation oder der Anforderungen im Projekt sich auch die Datenschutzerfordernisse ändern können (vgl. [European Union 2016]).

Ethikkommission

Die Arbeit einer Ethikkommission dient der Beurteilung von Projekt- und Forschungsvorhaben an Menschen, um ethische und rechtliche Rahmenbedingungen zu prüfen und einschätzen zu können. Nach der Vereinbarung von Helsinki, über die Grundsätze für die medizinische Forschung am Menschen, wird geprüft, ob und in wie weit die Würde des Menschen angetastet wird (vgl. [World Medical Association 2013, 1f.]). Eine Ethikkommission muss nicht zwangsläufig aus mehreren Personen bestehen. Auch einzelne Ethiker (z. B. Computerethiker) können zur Abdeckung bestimmter Anforderungen einbezogen werden und bilden einzeln die Ethikkommission (vgl. [Bynum 2018, 3f.]).

Jurist

Ist eine juristisch ausgebildete Person, idealerweise mit einem abgeschlossenen Jurastudium, die einen Überblick und die Expertise für die Prüfung von rechtlichen Rahmenbedingungen und Verträgen in der benötigten Domäne mitbringt. Nicht für jedes Projektvorhaben ist ein Jurist nötig. Dies hängt vom Anwendungsfall ab.

Fachabteilungen

Fachabteilungen sind diejenigen Abteilungen, die beim späteren Ausrollen des Projektergebnisses in den Regelbetrieb dieses in irgendeiner Form nutzen bzw. von diesem in ihrem aktuellen Regelbetrieb beeinflusst oder verändert werden. Sie müssen im Projektvorhaben als Stakeholder mit einbezogen und später für den reibungslosen Betrieb des Zielsystems geschult werden.

Systemarchitekt

Ein Systemarchitekt ist für die Konzeption und das Design der Gesamtsystemarchitektur und das Systemarchitekturmodell verantwortlich. Er definiert Komponenten und Schnittstellen des Systems und bezieht hierzu neben den funktionalen auch die nicht-funktionalen Anforderungen, wie Datenschutz oder Wartbarkeit, mit ein (vgl. [Rempp 2011, 98]).

Data Scientist

Ein Data Scientist beinhaltet mehrere Rollen, die seine Profession beschreiben. Er ist Statistiker, Machine Learning Engineer und Analytiker in einer Person. Er kann alle Tätigkeiten durchführen die nötig sind, um ein Machine Learning Modell zu erstellen. Hierzu zählen die Extraktion und Aufbereitung aller Daten, die Erstellung eines Machine Learning Modells bis hin zur Bereitstellung desselben. Dabei kann er auch die Rolle des Data Engineers übernehmen. Zudem stellt er die Ergebnisse seiner Arbeit dem Projektleiter vor und kann Teilprojektleiter-Tätigkeiten im Rahmen der Modellerstellung übernehmen (vgl. [Kozyrkov 2018] & [Nelson 2018]).

Data Engineer

Ein Data Engineer beschäftigt sich mit der Einbindung, Aufbereitung und dem zur Verfügung stellen von Datenquellen im Projekt. Er führt die Datenvorverarbeitung aus, damit im späteren Verlauf diese Daten in einen Machine-Learning-Workflow einfließen können. Diese Tätigkeiten können bei einer großen Anzahl von Datenquellen, mit einer hohen Unstrukturiertheit der Daten, in die verantwortliche Bearbeitung einer separaten Person ausgelagert oder, je nach Komplexität der Aufgaben, vom Data Scientist übernommen werden (vgl. [Tiedemann 2017], [Kozyrkov 2018] & [Nelson 2018]).

Entwickler (SW/HW)

Entwickler können sowohl für die Software als auch für die Hardwareentwicklung eingesetzt werden. Dies müssen nicht zwangsläufig, je nach Aufwand der zu bearbeitenden Tätigkeiten und der Profession des Entwicklers, unterschiedliche Personen sein. Entwickler sind für die Umsetzung und ggf. für das Testen des Systems nach den vorgegebenen Spezifikationen zuständig und können, je nach Vorgehensmodell der Entwicklung, in einem freien Rahmen oder strikt nach den Vorgaben des Projektleiters arbeiten.

Tester

Tester sind Personen, die Testfälle für das System entwickeln und diese ausführen, überprüfen und je nach Prozess als Qualitätsmanager die Auslieferung eines Produktes oder Prototyps aufgrund fehlgeschlagener Tests blockieren können. Diese Tester können die Entwickler des Systems selbst sein, was für kleine, nicht kritische Projektvorhaben möglich ist. Bei komplexen und kritischen Systemen, wie beim Einsatz einer autonomen

7 Validierung der Methode

Entscheidungsfindung, sollten separate Tester und Testteams für das Testen des Systems verantwortlich sein (vgl. [Sommerville 2011, 210f.]).

7 Validierung der Methode

Im Rahmen der Validierung wurde die Methode aus Kapitel 6 auf ein Szenario aus dem Forschungsprojekt UrbanLife+ angewandt. UrbanLife+¹ beschäftigt sich mit der Erhöhung der Mobilität von Senioren in Quartieren der Stadt Mönchengladbach, unter Einsatz von smarten städtebaulichen Objekten sowie intelligenter Unterstützung. Für die Validierung standen Fachexperten aus dem Pflegebereich, Ärzte sowie Softwareentwickler für die Bewertung zur Verfügung.

Das Szenario beschäftigte sich mit der Erhöhung der Mobilität innerhalb des Quartiers, durch eine individuelle und intelligente Wegführung, die die Senioren motivieren sollte, am Leben im Quartier wieder teilzuhaben. Gleichwohl erfüllte dieses Szenario mehrere therapeutische Aspekte, z. B. den einer der Erhöhung der Bewegungsbereitschaft sowie der Erhöhung des subjektiven Sicherheitsempfindens.

In den nachfolgenden Abschnitten wird das Szenario der Mobilitätsunterstützung beschrieben und gemäß der Methode durchgeführt sowie anschließend ausgewertet.

7.1 Szenario – Mobilitätsunterstützung für geriatrische Patienten

Das in diesem Abschnitt vorgestellte Szenario hat zum Ziel, die allgemeine Bewegungsbereitschaft geriatrischer Patienten im Quartier zu erhöhen. Dies soll die Senioren, auch im hohen Alter und bei stationärer Pflege, wieder mehr in das städtische Leben einbinden und für eine generell höhere Zufriedenheit der Patienten sorgen. Der vorangegangene Satz definiert zugleich schon die Zielstellung und Idee aus der Analysephase der Methode (vgl. Absatz 6.2.1.1).

¹ <https://www.urbanlifeplus.de/>

7 Validierung der Methode

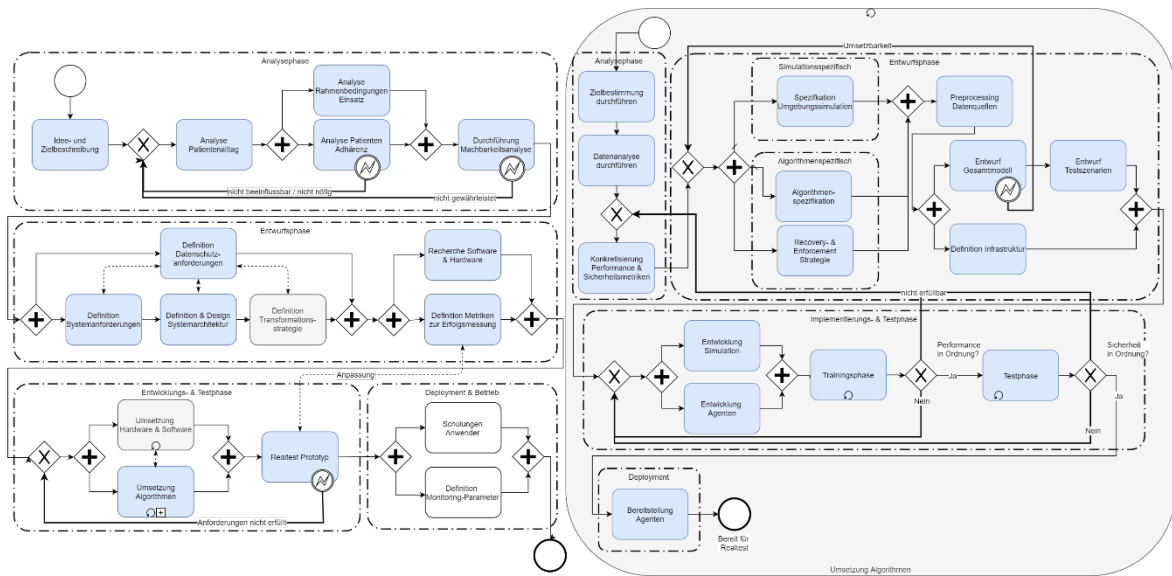


Abbildung 49 Gesamtvorgehensmodell Szenario Mobilitätsunterstützung

Aus der Abbildung 49 sind die relevanten und durchgeführten Aktivitäten des Szenarios ersichtlich. Da die Evaluation der Methode dem Vorgehen und ihrer Besonderheiten (insb. der sicheren Algorithmenentwicklung) dient, jedoch nicht der Erstellung eines finalen Produktes, welches auch im Rahmen eines Forschungsprojektes nicht angedacht ist, wurden die Schritte des Deployments, der Transformationsstrategie sowie der allgemeinen Softwareentwicklung nur theoretisch durchlaufen. Eine prototypische Umsetzung wurde jedoch für den gesamten Subprozess der Algorithmenentwicklung durchgeführt.

Grund für die Überlegung dieser Maßnahme war, dass Einschränkungen der Verhaltensmuster, durch aufgetretene Krankheiten und die daraus resultierenden Einschränkungen, zur Ablehnung von therapeutischen Maßnahmen führen können und somit eine schlechte Adhärenz und eine generell schlechtere Gesundheit, höhere stationäre Einweisungsraten und einer höhere Morbiditätsrate bedingen (vgl. [Weltgesundheitsorganisation 2003, 20f.] & [Pahor et al. 2014, 2388f.]). Durch Stärkung der Bewegungsmotivation kann diesen Resultaten entgegengewirkt werden.

7.1.1 Schritte Hauptprozess

Analyse Patientenalltag

Die Zielgruppe des Szenarios gleicht der Zielgruppe des Forschungsprojektes UrbanLife+. Hierbei handelt es sich um Senioren, welche das 65. Lebensjahr vollendet haben und stationär gepflegt werden. Die Zielgruppe hält sich vorrangig in der Pflegeeinrichtung auf und nutzt selten bis gar nicht die Möglichkeit, Ausflüge oder Spaziergänge außerhalb des stationären Geländes zu unternehmen. Ein Grund hierfür stellt die Multimorbidität der

7 Validierung der Methode

Zielgruppe dar. Diese beinhaltet oftmals physische Einschränkungen, welche sich in deutlichen Bewegungseinschränkungen niederschlägt. Gleichfalls lässt sich feststellen, dass die subjektive Sicherheit eine entscheidende Rolle für die Bewegung außerhalb von bekannten und gewohnten Umgebungen spielt (vgl. [Leukel et al. 2017, 14]).

Somit lässt sich schlussfolgern, dass die Reduzierung der Barrieren, sowohl physisch als auch mental, einen positiven Einfluss auf das Bewegungsverhalten der Zielgruppe haben kann.

Analyse Rahmenbedingungen

Die Rahmenbedingungen lassen sich gemäß der in Absatz 6.2.1.3 definierten Struktur gliedern. So ist bei der Überlegung zur Auswahl der Endgeräte für die Zielgruppe einzubeziehen, dass diese durch verschiedene Morbiditäten in ihren physischen, visuellen und akustischen Kompetenzen eingeschränkt sein können. Somit muss die Auswahl der technischen Ausstattung vor allem gut lesbare Endgeräte mit einer klaren, akustischen Ausgabe beinhalten. Ebenso sind leicht bedienbare Halterungen o. Ä. für die Geräte von Vorteil für die Zielgruppe.

Bei den organisatorischen Rahmenbedingungen ist zu betrachten, dass regelmäßige Medikamentenregime oder andere Pflegeprozesse, wie bspw. der Pflege von chronischen Wunden, die Entfernung und die Freizügigkeit der Bewegungsmotivation einschränken. Ebenso müssen kognitiv eingeschränkte Personen gesondert betrachtet und ggf. begleitet werden. Da allgemeine medizinische Daten verwendet werden (z. B. Art der Einschränkung), muss der Datenschutz entsprechend beachtet und angewandt werden.

Unter den ethischen Aspekten ist vor allem die Abgrenzung zu Medizinprodukten festzustellen. Gemäß der MDR, liegt im aktuellen Falle keine medizinische Indikation für ein Medizinprodukt der Klasse IIa oder höher vor, da weder therapeutische Entscheidungen noch kontinuierliche Messungen durchgeführt werden (vgl. [European Union 2017]). In diesem Falle entfällt die Betrachtung des Systems unter Begutachtung einer Ethikkommission.

Analyse Therapie & Adhärenz

Vorgeschlagen durch die WHO, lassen sich die relevanten Faktoren für chronische Erkrankungen untergliedern (siehe Unterabschnitt 3).

7 Validierung der Methode

Die Zielgruppe in diesem Vorhaben ist auf Personen beschränkt, die zwar physische, jedoch keine psychischen Einschränkungen, wie z. B. Demenz, haben. Hierdurch ergibt sich eine breite Streuung an Mobilitätseinschränkungen, welche sich unter die personenbezogenen Faktoren zusammenfassen lassen. Diesen Faktoren wird durch die Anwendung von unterstützender Technik, wie Rollatoren oder Rollstühlen, entgegengewirkt. Zusätzlich können noch Einschränkungen der Motorik, verlangsamte Bewegungsabläufe und Presbyopie sowie evtl. leichte Formen der Katarakt oder des Glaukom hinzukommen. Diese Ausprägungen, zusammen mit individuellen Medikamentenregimen, finden sich in der Therapie- und krankheitsbezogenen Gliederung der WHO wieder. Weitere Komorbiditäten können z. B. die Ausbildung von Depressionen, durch die krankheitsbezogenen Einschränkungen (vgl. [Bastos et al. 2015, 156f.]) sein, welche jedoch in der Auswahl der Zielgruppe nicht berücksichtigt wurden. In der Betrachtung der Arzt-bezogenen Faktoren finden sich keine besonderen Rahmenbedingungen, außer die stationäre Dauerbehandlung.

Machbarkeit analysieren

Damit im Rahmen des Forschungsprojektes der personelle, zeitliche und finanzielle Rahmen gewahrt bleibt, ist die Zielgruppe auf eine Gruppe mit gleichen Eigenschaften einzuschränken. Ebenso standen keine Ressourcen für begleitende Spaziergänge zur Verfügung, deshalb wurden nach Ermessen der Experten Realtests nur mit den Experten selbst durchgeführt. Zudem muss davon ausgegangen werden, dass nach Beendigung des Forschungsprojektes die zur Verfügung stehenden Ressourcen für den Betrieb der Lösung nicht mehr vorhanden sind, was die Einführung in einen Regelbetrieb schwierig macht. Somit steht vor allem der Erkenntnisgewinn im Vordergrund, der sowohl ressourcen- als auch risikoärmer ist. Eine detaillierte Aufstellung aller Stärken, Risiken, Schwächen und Möglichkeiten findet sich in Tabelle 15. Einer allgemeinen Umsetzung des Projektes widerspricht jedoch kein Punkt.

Tabelle 15 SWOT-Analyse für Szenario Mobilitätsunterstützung ohne Strategien

Stärken	Schwächen
<ul style="list-style-type: none">- Jahrelange Pflegeexpertise- Offen für technische Innovationen	<ul style="list-style-type: none">- Unternehmensgröße- Abhängigkeit von Gemeinde- Zielgruppe sind technische Laien & ggf. abweisend gegenüber Technik
Chancen	Risiken
<ul style="list-style-type: none">- Digitalisierung im Gesundheitsbereich erweitert sich	<ul style="list-style-type: none">- Externe Unternehmen sind schneller- Einsatz wird stärker reguliert

7 Validierung der Methode

- Fachkräftemangel wird sich nicht so schnell reduzieren	- Einstellung nach Beendigung Forschungsprojekt
- Zugriff auf Experten aus Forschungsprojekt	

Definition Datenschutzerfordernungen

Im Rahmen des Datenschutzes muss gewährleistet werden, dass die personenbezogenen Daten und darunter vor allem die medizinischen Informationen, gemäß der DSGVO und des BDSG gehandhabt werden. Auf Seiten des Pflegepartners steht ein Datenschutzbeauftragter zur Verfügung, der die Bewertung durchgeführt hat.

Die personenbezogenen Daten liegen in einem Rechenzentrum einer Universität in Deutschland und sind somit nach den Datenverarbeitungsgrundlagen DSGVO-konform. Zum Schutze der Personen wurden die entsprechenden medizinischen Daten verschlüsselt übertragen, so gering wie nötig gehalten und vor unautorisierten Zugriffen doppelt geschützt.

Aufgrund der geringen datenschutzrelevanten Datenmenge und deren eingeschränkter Auswertung, gab es keinen Bedarf für eine Datenschutzfolgeabschätzung.

Definition Systemanforderungen

Die Lösung für die Durchführung der gesetzten Ziele soll eine mobile Anwendung sein, die, auf Basis der aktuellen Position der Person, den für die Person individuell besten Weg ausgibt. Hierfür sind mobile Tablets vorgesehen, welche sowohl von der Größe, für die gute Sichtbarkeit als auch durch die technische Ausstattung, mit GPS und entsprechenden Betriebssystem, ideal zur Lösung beitragen. Eine Halterung für die Tablets auf ggf. vorhandenen Hilfsmitteln, soll eventuelle motorische Einschränkungen kompensieren. Die Anzeige der Route für eine einzelne Person der jeweiligen Zielgruppe erfolgt über eine vereinfachte Darstellung mit ausschließlich relevanten Informationen für die nächste notwendige Handlung.

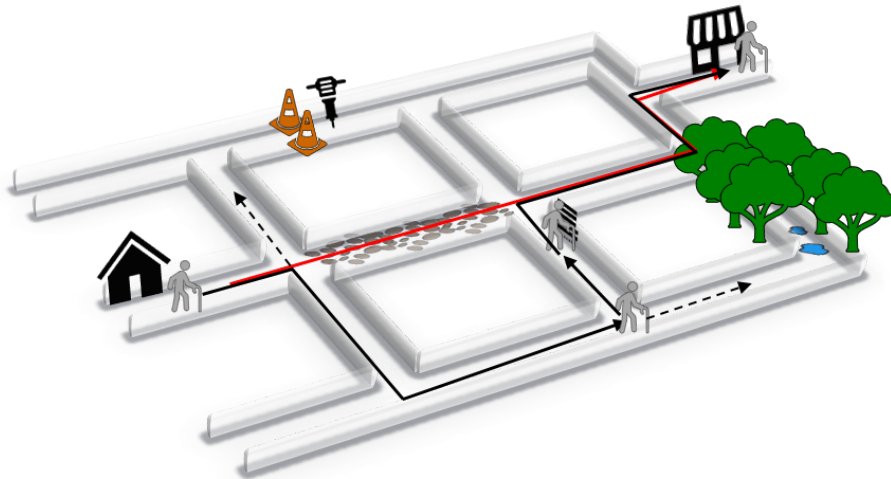


Abbildung 50 Schematische Darstellung Szenario Mobilitätsunterstützung

Abbildung 50 bildet schematisch die personalisierte Wegführung unter Berücksichtigung individueller Kompetenzen der Personen ab. Wegeeigenschaften und individuelle Anforderungen der Personen werden einbezogen, um eine Navigation vom Start zum Ziel zu finden, die zwar nicht der kürzesten aber der individuell passenderen Route entspricht.

Zur Ermöglichung der Lösung muss der entscheidungsunterstützende Algorithmus vorausschauend planen und bei ungeplanten Wegeänderungen eine Neuberechnung durchführen können. Aufgrund der geringen Bewegungsgeschwindigkeit der Zielgruppe ist eine Echtzeitreaktion des Systems nicht nötig. Ebenso ist eine permanente Internetverbindung möglich, da das System nur im urbanen Raum genutzt wird und eine dauerhafte Internetverbindung gewährleistet werden kann.

Die zu betrachtende Zielgruppe wurden in Rollator-Nutzer, Rollstuhlfahrer und Personen ohne Gehhilfe klassifiziert. Vorrangig vorliegende Straßeneigenschaften waren durch festen Untergrund (z. B. Asphalt), steinartig (z. B. Kopfsteinpflaster) oder waldartig (z. B. Wege mit Wurzeln, Sand oder Erde) gekennzeichnet.

Definition & Design Systemarchitektur

Als Systemarchitektur-Typ wurde auf Basis der Systemanforderungen die Client-Server-Architektur gewählt. Die im Backend vorhandenen Funktionalitäten wurden als Microservices umgesetzt und für eine entsprechende Skalierung bei multipler Nutzung in einem Kubernetes-Cluster gehostet. Gleiches gilt für das Hosting des Algorithmus, welcher, wie alle anderen Microservices, über eine REST-API dem Klienten zur Verfügung steht.

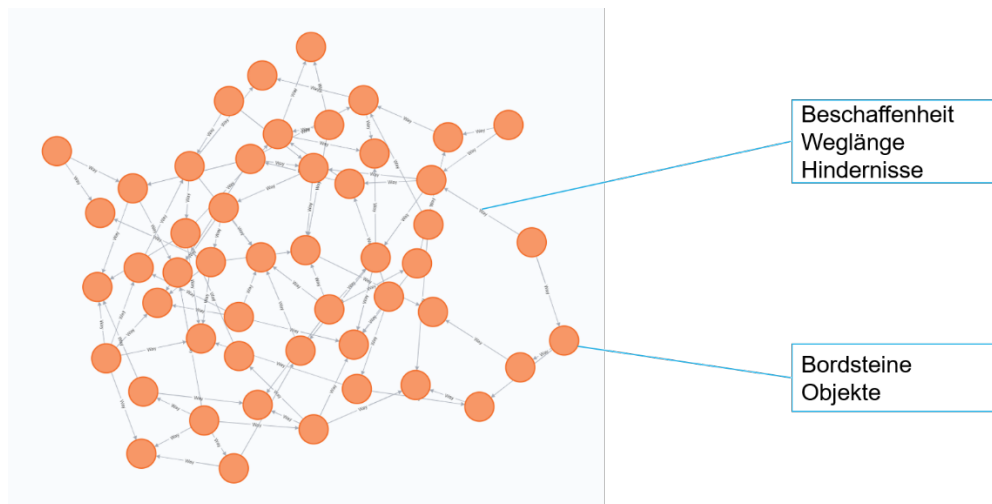


Abbildung 51 Speicherung der Wegeigenschaften als Graph

Die Informationen über die Wege und deren Attribute wurden in einer Graphdatenbank abgelegt (siehe Abbildung 51), welche eine einfache Traversierung beim Training des Algorithmus ermöglicht.

Definition Transformationsstrategie

Von einer möglichen Überführung des Systems in den Regelbetrieb wurde aus Gründen des im Vordergrund stehenden Forschungscharakters des Projektes und der fehlenden Ressourcen während und vor allem nach Projektbeendigung abgesehen. Somit war eine Integration in bestehende Systeme und Organisationsstrukturen nicht nötig.

Definition Metriken zur Erfolgsmessung

Im Rahmen des Projektes wurde als Zielmetrik die erfolgreiche Erstellung des Algorithmus mit entsprechenden Sicherheitstests ausgewählt sowie die Entscheidung der Pflegeexperten, ob der Algorithmus für die Zielgruppe als sicher zu betrachten ist oder nicht. Die zu definierenden Sicherheitsmetriken sind unter Einbezug von Domänenexperten mit Grenzen zu versehen, welche ebenfalls für die erfolgreiche Umsetzung eingehalten werden sollen (siehe Unterabschnitt 7.1.2).

Recherche & Auswahl HW / SW

In der Gestaltung der technischen Rahmenbedingungen für die gesonderten Anforderungen der Zielgruppe wurden Tablets als mobile Endgeräte ausgewählt. Vorrangig aufgrund der besseren Erkennbarkeit der Informationen auf den größeren Bildschirmen. Als Software für die Tablets dient Android als Betriebssystem, da dieses unabhängig vom Entwicklungs-

7 Validierung der Methode

Betriebssystem umgesetzt werden kann. Für die Entwicklung der Algorithmen wird auf das Tensorflow Framework zurückgegriffen.

Umsetzung HW / SW

Das Hauptaugenmerk der Evaluation lag auf der Algorithmenentwicklung und -sicherheit, weshalb bei der Evaluation auf die Durchführung einer App-Entwicklung für mobile Endgeräte verzichtet wurde. Im theoretischen Falle der Umsetzung dieser, wäre ein Scrum-basiertes agiles Entwicklungsvorgehen durchgeführt worden.

Durchführung Realtests

Die Realtests wurden zusammen mit den Experten aus der Pflege durchgeführt, indem der Algorithmus auf reale Daten im Umfeld des Quartiers simulativ angewandt wurde und die Ergebnisse von den Experten betrachtet und an die Entwicklung zurückgespielt wurden.

Deployment

Der Deployment-Part wurde in Folge der Konzentration auf die Entwicklung des Algorithmus, der nicht angedachten Einführung in den Regelbetrieb und der Vorgabe in Forschungsprojekten, keine Produktstudien durchzuführen, nicht durchgeführt.

7.1.2 Schritte Subprozess

Zielbestimmung durchführen

Der Algorithmus muss in der Lage sein, Ausgangslagen selbständig bewerten und Entscheidungen eigenständig treffen zu können. Als Eingangsdaten sind Wegeeigenschaften und Personeneigenschaften zu erfassen, als Ausgangsdaten muss eine Richtung für die nächste Kreuzung ausgegeben werden. Dabei muss auf eine variierende Anzahl an möglichen Entscheidungen pro Kreuzung geachtet werden. Auch hat jeder Weg individuelle Eigenschaften, die Berücksichtigung finden müssen.

Ziel des Algorithmus muss es sein, den kürzesten, aber optimalen Weg für die betreffende Gruppe der Zielgruppe zu bestimmen. Optimal ist der Weg dann, wenn er möglichst kurz ist aber trotzdem den gesonderten Merkmalen der Gruppe entsprechend Wege meidet, deren Eigenschaften nicht optimal für die Gruppe sind.

Technisch bietet sich für den Algorithmus der Einsatz aktueller Ansätze von präskriptiven Algorithmen an, welche während der Algorithmenspezifikation näher benannt werden.

Datenanalyse durchführen

Für die Durchführung des Trainings wurden Daten aus einer manuellen Erhebung im Quartier genutzt, welche im Projekt allen Partnern zur Verfügung gestellt wurden. Diese beinhalteten neben Straßennamen und geographischen Informationen Eigenschaften wie Wegbeschaffenheit und -längen. Der bereitgestellte Datensatz enthält alle relevanten Informationen für das Training des Algorithmus.

Konkretisierung Performance- & Sicherheitsmetriken

Für die Ermittlung der Performance-Kennzahlen wurden Performance-Metriken für aktuelle präskriptive Algorithmen herangezogen. Resultierend daraus wurde die Belohnungsmetrik (zu engl. auch Reward genannt) als hauptsächliche Performance-Kennzahl ausgewählt.

Für die Sicherheitsmetriken mussten zuerst die Hauptursachen für die Beeinträchtigung der Sicherheit der Wegführung identifiziert werden. Hierfür wurde ein Fishbone-Modell (vgl. [Kan 2008, 152ff.]) herangezogen.

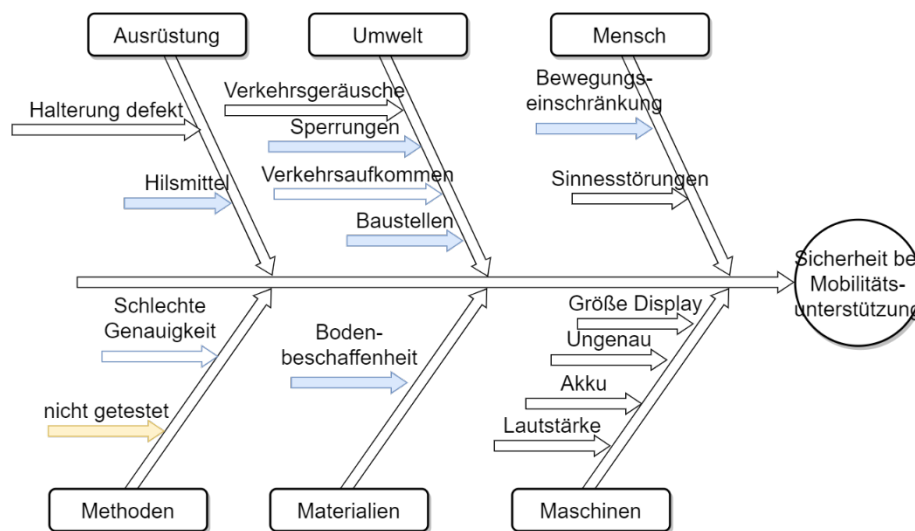


Abbildung 52 Fishbone für Szenario Mobilitätsunterstützung

Abbildung 52 zeigt das erstellte Fishbone-Modell. Dies wurde zusammen mit Experten aus dem Pflegebereich erstellt und identifiziert die hauptsächlichen Ursachen für die Reduzierung der Sicherheit bei der unterstützten Wegführung. Hieraus ergaben sich zur Abdeckung aller fünf mit blau gekennzeichneten Ursachen drei relevante Sicherheitsmetriken, welche in Tabelle 16 beschrieben sind. Die Ursache „nicht-getestet“ ist inhärent und durch die Erstellung von Testszenarien mit betrachtet.

7 Validierung der Methode

Tabelle 16 Sicherheitsmetriken für Szenario Mobilitätsunterstützung

Metrik	Beschreibung
Abweichung	Ist die Abweichung der Strecke in Metern vom theoretisch kürzesten Pfad.
	$ABW = \frac{\sum_{i=1}^n w_i^{sp} - w_i^I }{n} n \in \mathbb{N} \{0 < n \leq \infty\}; w^{sp}, w^I \in \mathbb{R}$
Erfolg	Beschreibt das erfolgreiche Erreichen des Ziels ohne Überschreitung von individuellen Grenzen (Wegtypen, Wegstrecke etc.)
	$ERF = \begin{cases} 1 & w_{length} < 200 \\ 0 & sonst \end{cases}$
Wegrate	Ist die Rate, wie häufig der potentiell schlechteste Wegtyp für die individuelle Zielgruppe gegenüber den anderen Wegtypen genutzt wurde.
	$WR = \frac{\sum_{i=1}^n w_i^{worst}}{n} n \in \mathbb{N} \{0 < n \leq \infty\}$

Die Grenzen für die besagten Metriken wurden zusammen mit den Pflegeexperten wie folgt definiert:

- Abweichung: 10% vom Normalwert
- Erfolg: 97%
- Wegrate: 15%

Datenselektion & Preprocessing

Für die Datenvorverarbeitung wurden die irrelevanten Informationen aus der Datenquelle entfernt, fehlende Daten markiert und ersetzt sowie die Datenbeschriftungen vereinheitlicht und ggf. kategorisiert. Danach wurden mithilfe eines Skripts die bereinigten Daten in die Graphdatenbank überführt, wobei Straßen = Kanten und Kreuzungen = Knoten, mit ihren entsprechenden Eigenschaften, ergeben.

Spezifikation Umgebungssimulation

Bei einer Recherche konnte keine bestehende Umgebungssimulation für die hier benötigten Rahmenbedingungen gefunden werden. Deshalb wurde eine eigenen Umgebungssimulation definiert und erstellt. Es wurde auf eine grafische Ausgabe verzichtet, da alle relevanten Informationen aus den Metriken gelesen werden können. Zur Realitätsannäherung wurden für die Traversierung die Daten aus der im letzten Schritt erstellten Graphdatenbank genutzt.

7 Validierung der Methode

Der State-Space besteht aus vier und weiteren fünf Informationen für den Agenten. Die ersten zwei Informationen sind die normalisierten IDs des aktuellen Knotens und des Zielknotens. Die dritte Information ist der aktuelle Zielgruppentyp (Rollator, Rollstuhlfahrer oder ohne Hilfsmittel). Die vierte Information ist der aktuell theoretisch kürzeste Weg in Metern. Die weiteren fünf Informationen sind die für den Agenten möglichen Wege, die er zur Auswahl seiner nächsten Aktion besitzt. Dabei enthält jede Weginformation sowohl die Weglänge in Metern als auch den Wegtyp.

Der Action-Space besteht lediglich aus einer Auswahl aus fünf Einzelwerten, wobei jeweils immer nur ein Wert wahr sein kann. Dies ist die Wahl für einen der maximal fünf möglichen Wege vom aktuellen Knoten aus.

Das Reward-Signal besteht aus drei Teilen. Jede Traversierung wird negativ bewertet, ebenso die Nutzung nicht-optimaler Wegtypen in Abhängigkeit vom aktuellen Zielgruppentyp. Zusätzlich wird das Erreichen oder nicht-Erreichen des Ziels (Sicherheitsmetrik Erfolg) entsprechend positiv oder negativ bewertet, wenn eine Bedingung davon zutrifft. Abbildung 53 zeigt die Definition des Reward Signals auf.

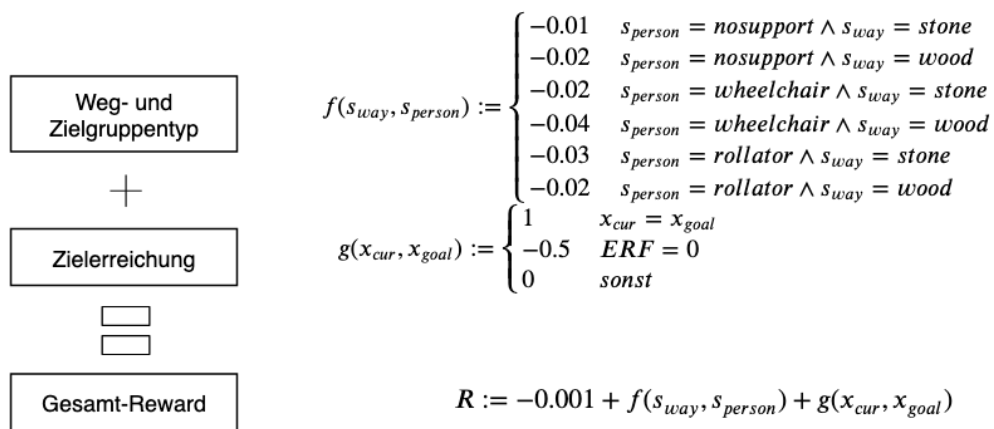


Abbildung 53 Reward-Definition für Szenario Mobilitätsunterstützung

Enforcement & Recovery Strategie

Da das Zielsystem nicht in den Regelbetrieb übergeben wurde, war die Definition entsprechender Prozesse für die Sicherstellung des Betriebs nicht nötig. Theoretisch denkbar wäre für die Enforcement-Strategie gewesen, dass Grenzwerte bereits beim Training auf Basis der Sicherheitsmetriken festgelegt werden, damit diese auch während der Ausführung gemessen werden können. Ebenso wäre beim Fallback-Handling denkbar, dass bei Ausfall der Internetverbindung oder beim Aussetzen des Agenten, der kürzeste Pfad berechnet wird,

7 Validierung der Methode

statt der optimale. Eine spezielle Betrachtung der Reversibilität ist nicht nötig, da keine Entscheidung des Agenten einen irreversiblen Zustand erzeugen würde.

Algorithmenspezifikation

Aufgrund großer Durchbrüche in den letzten Jahren im Bereich des Reinforcement Learnings für präskriptive Anwendungen (vgl. Unterabschnitt 2.2.8 & [Vinyals et al. 2019, 350ff.]), durch den verbundenen Einsatz von Deep Learning Methoden, wurde diese Technik ebenfalls für Traversierungs- und Routing-Anwendungen genutzt (vgl. [Mammeri 2019, 55916ff.]). Somit fiel, auch aufgrund der möglichen Flexibilität des Reinforcement Learnings, die Wahl für den Anwendungsfall auf diese Technik.

Als Algorithmus wurde der Proximal Policy Optimization Algorithmus verwendet (vgl. [Schulman et al. 2017, 2ff.]), da dieser gerade bei stetigen State Spaces ein gutes exploratives Verhalten und mit einer besseren Konvergenz (vor allem durch die Anwendung der Kullback-Leibler-Divergenz bei der Policy-Aktualisierung) als bspw. DQN-Algorithmen überzeugt. Er lernt somit stabiler.

Auf den Einsatz von model-based RL wurde verzichtet, da bereits auf der Basis von realen Daten als Abbild der Realität gelernt wird und nicht zwingend eine zusätzliche Modellbildung erforderlich ist.

Entwurf Gesamtmodell

Im Rahmen des Gesamtmodells wurden unter Berücksichtigung der vorangegangenen Schritte die Agentenstruktur und das Agentensystem beschrieben.

Da das Routing unabhängig von anderen sich auf der Route befindenden Personen funktionieren soll, ist eine Definition der Interaktion mit anderen Agenten in einem Multi-Agenten-Einsatz nicht notwendig.

7 Validierung der Methode

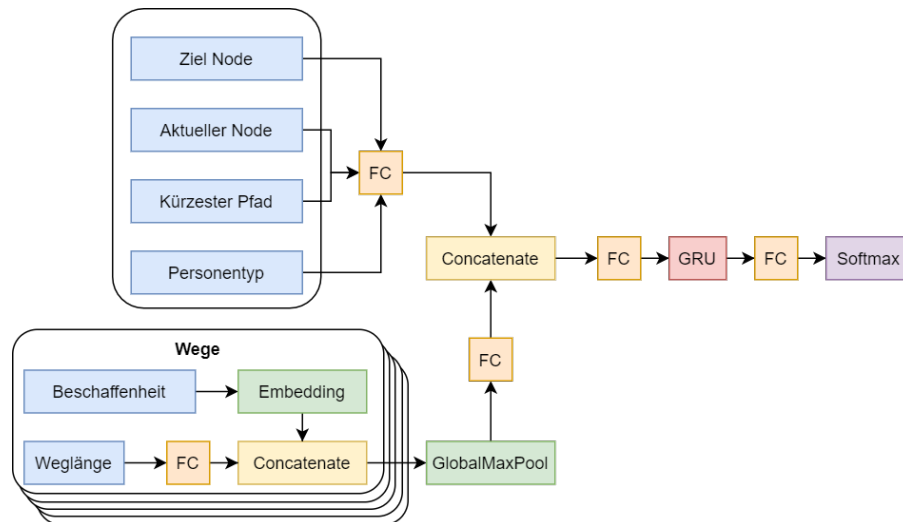


Abbildung 54 Agentennetzwerk für Szenario Mobilitätsunterstützung

Die Struktur des Netzwerks für den Agenten (Abbildung 54) verbindet den State Space Eingang (blau) der Simulationsumgebung mit dem Action Space Ausgang (violett) in die Simulationsumgebung (siehe Anhang 4). Die einzelnen möglichen Wege, die durch den Agenten vom aktuellen Knoten aus begangen werden können, werden über eine Aggregationsfunktion (Global-Maximum-Pooling) zusammengefasst. Dies hat den Vorteil, dass nur Informationen über den besten nächsten Weg weitergegeben werden. Die Embedding-Schicht bietet die Möglichkeit, höherdimensionale Beschaffenheiten und Eigenschaften für die Wege zu definieren. In den GRU-Zellen (vgl. [Chung et al. 2014, 3ff.]) findet die Sequenzverarbeitung und das „Merken“ des Agenten statt.

Definition Infrastruktur-Anforderungen

Für das Training des Algorithmus und die Ausführung der Simulation wurde eine Server Infrastruktur genutzt, die auf dem Master-Slave Ansatz (vgl. [Kong et al. 2017, 1f.] & [Liang et al. 2018, 1ff.]) aufsetzte. Der Master koordinierte die Trainingsrunden und die Slaves führten die Berechnungen der Simulationen und Agenten durch. Dadurch konnten mehrere Agenten und Simulationen nebeneinander durchgeführt werden, wodurch ein schnelleres Anlernen ermöglicht wurde.

Entwurf Testplanung

Die in den Sicherheitsmetriken definierten Ursachen von Problemen wurden in Testfälle überführt. Hierfür wurden die Daten der Graphdatenbank dupliziert und entsprechend für die Testfälle modifiziert. Konkret wurden Testfälle für veränderte Attribute der Wegführung erstellt ebenso wie für das Fehlen von Wegen.

7 Validierung der Methode

Tabelle 17 Testfälle für Szenario Mobilitätsunterstützung

Testfall	Veränderung
Art des Weges	10% der Wegtypen wurden zufällig geändert (Simulation von Baustellen oder Verschmutzungen)
Fehlen von Wegen	5% der Wege wurden zufällig entfernt (Simulation von Sperrungen)

Diese Testfälle wurden sequenziell ausgeführt und mit den Standardwerten verglichen.

Entwicklung Simulation & Agenten

Die Simulation der Umgebung wurde in der Programmiersprache Python umgesetzt und bindet die Graphdatenbank an die relevanten Trainingsdaten an. Der Agent wurde ebenfalls mit Python umgesetzt.

Trainings- & Testphase

Für das Training des Agenten wurde das RLlib Framework genutzt (vgl. [Liang et al. 2018, 1ff.]), da hierin bereits getestete Algorithmen implementiert waren und somit Implementierungsfehler in diesem Rahmen ausgeschlossen werden konnten. Gleichwohl konnte durch die Integration mit dem Tune-Framework ein Parameter-Tuning vorgenommen werden (vgl. [Liaw et al. 2018, 1ff.]), was von einer manuellen Hyperparameter-Suche entlastete (siehe Parameter im Anhang 3). Beide Frameworks werden von der Berkeley Artificial Intelligence Research Gruppe (kurz BAIR-Gruppe) weiterentwickelt und betreut, was für die Qualität der Frameworks spricht.

Für das Training wurden 150 Training-Iterationen pro Konfiguration durchgeführt, was 90.000 Schritte pro Konfiguration entsprach. Die Konfiguration mit dem besten mittleren Reward und somit mit der besten Performance, wurde für die Weiterverwendung genutzt. Nicht vielversprechende Konfigurationen wurden vom eingesetzten Hyperparameter-Optimierungs-Algorithmus (vgl. [Li et al. 2018, 6ff.]) frühzeitig gestoppt (siehe Abbildung 55).

7 Validierung der Methode

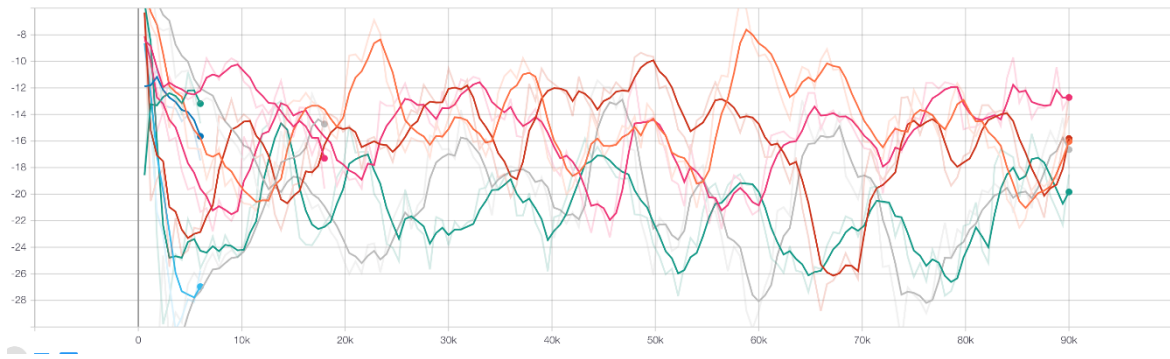


Abbildung 55 Trainingsverlauf mit Hyperparameteroptimierung (x-Achse: Schritte, y-Achse: Reward); siehe Anhang 6

Die als optimal gekennzeichnete Konfiguration wurde daraufhin 200 Mal in zufallsbasierten Start-End-Routing-Szenarien ausgeführt, um die Normalwerte der definierten Sicherheitsmetriken, also die s_0^x -Werte gemäß Absatz 6.3.1.3, zu erhalten. Daraufhin wurden die definierten Testszenarien mit gleicher Konfiguration aber unterschiedlicher Datenbasis ausgeführt, um die Sicherheitskennzahlen unter Sonderfällen zu messen.

Problem	Ursachen	Sicherheitsmetriken	Default	Test Sperrung	Test Wegtyp	Grenze	Schaden	Eintritt	RPZ Ist
Sicherheit Mobilität	Sperrungen	Abweichung	49	53	56	53,9	1,3	0,4	1,3
Sicherheit Mobilität	Baustellen	Abweichung	49	53	56	53,9	1,3	0,4	1,3
Sicherheit Mobilität	Hilfsmittel	Erfolg	97%	95,5%	96%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bewegungseinschränkung	Erfolg	97%	95,5%	96%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bodenbeschaffenheit	Wegrate	13%	19,6%	18,5%	15,0%	3,4	2,3	7,9

Abbildung 56 FMEA für Szenario Mobilitätsunterstützung

Es ergab sich für die Testfälle eine durchweg geringe Abweichung vom Normalfall (siehe Abbildung 56). Lediglich bei der Wegrate lies sich eine Verschlechterung in beiden Testfällen nachweisen, wobei die Bodenbeschaffenheit mit einer RPZ von 7,9 von 10 durchaus hoch ist und gesenkt werden sollte.

Bei einer kurzen Verhaltensanalyse des Agenten (vgl. [Zahavy et al. 2017, 5ff.]) war zu erkennen, dass sich die Situation vor allem auf zwei Gruppen nahe beieinanderliegender Ziele im Graphen bezog, welche fast nur über nicht-optimale Wege erreichbar waren (siehe Abbildung 57). Jeder Punkt in dem t-sne-Graphen entspricht einem Status während der Ausführung und ist nach optimalen und nicht-optimalen Wegen eingefärbt.

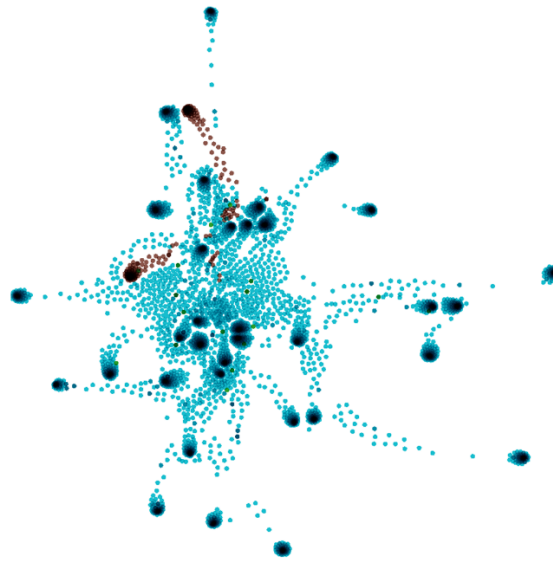


Abbildung 57 t-SNE für das Wegrate-Verhalten (nicht-optimal sind die braunen Ansammlungen)

Zur Korrektur dieses Verhaltens und zur Anpassung des Algorithmus wurde ein Action Masking für die Ausgabe aus der Softmax-Schicht eingeführt. Dies hatte den Vorteil, dass beim Trainieren des Algorithmus eine variable Anzahl an Ausgabewerten besser unterstützt wurde als in der aktuellen Fassung mit dem „Nullen“ aller nicht vorhandenen Wege (siehe Anhang 5).

Problem	Ursachen	Sicherheitsmetriken	Default	Test Sperrung	Test Wegtyp	Grenze	Schaden	Eintritt	RPZ
Sicherheit Mobilität	Sperrungen	Abweichung	49	53	56	53,9	1,3	0,4	1,3
Sicherheit Mobilität	Baustellen	Abweichung	49	53	56	53,9	1,3	0,4	1,3
Sicherheit Mobilität	Hilfsmittel	Erfolg	97%	95,5%	96%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bewegungseinschränkung	Erfolg	97%	95,5%	96%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bodenbeschaffenheit	Wegrate	13%	19,6%	18,5%	15,0%	3,4	2,3	7,9
Nach Anpassung									
Sicherheit Mobilität	Sperrungen	Abweichung	49	52	58	53,9	1,6	0,7	1,6
Sicherheit Mobilität	Baustellen	Abweichung	49	52	58	53,9	1,6	0,7	1,6
Sicherheit Mobilität	Hilfsmittel	Erfolg	97%	97,2%	95,5%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bewegungseinschränkung	Erfolg	97%	97,2%	95,5%	96%	0,2	0,1	1,0
Sicherheit Mobilität	Bodenbeschaffenheit	Wegrate	13%	16,2%	17,0%	15,0%	2,4	1,2	2,8

Abbildung 58 Soll-FMEA in Szenario Mobilitätsunterstützung

Final konnte somit die RPZ (siehe Abbildung 58), für den Sonderfall der Bodenbeschaffenheit nach einem erneuten Training des Algorithmus mit der optimalen Konfiguration, reduziert werden. Insgesamt wurde für die Verbesserung des Algorithmus der komplette Zyklus, wie in Abbildung 36 Aggregierter Sicherheitsprozess dargestellt, durchlaufen.

Bereitstellung

Der Algorithmus wurde als Modell über eine REST-API in dem genutzten Kubernetes-Cluster manuell bereitgestellt.

7.1.3 Zusammenfassung

Anhand eines Beispiels konnte die Funktionalität der vorgestellten Methode aufgezeigt werden. Des Weiteren wurde der gesamte Prozess, inklusive der Ergebnisse aus der Testphase, eng von Experten aus dem Pflegebereich begleitet und als eine sicherere Einsatzform von präskriptiven Algorithmen in der Therapieunterstützung eingeschätzt. Ebenso wurden Teilergebnisse auf praxisrelevanten Entwicklerkonferenzen (vgl. [Skowron 2019b]) und wissenschaftlichen Konferenzen (vgl. [Skowron 2019a, 6ff.]) vorgestellt und im Rahmen dieser Vorstellung als positives und nützliches Vorgehen hervorgehoben. Das Thema Adhärenz und Therapie zur Bewegungsförderung wurde im Rahmen der Anforderungsaufnahme mit einbezogen und bildet sich in den Parametern für Strecken und Nutzergruppen ab. Dies unterstützt die Zielgruppe bei selbständigen Spaziergängen oder Beschaffungen, da die Sicherheit bei den gegangenen Wegen erhöht werden kann. Bei zunehmendem Einbezug von weiteren Morbiditäten (z. B. Sehstörungen), wären die Simulationsumgebung sowie der Algorithmus entsprechend zu erweitern, um auch hier einen Einbezug von Therapie und Therapieunterstützung zu gewährleisten. Hierbei würde sich der Umfang der Simulation, der Vorbetrachtungen und der Metriken erweitern. Das Vorgehen hingegen würde gleich ablaufen, wodurch eine Skalierbarkeit für sowohl große als auch kleine Vorhaben möglich ist.

8 Zusammenfassung & Ausblick

Im Zuge dieser Arbeit wurde eine Methode zur Entwicklung sicherer präskriptiver Therapiesysteme designt und beschrieben. Die Motivation für das Vorhaben dieser Arbeit wurde in der Einleitung mit der Analyse der aktuellen Entwicklungen auf dem Forschungsfeld von entscheidungsunterstützenden mHealth Lösungen im Gesundheitswesen und den Auswirkungen des bestehenden Mangels an Fachkräften in diesem Bereich hinreichend beschrieben. Nach einer kurzen Definition der Problemstellung wurden mit Expertenbefragungen und einer systematischen Literaturanalyse die Anforderungen für die Entwicklung der oben benannten Systeme entwickelt. Folgend wurden bestehende Ansätze und Methoden identifiziert und mit den gegebenen Anforderungen abgeglichen. Hieraus ergab sich, dass die aktuell existierenden Ansätze und Methoden als Vorgehensmodelle die spezielle Zusammenstellung der Anforderungen für entscheidungsunterstützende Therapiesysteme nicht oder nur teilweise erfüllen. Daraufhin wurde mithilfe des Method Engineerings eine Methode mit all ihren relevanten Teilartefakten erstellt, welche den identifizierten Anforderungen vollumfänglich entspricht.

8 Zusammenfassung & Ausblick

Im Anschluss wurde die Methode im Rahmen eines Anwendungsfalls in einem Forschungsprojekt angewandt und vollumfänglich durchlaufen. Hierbei zeigte sich, dass die Schwerpunkte der Sicherheit der eingesetzten präskriptiven Algorithmen sowie die Einbindung individueller therapeutischer Rahmenbedingungen, durchgängig in die klassische Entwicklung integriert werden konnten. Damit kann sichergestellt werden, dass bei der Anforderungsdefinition und Umsetzung solcher entscheidungsunterstützender Therapieunterstützungssysteme einerseits immer auf die individuellen Rahmenbedingungen der Patienten eingegangen wird und, andererseits eine explizite Betrachtung der Sicherheit der angewandten Algorithmen, sowohl während der Entwicklung als auch im produktiven Einsatz, stattfindet. Mithilfe einer Erweiterung bestehender Ansätze der Prozessoptimierung, kann die Zielsetzung und die Messung der Sicherheit in solchen Systemen in bereits etablierte Managementprozesse eingebunden werden, ohne dabei grundlegende Mechaniken der angewandten Techniken verändern zu müssen. Weiterhin verhelfen eine explizite Einbindung der Therapietreue und der patientenindividuellen Rahmenbedingungen dagegen eher zu einer therapeutisch-patientenorientierten, anstatt zu einer schwerpunktmäßig technikgetriebenen Entwicklung. Damit wird sichergestellt, dass die geplanten Systeme medizinisch betrachtet zielgruppenspezifisch ausgerichtet sind und somit im praktischen Umfeld einen positiven Effekt bei der Unterstützung der Gesundheitsversorgung erzeugen. Die nachhaltige Einbettung relevanter medizinischer Fachkräfte in den Prozess der Erarbeitung dieser Anforderungen sichert die Entwicklung der genannten Systeme zusätzlich ab, wodurch zukünftige Entwicklungen präskriptiver Therapiesysteme verstärkt die Sicherheit und den Nutzen, nicht nur für den Anwender, sondern auch für den Patienten, in den Vordergrund stellen.

Literaturverzeichnis

- Abts, D., Mülder, W., Grundkurs Wirtschaftsinformatik, 6., überarb. und erw. Aufl. Aufl., Vieweg + Teubner, Wiesbaden, 2009.
- Adherence to long-term therapies, Geneva, 2003.
- Agile Alliance (Hrsg.), What is Agile?, 2019, URL: <https://www.agilealliance.org/agile101/>, gelesen am 07.05.2019.
- Aleithe, M., Skowron, P., Carell, A., Boettger, D., Goblirsch, T., Franczyk, B., Simulation Framework for Mobile Patient Monitoring Systems, Proceedings of the International Workshop on Innovative Simulation for Health Care, 2018, S. 1–7.
- Alpar, P., Alt, R., Bensberg, F., Grob, H.L., Weimann, P., Winter, R., Anwendungsorientierte Wirtschaftsinformatik, 8. Auflage. Aufl., Springer Vieweg, Wiesbaden, 2016.
- Alt, R., Österle, H., Real-time Business, Springer, Berlin, 2004.
- Ambler, S.W., The elements of UML 2.0 style, Cambridge University Press, Cambridge, 2005.
- Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D., Concrete Problems in AI Safety, in: ArXiv (2016) abs/1606.06565.
- Angermeier, G., Make-or-Buy-Entscheidung, 2009, URL: <https://www.projektmagazin.de/glossarterm/make-or-buy-entscheidung>, gelesen am 22.02.2019.
- Antwerpes, F., Merz, S., Therapie, 2017, URL: <https://flexikon.doccheck.com/de/Therapie>, gelesen am 09.05.2019.
- Aronson, E., Wilson, T.D., Akert, R.M., Sozialpsychologie, 6., aktualisierte Aufl., [Nachdr.] Aufl., Pearson Studium, München, 2011.
- Ashlock, D., Evolutionary Computation for Modeling and Optimization, Springer Science+Business Media Inc, New York, NY, 2006.
- Auer, P., Cesa-Bianchi, N., Fischer, P., Finite-time Analysis of the Multiarmed Bandit Problem, in: Machine Learning, 47 (2002) 2/3, S. 235–256.
- Bastos, A.M., Faria, C.G., Moreira, E., Morais, D., Melo-de-Carvalho, J.M., Paul, M.C., The importance of neighborhood ecological assets in community dwelling old people aging outcomes: A study in Northern Portugal, in: Frontiers in aging neuroscience, 7 (2015), S. 156.
- Bauftragter der Bundesregierung für die Informationstechnik, V-Modell XT Bund, 2009.

Literaturverzeichnis

- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., Manifest für Agile Softwareentwicklung, 2001, URL: <https://agilemanifesto.org/iso/de/manifesto.html>, gelesen am 07.05.2019.
- Becker, J.H., Ebert, H., Pastoors, S., Praxishandbuch berufliche Schlüsselkompetenzen, Springer, Berlin, Heidelberg, 2018.
- Becker, T., Herrmann, R., Sandor, V., Schäfer, D., Wellisch, U., Deskriptive Statistik und explorative Datenanalyse, in: Becker, T., Herrmann, R., Sandor, V., Schäfer, D., Wellisch, U. (Hrsg.), Stochastische Risikomodellierung und statistische Methoden, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, S. 27–91.
- Beims, M., Ziegenbein, M., IT-Service-Management in der Praxis mit ITIL, 4., überarbeitete und erweiterte Auflage. Aufl., Hanser, München, 2015.
- Beuth, P., Twitter-Nutzer machen Chatbot zur Rassistin, 2016, URL: <https://www.zeit.de/digital/internet/2016-03/microsoft-tay-chatbot-twitter-rassistisch>, gelesen am 17.06.2019.
- Bibliographisches Institut GmbH (Hg.), The-ra-pie, die, 2019, URL: <https://www.duden.de/rechtschreibung/Therapie>, gelesen am 09.05.2019.
- Bitkom e. V., Was muss ich wissen zur EU-Datenschutz Grundverordnung?, 2016.
- Bitkom e. V., The Digital Health Sector, Bexbach, 2018.
- Bourque, P., Fairley, R.E., Guide to the software engineering body of knowledge, Version 3.0. Aufl., IEEE Computer Society, [Los Alamitos, Calif.], 2014.
- Brown, M.T., Bussell, J.K., Medication adherence: WHO cares?, in: Mayo Clinic proceedings, 86 (2011) 4, S. 304–314.
- Brownlee, J., Statistical Methods for Machine Learning, selbstverlegt, 2018.
- Brundage et al., The Malicious Use of Artificial Intelligence, 2018.
- Bundesministerium der Justiz und für Verbraucherschutz, Gesetz über die elektromagnetische Verträglichkeit von Betriebsmitteln, 2016, URL: https://www.gesetze-im-internet.de/emvg_2016/BJNR287910016.html, gelesen am 22.02.2019.
- Bundesministerium der Justiz und für Verbraucherschutz, Gesetz über Medizinprodukte, 2017.
- Bundesministerium der Justiz und für Verbraucherschutz, Bundesdatenschutzgesetz, 2018, gelesen am 15.02.2019.

- Burda et al., Large-Scale Study of Curiosity-Driven Learning, 2018.
- Burghardt, M.H., Projektmanagement, 9., wesentlich überarb. und erw. Aufl. Aufl., Publicis Publ, Erlangen, 2012.
- Bynum, T., Computer and Information Ethics, in: Edward N. Zalta (Hrsg.), The Stanford Encyclopedia of Philosophy, Summer 2018. Aufl., Metaphysics Research Lab, Stanford University, 2018.
- Cesa-Bianchi et al., Boltzmann Exploration Done Right, 2017.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., CRISP-DM 1.0, 2000, URL: <https://www.the-modeling-agency.com/crisp-dm.pdf>, gelesen am 30.04.2019.
- Chollet, F., Deep learning with Python, Manning, Shelter Island, NY, 2018.
- Chung et al., Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014.
- DAK, Umfrage zur Überforderung durch eine Pflegetätigkeit in Deutschland nach Altersgruppen im Jahr 2015, Statista, 2015.
- Data science & big data analytics, John Wiley & Sons, Indianapolis, IN, 2015.
- Datenethikkommission der Bundesregierung (Hrsg.), Gutachten der Datenethikkommission der Bundesregierung, Berlin, 2019.
- Deshpande, A., Kumar, M., Artificial Intelligence for big data, Packt Publishing, Birmingham, UK, 2018.
- Digitale Wörterbuch der deutschen Sprache (Hg.), Therapie, die, 2019, URL: <https://www.dwds.de/wb/Therapie>, gelesen am 09.05.2019.
- Doran, G.T., There's a S.M.A.R.T. way to write managements's goals and objectives, in: Management Review, 70 (1981) 11, S. 35.
- Dulac-Arnold et al., Deep Reinforcement Learning in Large Discrete Action Spaces, 2015.
- Dziak, D., Jachimczyk, B., Kulesza, W., IoT-Based Information System for Healthcare Application: Design Methodology Approach, in: Applied Sciences, 7 (2017) 6, S. 596.
- Eggert et al., Schülerbefragung Pflege: Eigene Erfahrungen und Interess an Pflegeberufen, 2019.
- Elmer, A., Matusiewicz, D., Althammer, T., Die digitale Transformation der Pflege, Erste Auflage. Aufl., MWV Medizinisch Wissenschaftliche Verlagsgesellschaft, Berlin, 2019.

- European Union, Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG, 2016, URL: <https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32016R0679&from=EN>, gelesen am 15.02.2019.
- European Union, (EU) 2017/746 Medical Device Regulation, 2017, URL: <https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32017R0745&from=EN>, gelesen am 24.10.2018.
- Evans, J.R., Lindner, C.H., Business analytics, in: Decision Line, 43 (2012) 2, S. 4–6.
- Feldmann, C., Gorj, A., Forschungsmethodik, in: Feldmann, C., Gorj, A. (Hrsg.), 3D-Druck und Lean Production, Springer Fachmedien Wiesbaden, Wiesbaden, 2017, S. 11–15.
- Felkai, R., Beiderwieden, A., Projektmanagement für technische Projekte, 3., überarbeitete und erweiterte Auflage. Aufl., Springer Vieweg, Wiesbaden, 2015.
- Fittkau, T., Ruf, W., Ganzheitliches IT-Projektmanagement, De Gruyter, München, 2008.
- Foglyano, K.M., Schnellenberger, J.R., Kobetic, R., Lombardo, L., Pinault, G., Selkirk, S., Makowski, N.S., Triolo, R.J., Accelerometer-based step initiation control for gait-assist neuroprostheses, in: Journal of rehabilitation research and development, 53 (2016) 6, S. 919–932.
- Forkan, A.R.M., Khalil, I., A clinical decision-making mechanism for context-aware and patient-specific remote monitoring systems using the correlations of multiple vital signs, in: Computer methods and programs in biomedicine, 139 (2017), S. 1–16.
- Fortino, G., Russo, W., ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems, in: Information and Software Technology, 54 (2012) 6, S. 608–624.
- François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J., An Introduction to Deep Reinforcement Learning, in: Foundations and Trends® in Machine Learning, 11 (2018) 3-4, S. 219–354.
- Fricklas, K., Shukla, N., Machine learning with TensorFlow, Manning, Shelter Island, NY, 2018.
- Friedrichs, J., Methoden empirischer Sozialforschung, 14. Auflage. Aufl., VS Verlag für Sozialwissenschaften, Wiesbaden, s.l., 1990.

- Gadatsch, A., Grundkurs Geschäftsprozess-Management, 6., aktualisierte Auflage. Aufl., Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2010.
- Gadatsch, A., Datenmodellierung für Einsteiger, Springer Fachmedien Wiesbaden, Wiesbaden, 2017.
- Gauci et al., Horizon: Facebook's Open Source Applied Reinforcement Learning Platform, 2018.
- Gloger, B., Scrum, 5., überarbeitete Auflage. Aufl., Hanser, München, 2016.
- Gluchowski, P., Dittmar, C., Gabriel, R., Management Support Systeme und Business Intelligence, 2, vollst. überarb. Aufl. Aufl., Springer, Berlin, Heidelberg, 2008.
- Goltsche, W., COBIT kompakt und verständlich, 1. Aufl. Aufl., Vieweg, s.l., 2006.
- Goodfellow, I., Bengio, Y., Courville, A., Deep learning, MIT Press, Cambridge, Massachusetts, London, England, 2016.
- Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., Celi, L.A., Guidelines for reinforcement learning in healthcare, in: Nature medicine, 25 (2019) 1, S. 16–18.
- Gowal et al., On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models, 2018.
- Gröger, C., Schwarz, H., Mitschang, B., Prescriptive Analytics for Recommendation-Based Business Process Optimization, in: van der Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C., Abramowicz, W., Kokkinaki, A. (Hrsg.), Business Information Systems, Springer International Publishing, Cham, 2014, S. 25–37.
- Grønhaug, G., Addressing the elephant in the room: a possible new way to increase patient adherence to medical advice, in: Patient preference and adherence, 11 (2017), S. 1083–1089.
- Gudehus, T., Logistik, 4., aktualisierte Aufl 2010. Aufl., Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.
- Gudimella et al., Deep Reinforcement Learning for Dexterous Manipulation with Concept Networks, 2017.
- Gutzwiller, T.A., Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica-Verlag HD, Heidelberg, 1994.
- Haarnoja et al., Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, 2018.

- Hagerty, J., 2017 Planning Guide for Data and Analytics, 2016, URL: https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/2017_planning_guide_for_data_analytics.pdf, gelesen am 20.08.2018.
- Handelsblatt, McKinsey, MarketsandMarkets, Bayer, Tufts Center for the Study of Drug Development, Umsatz durch künstliche Intelligenz im Gesundheitswesen weltweit in den Jahren 2018 und 2025, in: Handelsblatt (2019) 23, S. 48.
- Hars, A., Referenzdatenmodelle, Gabler Verlag, Wiesbaden, 1994.
- Hassan, Q.F., Khan, A.u.R., Madani, S.A., Internet of things, Taylor & Francis, CRC Press, Boca Raton, 2018.
- Hehner et al., Digitalisierung im Gesundheitswesen: die Chancen für Deutschland, 2018.
- Henderson et al., Deep Reinforcement Learning that Matters, 2017.
- Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M., Situational method engineering, Springer, Berlin, 2014.
- Herold, G., Innere Medizin 2013, Selbstverl., Köln, 2013.
- Hockenbrink, T., Nachhaltigkeit und Balanced Scorecard, Diplomica Verl., Hamburg, 2012.
- Initiative Neue Qualität der Arbeit (BAuA), Digitalisierung in der Pflege, 2018.
- Kan, S.H., Metrics and models in software quality engineering, 2. ed., 8. print. Aufl., Addison-Wesley, Boston, 2008.
- Karlstedt, F., Qualitätskennzahlen im Projektmanagement, Igel Verl. RWS, Hamburg, 2014.
- Kaspersky Inc. (Hrsg.), Was ist Spear-Phishing?, 2019, URL: <https://www.kaspersky.de/resource-center/definitions/spear-phishing>, gelesen am 17.06.2019.
- Kaune, A., Change Management mit Organisationsentwicklung, 2., neu bearbeitete Auflage. Aufl., Schmidt, Berlin, 2010.
- Kim, G., Humble, J., Debois, P., The DevOps handbook, First edition. Aufl., 2016.
- Koch, S., Einführung in das Management von Geschäftsprozessen, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.
- Kong et al., Revisiting the Master-Slave Architecture in Multi-Agent Deep Reinforcement Learning, 2017.
- Kotler, P., Keller, K.L., Marketing management, Fourteenth edition. Aufl., Prentice Hall/Pearson, Boston, Mass., 2012.
- Kozyrkov, C., Top 10 roles in AI and data science, 2018, URL: <https://hackernoon.com/top-10-roles-for-your-data-science-team-e7f05d90d961>, gelesen am 16.04.2019.

- Lange, C., Daten und Fakten: Ergebnisse der Studie "Gesundheit in Deutschland aktuell 2009", Robert-Koch-Institut, Berlin, 2011.
- Leike et al., AI Safety Gridworlds, 2017.
- Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., Legg, S., Scalable agent alignment via reward modeling: a research direction, in: ArXiv (2018) abs/1811.07871.
- Leukel, J., Schehl, B., Wallrafen, S., Hubl, M., Impact of IT Use by Older Adults on Their Outdoor Activities, Proceedings of the 38th International Conference on Information Systems, 38. Aufl., 2017, S. 12–24.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, in: Journal of Machine Learning Research, 2018 (2018) 18, S. 1–52.
- Liang, E., Liaw, R., Moritz, P., Nishihara, R., Fox, R., Goldberg, K., Gonzalez, J.E., Jordan, M.I., Stoica, I., RLlib: Abstractions for Distributed Reinforcement Learning, Proceedings of the 35 th International Conference on Machine Learning, 35. Aufl., 2018.
- Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I., Tune: A Research Platform for Distributed Model Selection and Training, in: ICML/IJCAI-ECAI (Hrsg.), Proceedings of the Thirty-fifth International Conference on Machine Learning, 35. Aufl., 2018, S. 1–8.
- Lillicrap et al., Continuous control with deep reinforcement learning, 2015.
- Lin, P., The Data Science Process, 2004, URL: [https://www-01.ibm.com/events/wwe/grp/grp304.nsf/vLookupPDFs/Polong%20Lin%20Presentation/\\$file/Polong%20Lin%20Presentation.pdf](https://www-01.ibm.com/events/wwe/grp/grp304.nsf/vLookupPDFs/Polong%20Lin%20Presentation/$file/Polong%20Lin%20Presentation.pdf), gelesen am 30.04.2019.
- Lustig, I., Dietrich, B., Johnson, C., Dziekan, C., The Analytics Journey, in: Analytics Magazine, 2010 (2010), S. 11–18.
- Mammeri, Z., Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches, in: IEEE Access, 7 (2019), S. 55916–55950.
- Matiisen, T., THE USE OF EMBEDDINGS IN OPENAI FIVE, 2018, URL: <https://neuro.cs.ut.ee/the-use-of-embeddings-in-openai-five/>, gelesen am 12.06.2019.
- Mauerer, J., Was ist was bei Predictive Analytics?, 2018, URL: <https://www.cio.de/a/was-ist-was-bei-predictive-analytics,3098583,4>, gelesen am 08.08.2018.
- Mayntz, R., Holm, K., Hübner, P., Einführung in die Methoden der empirischen Soziologie, 5. Aufl. Aufl., Westdt. Verl., Opladen, 1978.

- Mensing, W., Erfolgreiches Projektmanagement ohne externe Berater in KMUs, Springer Gabler, Wiesbaden, 2015.
- Mertens, P., Meier, M.C., Integrierte Informationsverarbeitung 2, 10., vollständig überarbeitete Auflage. Aufl., Gabler Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, Wiesbaden, 2009.
- Messmittelmanagement gemäß ISO 9001, URL: https://www.qz-online.de/qualitaetsmanagement/qm-basics/messen_pruefen/messtechnik/artikel/messmittelmanagement-gemaess-iso-9001-176025.html, gelesen am 08.03.2019.
- Mezick, D., Pfeffer, J., Pontes, D., Sasse, M., Sheffield, M., Shinsato, H., Kold-Taylor, L., Das OpenSpace Agility Handbuch, 1. Auflage. Aufl., peppair, Wangen, 2019.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., Human-level control through deep reinforcement learning, in: Nature, 518 (2015) 7540, S. 529–533.
- Müller-Hengstenberg, C.D., Rechtliche Risiken autonomer und vernetzter Systeme, EBOOK PACKAGE COMPLETE 2016 : EBOOK PACKAGE Engineering, Computer Sciences 2016, De Gruyter Oldenbourg, Berlin, Boston.
- Nelson, S., 4 Types of Data Science Jobs, 2018, URL: <https://blog.udacity.com/2018/01/4-types-data-science-jobs.html>, gelesen am 16.04.2019.
- OECD/European Union, Health at a Glance, 2019. Aufl., OECD Publishing, 2018.
- Ortega, P., Maini, V., Building safe artificial intelligence: specification, robustness, and assurance, 2018a, URL: <https://medium.com/@deepmindsafetyresearch/building-safe-artificial-intelligence-52f5f75058f1>, gelesen am 01.10.2018.
- Ortega, P., Maini, V., Building safe artificial intelligence: specification, robustness, and assurance, 2018b, URL: <https://medium.com/@deepmindsafetyresearch/building-safe-artificial-intelligence-52f5f75058f1>, gelesen am 13.06.2019.
- Österle, H., Winter, R., Brenner, W., Gestaltungsorientierte Wirtschaftsinformatik, Infowerk, Nürnberg, 2010.
- Pahor, M., Guralnik, J.M., Ambrosius, W.T., Blair, S., Bonds, D.E., Church, T.S., Espeland, M.A., Fielding, R.A., Gill, T.M., Groessl, E.J., King, A.C., Kritchevsky, S.B., Manini, T.M., McDermott, M.M., Miller, M.E., Newman, A.B., Rejeski, W.J., Sink, K.M., Williamson, J.D., Effect of structured physical activity on prevention of major

- mobility disability in older adults: the LIFE study randomized clinical trial, in: JAMA, 311 (2014) 23, S. 2387–2396.
- Pathak et al., Curiosity-driven Exploration by Self-supervised Prediction, 2017.
- Patig, S., Dibbern, J., Requirements Engineering, 2018, URL: <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaeten-der-Systementwicklung/Problemanalyse-/Requirements-Engineering/index.html>, gelesen am 04.02.2019.
- Pepels, W., Bruns, J., Verfahren, Datenauswertung, Ergebnisdarstellung, 2. überarb. und erw. Aufl. Aufl., Symposion Publ, Düsseldorf, 2008.
- Persse, J.R., The ITIL process manual, Van Haren Publishing, Zaltbommel, Netherlands, 2012.
- Pfetzing, K., Rohde, A., Ganzheitliches Projektmanagement, 3., bearb. Aufl. Aufl., Götz Schmidt, Gießen, 2009.
- Pfeufer, H.-J., FMEA, 1. Aufl. Aufl., Hanser, München, 2015.
- Phillips-Wren, G., Intelligent Decision Support Systems, in: Doumpos, M., Grigoroudis, E. (Hrsg.), Multicriteria Decision Aid and Artificial Intelligence, John Wiley & Sons, Ltd, Chichester, UK, 2013, S. 25–44.
- Polasek, W., EDA Explorative Datenanalyse, Zweite, neubearbeitete und erweiterte Auflage. Aufl., Springer, Berlin, Heidelberg, 1994.
- Ponniah, P., Data modeling fundamentals, Wiley-Interscience, Hoboken, N.J, 2007.
- Preißner, A., Marketing auf den Punkt gebracht, De Gruyter, München, 2008.
- Puget, J.F., Analytics Landscape, 2015, URL: https://www.ibm.com/developerworks/community/blogs/jfp/entry/Analytics_Models?lang=en, gelesen am 08.08.2018.
- Pumperla, M., Ferguson, K., Graepel, T., Deep learning and the game of Go, Manning, Shelter Island, New York, 2019.
- Raghu et al., Deep Reinforcement Learning for Sepsis Treatment, 2017.
- Rempp, G., Model Driven SOA, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.
- Richardson, C., Microservice Patterns MEAP, Manning, Shelter Island, NY, 2017.
- Riedl, R., Strategische Planung von Informationssystemen, Physica-Verlag HD, Heidelberg, 1991.

- Risch, A.K., Stangier, U., Heidenreich, T., Hautzinger, M., Kognitive Erhaltungstherapie bei rezidivierender Depression, Springer-Verlag Berlin Heidelberg New York, Berlin, Heidelberg, 2012.
- Rothgang et al., Themenreport „Pflege 2030“, 2012.
- Rupp, C., Simon, M., Hocker, F., Requirements Engineering und Management, in: HMD Praxis der Wirtschaftsinformatik, 46 (2009) 3, S. 94–103.
- Russell, S.J., Norvig, P., Artificial intelligence, 2. ed. Aufl., Prentice Hall, Upper Saddle River, NJ, 2003.
- Schäfer, C., Grundlagen der Patientencompliance und Adhärenz, in: Schäfer, C. (Hrsg.), Patientencompliance, Springer Fachmedien Wiesbaden, Wiesbaden, 2017, S. 13–40.
- Schnell, R., Hill, P.B., Esser, E., Methoden der empirischen Sozialforschung, 5., völlig überarb. und erw. Aufl. Aufl., Oldenbourg, München, 1995.
- Schulman et al., Proximal Policy Optimization Algorithms, 2017.
- Seemann, J., Gudenberg, J.W., Software-Entwurf mit UML, Springer, Berlin, Heidelberg, 2000.
- Seiter, M., Business Analytics, Franz Vahlen, München, 2017.
- Silver, D., Lecture 7: Policy Gradient, 2018a.
- Silver, D., Lecture 8: Integrating Learning and Planning, 2018b, URL: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/dyna.pdf, gelesen am 03.06.2019.
- Skowron, Methode zur Gestaltung sicherer präskriptiver Systeme für die Therapieunterstützung, 2019a.
- Skowron, P., Künstliche Intelligenz ist auch nur ein Service. JUG Saxony Day 2019b, Dresden.
- Sommerville, I., Software engineering, 9. ed. Aufl., 2011.
- Sowa, A., Fedtke, S., Metriken - der Schlüssel zum erfolgreichen Security und Compliance Monitoring, 1. Aufl. Aufl., Vieweg+Teubner (GWV), s.l., 2011.
- Spreckelsen, C., Spitzer, K., Wissensbasen und Expertensysteme in der Medizin, Vieweg+Teubner, Wiesbaden, 2009.
- Stachowiak, H., Allgemeine Modelltheorie, Springer, Wien, 1973.
- Stewart, T., McMillan, C., Descriptive and prescriptive models for judgment and decision making: implications for knowledge engineering, in: Expert judgment and expert systems, 1987 (1987), S. 305–320.

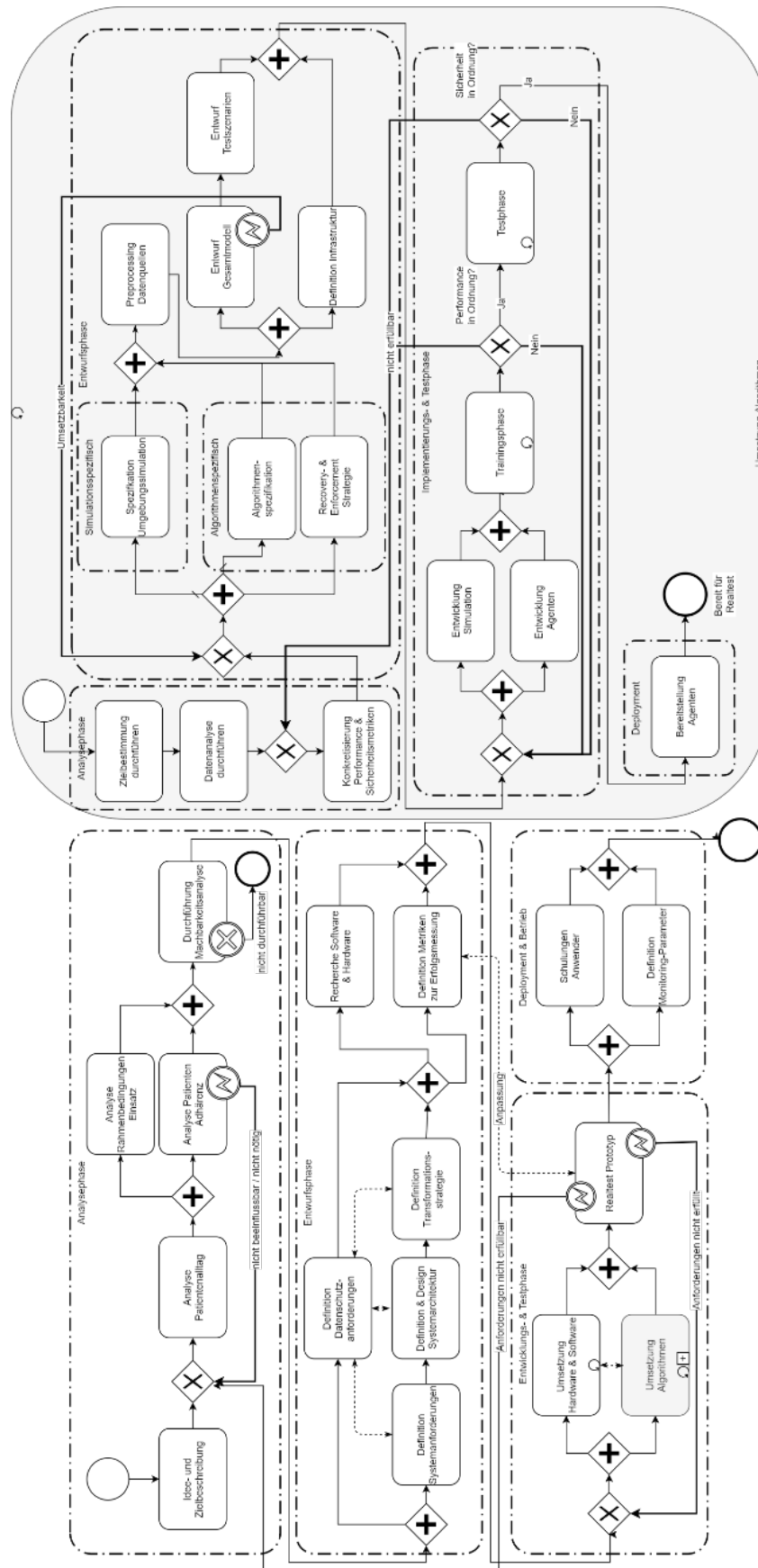
- Stimmel, C.L., Big data analytics strategies for the smart grid, CRC Press, Boca Raton, FL, 2015.
- Sutton, R.S., Barto, A.G., Reinforcement learning, [Nachdr.]. Aufl., MIT Press, Cambridge, Mass., 2018.
- Sutton, R.S., McAllester, D., Singh, S., Mansour, Y., Policy Gradient Methods for Reinforcement Learning with Function Approximation, Proceedings of the 12th International Conference on Neural Information Processing Systems, MIT Press, Cambridge, MA, USA, 1999, S. 1057–1063.
- Tesla kracht fast ungebremst in verunglückten Lkw, 2020, URL: <https://www.welt.de/vermishtes/article208873091/Taiwan-Tesla-rast-trotz-Autopilot-fast-ungebremst-in-liegenden-Lkw.html>, gelesen am 08.09.2020.
- Tetali Perraju, ARTIFICIAL INTELLIGENCE AND DECISION SUPPORT SYSTEMS, 2013.
- Tewes, R., Führungskompetenz ist lernbar, 2. aktualisierte Auflage. Aufl., Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.
- Tiedemann, M., Vielfach unterschätzt, aber unschätzbar wertvoll: Der Data Engineer, 2017, URL: <https://www.alexanderthamm.com/de/artikel/vielfach-unterschaetzt-unschaetzbar-wertvoll-der-data-engineer/>, gelesen am 16.04.2019.
- Tiemeyer, E., Handbuch IT-Projektmanagement, 2., überarb. und erw. Aufl. Aufl., Hanser, München, 2014.
- Tremblay et al., Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization, 2018.
- Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Vezhnevets, A.S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T.L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., Silver, D., Grandmaster level in StarCraft II using multi-agent reinforcement learning, in: Nature, 575 (2019) 7782, S. 350–354.
- V-Modell XT, 2006, URL: <http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/2.2/V-Modell-XT-Gesamt.pdf>, gelesen am 25.02.2019.

- Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.,
Reconstructing the giant: On the importance of rigour in documenting the literature
search process, ECIS, 2009, S. 2206–2217.
- Weiss, G., Multiagent systems, 2. print. Aufl., MIT Press, Cambridge, Mass., 2000.
- Wieczorrek, H.W., Mertens, P., Management von IT-Projekten, 2., überarbeitete und
erweiterte Auflage. Aufl., Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg,
2007.
- Winkelhofer, G.A., Methoden für Management und Projekte, Springer, Berlin, Heidelberg,
1997.
- World Health Organization, Classification of Digital Health Interventions v1.0, 2018.
- World Medical Association, WMA Deklaration von Helsinki - Ethische Grundsätze für die
medizinische Forschung am Menschen, 64. WMA-Generalversammlung, 2013.
- Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., Hochberg, I.,
Encouraging Physical Activity in Patients With Diabetes: Intervention Using a
Reinforcement Learning System, in: Journal of medical Internet research, 19 (2017)
10, 338-349.
- Yoon, J., Davtyan, C., van der Schaar, M., Discovery and Clinical Decision Support for
Personalized Healthcare, in: IEEE journal of biomedical and health informatics, 21
(2017) 4, S. 1133–1145.
- Zahavy, T., Zrihem, N.B., Mannor, S., Graying the black box: Understanding DQNs, in:
ArXiv (2017) abs/1602.02658, S. 1–32.
- Zhang, L., Alharbe, N.R., Atkins, A.S., A self-adaptive distributed decision support model
for Internet of Things applications, in: Transactions of the Institute of Measurement
and Control, 39 (2017) 4, S. 404–419.

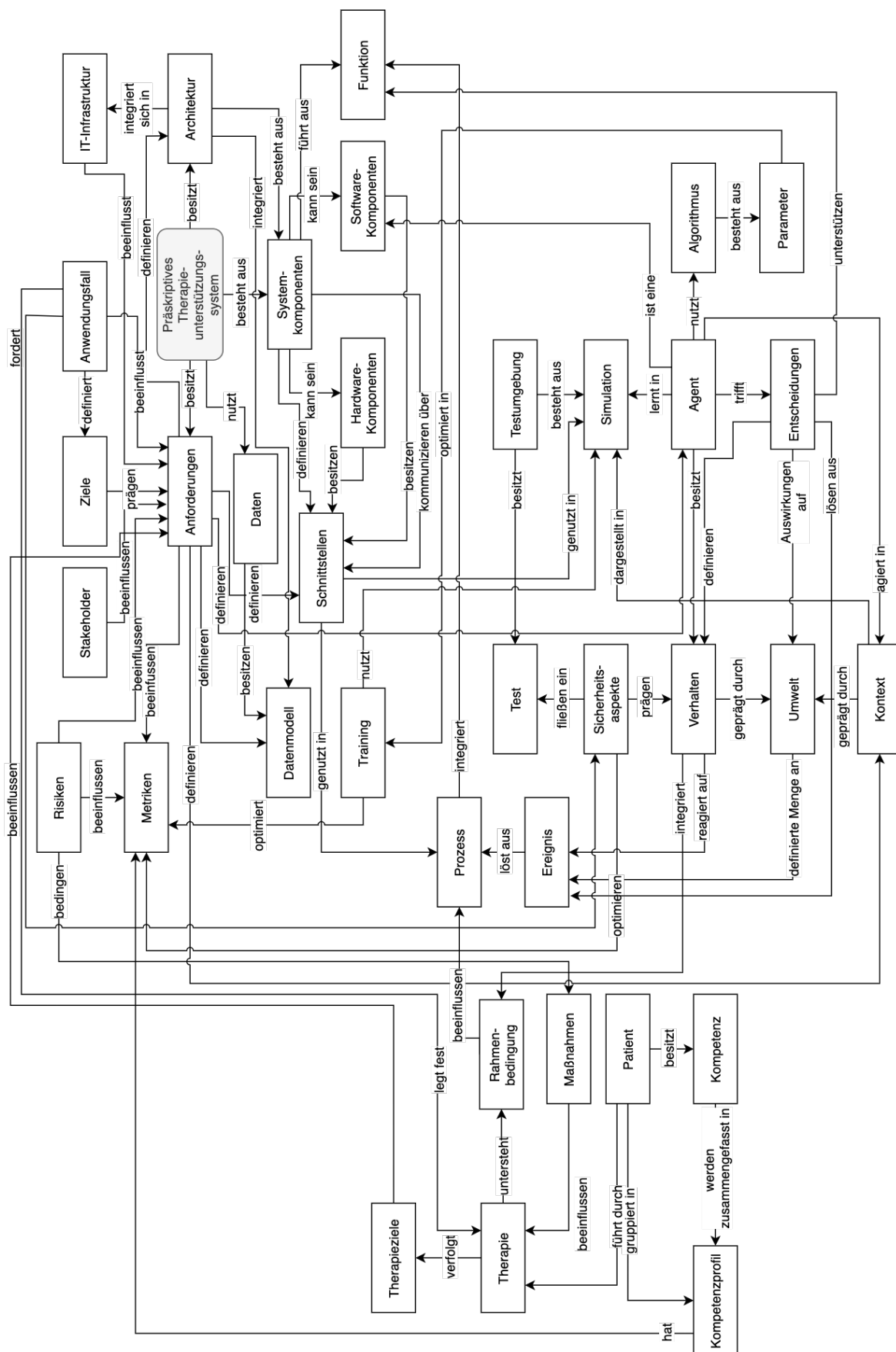
Anhang

Anhang

Anhang 1: Vorgehensmodell der Methode



Anhang 2: Metamodell der Methode



Anhang 3: Hyperparameter Suchparameter

```
44 # Hyperopt search space
45
46 search_space = {
47     "lr": hp.uniform("lr", 0.001, 0.1),
48     "train_batch_size": hp.choice("train_batch_size", [128, 256, 512]),
49     "model": {
50         "custom_options": {
51             "lstm_cell_size": hp.choice("lstm_cell_size", [64, 128, 256]),
52             "pre_dense_size": hp.choice("pre_dense_size", [64, 128, 256]),
53             "post_dense_size": hp.choice("post_dense_size", [64, 128, 256]),
54             "emb_size": hp.choice("emb_size", [8, 16, 32])
55         }
56     }
57 }
58
59 init_params = [
60     {
61         "lr": 0.007,
62         "train_batch_size": 2,
63         "lstm_cell_size": 2,
64         "pre_dense_size": 1,
65         "post_dense_size": 2,
66         "emb_size": 1
67     }
68 ]
```

Anhang 4: RL Modell

```

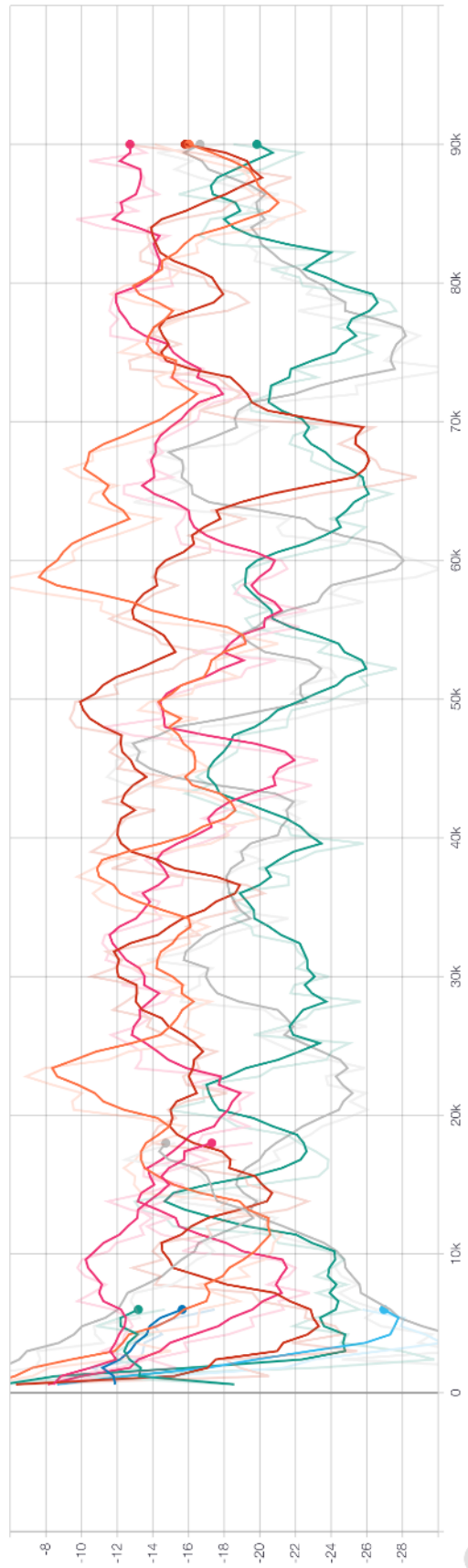
15 class RoutingModel(TFModelV2):
16     def __init__(self, obs_space, action_space, num_outputs, model_config, name):
17         super(RoutingModel, self).__init__(obs_space, action_space, num_outputs, model_config, name)
18
19         # Cell Größen
20         self._lstm_cell_size = model_config['custom_options']['lstm_cell_size']
21         self._pre_dense_size = model_config['custom_options']['pre_dense_size']
22         self._post_dense_size = model_config['custom_options']['post_dense_size']
23         self._emb_size = model_config['custom_options']['emb_size']
24
25         # Shapes holen
26         obs_general_shape = obs_space.original_space['general'].shape # (2,)
27
28         # Observation Inputs
29         self.general_inputs = tf.keras.layers.Input(shape=obs_general_shape, name="general_obs")
30         self.wayprops_input = tf.keras.layers.Input(shape=(5, 1), name="wayprops_obs")
31         self.waytype_input = tf.keras.layers.Input(shape=(5, 1), name="waytype_obs")
32
33         # Preprocessing Observations
34         x = tf.keras.layers.Dense(128, activation=tf.nn.relu, name="pre_obs")(self.general_inputs)
35
36         # Preprocessing Wayprops
37         y = tf.keras.layers.Embedding(4, self._emb_size, input_length=1, name="emb")(self.waytype_input)
38
39         w = tf.keras.layers.Dense(32, activation=tf.nn.relu, name="pre_wayprop")(self.wayprops_input)
40         w = tf.keras.layers.Reshape((5, 1, 32), name="reshape_wayprop")(w)
41         # w = tf.expand_dims(w, 2)
42         w = tf.keras.layers.Concatenate()([y, w])
43
44         v = tf.keras.layers.GlobalMaxPooling2D()(w)
45         v = tf.keras.layers.Dense(128, activation=tf.nn.relu)(v)
46
47         # Preprocessing zusammenführen
48         concatenate_pre = tf.keras.layers.Concatenate()([x, v])
49         concatenate_pre = tf.keras.layers.Dense(self._pre_dense_size, activation=tf.nn.relu, name="pre_conc")(concatenate_pre)
50         concatenate_pre = tf.keras.layers.Reshape((1, self._pre_dense_size), name="reshape_conc")(concatenate_pre)
51         # concatenate_pre = tf.expand_dims(concatenate_pre, 1)
52         # concatenate_pre = tf.keras.layers.Lambda(check_dimensions, name="check_dim_wayprops")(concatenate_pre)
53
54         # GRU CPU
55         gru_out = tf.keras.layers.GRU(self._lstm_cell_size,
56                                     return_sequences=False,
57                                     return_state=False,
58                                     name="brain")(concatenate_pre)
59
60         # Postprocessing
61         z = tf.keras.layers.Dense(self._post_dense_size, activation=tf.nn.relu)(gru_out)
62         value_out = tf.keras.layers.Dense(1, activation=None)(z)
63         layer_out = tf.keras.layers.Dense(num_outputs, activation=tf.nn.softmax)(z)
64
65         # Modell bauen
66         self.rnn_model = tf.keras.Model(
67             inputs=[self.general_inputs, self.waytype_input, self.wayprops_input],
68             outputs=[layer_out, value_out])
69
70         self.register_variables(self.rnn_model.variables)
71         # self.rnn_model.summary()
72

```


Anhang 5: RL Modell Masking

```
73 def forward(self, input_dict, state, seq_lens):
74     # Wayprops aufteilen (5,2) -> 2x (5,1)
75     way_props, way_type = tf.split(input_dict['obs']['way_props'], [1, 1], -1)
76
77     model_out, self._value_out = self.rnn_model([input_dict['obs']['general'], way_type, way_props])
78
79     return self._masking(tf.reshape(model_out, [-1, self.num_outputs]), input_dict['obs']['action_mask']), state
80
81 def _masking(self, logits, mask):
82     mask = tf.cast(mask, tf.bool)
83     neg_inf = logits.dtype.min
84     neg_inf = tf.cast(tf.fill(dims=tf.shape(input=logits), value=neg_inf), logits.dtype)
85     logits = tf.where(mask, logits, neg_inf)
86
87     return logits
88
```

Anhang 6: Hyperparametertuning Ablauf



Anhang 7: Methoden der Adhärenz Messung (vgl. Schäfer 2017)

Direkte Methode	Beispiel	Anmerkung
Beobachtung	Tabletteneinnahme unter Aufsicht	I.d. Regel ungeeignet in der Praxis; evtl. im Krankenhaus oder Pflegeheim geeignet
Bestimmung des Arzneimittels oder eines Metaboliten in biologischen Flüssigkeiten	In Blut, Urin, Stuhl oder Speichel	Geeignet in vielen Fällen
Messung eines biologischen Markers	Messung von pharmakologisch unwirksamen Dosen einer Markersubstanz, die einem Placebo oder einer Behandlung beigelegt wird	Nur in klinischen Prüfungen geeignet

Indirekte Methode	Beispiel	Anmerkung
Patientenbefragung / Interviews	Mündliche oder schriftliche Befragung z.B. durch offene Fragen, strukturierte Interviews oder Fragebogen.	Für die Praxis gut geeignet.
„Klinische“ Einschätzung durch den Arzt oder Apotheker	Einschätzung nach Gefühl unter Berücksichtigung von Ansprechen auf Therapie, Patienten-Charakteristika und Erfahrung.	Relativ unzuverlässig, nur 50% Trefferwahrscheinlichkeit.
Arzneimittelverbrauch im Verhältnis zur Verordnung bilanzieren	„Pill count“, Tablettenzählen	In der Praxis mittels Patientenkarten erfassbar.
In Arzneimittelpackung eingebaute elektronische Verbrauchsmonitore	MEMS = medication event monitoring system	Besonders geeignet zur Erfassung des Adhärenzmusters.
Therapeutische Wirkparameter	Bluthochdruck, Blutzucker, Herzfrequenz	Über Befragung und Messung in der Praxis erfassbar.
Auswertung von Patiententagebüchern über den Medikamentverbrauch	Bei Asthma, Diabetes, Hypertonie mit Dokumentation von Medikation und Wirkparametern.	Keine starke Abhängigkeit von dem Erinnerungsvermögen, aber Schlagseite zum Überschätzen durch compliancegemäßes Ausfüllen.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass

1. die vorgelegte Dissertation ohne unzulässige Hilfe, insbesondere ohne die Inanspruchnahme eines Promotionsberaters, und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde und dass die aus fremden Quellen direkt oder indirekt übernommenen Gedanken in der Arbeit als solche kenntlich gemacht worden sind und
2. die vorgelegte Dissertation weder im Inland noch im Ausland in gleicher oder in ähnlicher Form einer anderen Prüfungsbehörde zum Zwecke einer Promotion oder eines anderen Prüfungsverfahrens vorgelegt und insgesamt noch nicht veröffentlicht wurde.

Leipzig, den 05. Oktober 2020

(Ort, Datum)

.....

(Unterschrift)